New Interfaces for Musical Expression

NIME 06

IRCAM – Centre Pompidou
Paris, France
June 4 - 8, 2006

Proceedings edited by

Norbert Schnell
Frédéric Bevilacqua
Michael Lyons
Atau Tanaka

Proceedings published by

IRCAM – Centre Pompidou
in collaboration with MINT/OMF – Sorbonne University

Copies may be ordered from

Service des Publications
IRCAM – Centre Pompidou
1, place Igor Stravinsky
75004 Paris, France

Web Sites

http://nime06.ircam.fr
http://www.nime.org

Cover design and photography

Heidi Leigh Cregg

ISBN 2-84426-314-3

ISBN 978-2-84426-314-8

NIME 06 is supported by

IRCAM – Centre Pompidou

MINT/OMF – Sorbonne University

Sony CSL Paris

ArsLonga
http://www.arslonga.org/

Point Éphémère
http://www.pointephemere.org/

Tripyque
http://www.letriptyque.com/

Neo Pop Art

Centre National de la Recherche Scientifique
http://www.cnrs.fr/

Association Française d'Informatique Musicale
http://www.afim-asso.org/

Cost287 – ConGAS
http://www.cost287.org/

I-MAESTRO
http://www.i-maestro.net/

LAM Research Network
http://www.livealgorithms.org/

STEIM
http://www.steim.org/

Apple Computer, Inc.

Microsoft Research

Yamaha Corporation

NIME 06 is organized by

IRCAM – Centre Pompidou in collaboration with
MINT/OMF – Sorbonne University as a part of the
Agora Festival and the Résonances Convention on Technologies for Music.

IRCAM Director: Frank Madlener
IRCAM R&D Director: Hugues Vinet
IRCAM Production Director: Alain Jacquinot
Résonances Director: Vincent Puig

NIME 06 Committee

Conference Chairs
Norbert Schnell
Frédéric Bevilacqua

Paper and Poster Chair
Michael Lyons

Performance and Installation Chair
Atau Tanaka

Demo Chair
Emmanuel Fléty

Administrative Coordinator
Sylvie Benoit

Local Organization

Agora / Résonances
Suzanne Berthy (coordinator)
Rachel Addra
Dany Baudouin
Murielle Ducas
Sylvia Gomes
Aude Grandveau
Vincent Gourson
Stéphanie Leroy
Olivier Lescurieux
Deborah Lopatin
Amanda MacBlane
Claire Marquet
Béatrice Montfort
Delphine Oster
Paola Palumbo
Véronique Pré

IRCAM Production
Pascale Bondu (coordinator)
Jerôme Boussard
François Gibouin
Thomas Leblanc
Nicolas Leroy
Remy Müller
David Poissonnier
Daniel Raguin

IRCAM Systems
Laurent Ghys (coordinator)
Arnaud Gomes-do-Vale
Youcef Bensaid
Olivier Labat
Frédéric Vandromme

ArsLonga (Club NIME)
Olivier Lescurieux (president)
Alexis Alyskewicz
Rémi Bonin

Proceedings Organizers
Marc Battier
Bruno Bossis
Pierre Couprie
Alireza Farhang
Elsa Filipe

NIME Steering Committee
Ivan Poupyrev
Michael Lyons
Tina Blaine
Atau Tanaka
Joe Paradiso
Marcelo Wanderley
Sile O'Modhrain
Sidney Fels
Yoichi Nagashima

Paper and Poster Program Committee

Meta-Reviewers
Michael Lyons (chair)
Gideon d'Arcangelo
Frédéric Bevilacqua
Emmanuel Fléty
Lalya Gaye
Sergi Jorda
Norbert Schnell
Marcelo Wanderley

Reviewers
Roberto Aimi
Xavier Amatriain
Daniel Arfib
Mark Argo
Marije Baalman
Curtis Bahn
Luke Barrington
Kirsty Beilharz
Tina Blaine
Sinan Bokesoy
Bert Bongers
Jan Borchers
Andrew Brouse
Jonah Brucker-Cohen
Nick Bryan-Kinns
Matthew Burtner
Joel Chadabe
Elaine Chew
Art Clay
Thierry Coduys
Perry Cook
David Cournapeau
Jean-Michel Couturier
Alain Crevoisier
Roger Dannenberg
Roman Danylak
Christopher Dobrian
John Eacott
Georg Essl
Stuart Favilla
Sid Fels
Brian Fisher
Edgar Flores
Alexandre Francois
Ichiro Fujinaga
David Gerhard
Rolf Inge Godøy
Suguru Goto
Saul Greenberg
Michael Gurevich
Keith Hamel
Kjetil F. Hansen
Björn Hartmann
Lars Erik Holmquist
Andy Hunt
Bob Huott
Alejandro Jaimes
Joanne Jakovich
Jordi Janer
Andrew Johnston
Linda Kaastra
Martin Kaltenbrunner
Ajay Kapur
Haruhiro Katayose
Loic Kessous
Mari Kimura
Ben Knapp
Julian Knowles
Jean-Baptiste Labrune
Rodger Lea
Eric Lee
Paul Lehrman
Sasha Leitman
Golan Levin
Michael Lew
Takuro Lippit
Thor Magnusson
Steve Mann
Jonatas Manzolli
Andrew Martin
Kenji Mase
Teemu Mäki-Patola
Ronald McCaig
Eduardo Miranda
Katherine Moriwaki
Axel Mulder
Yoichi Nagashima
Teresa Marrin Nakra
Kia Ng
Kazushi Nishimoto
Sile O'Modhrain
Dan Overholt
Gath Paine
Joseph Paradiso
Cornelius Poepel
Sam Roig Torrubiano
Gary Scavone
Andrew Schloss
Erwin Schoonderwaldt
Stefania Serafin
Xavier Serra
Eric Singer
Hugo Solís
Christa Sommerer
Laetitia Sonami
Hans-Christoph Steiner
Sachiyo Takahashi
Atau Tanaka
Anthony Tang
Daniel Trueman
George Tzanetakis
Doug Van Nort
Gualtiero Volpe
Ge Wang
Gil Weinberg
Tomoko Yonezawa
Diana Young
Michael Zbyszynski

Performance and Installation Program Committee
Atau Tanaka (chair)
Mark Applebaum
Jean-Baptiste Barrière
Magali Desbazeille
Gerhard Eckel
Toru Iwatake
Sergi Jorda
Mari Kimura
Ali Momeni
Dafna Naphtali
Garth Paine
Edwin van der Heide
Michel Waisvisz
Christian Zanesi

Special Thanks to
Michel Fingerhut
Riccardo Borghesi
Diemo Schwarz
Alice Daquet

Welcome to NIME 06 at IRCAM

Unique par l'ampleur de ses domaines réunissant la création musicale, la recherche et la transmission, l'IRCAM est l'un des rares lieux où l'intuition artistique, la formalisation scientifique et l'expérimentation technologique se confrontent et avancent ensemble. L'histoire riche de l'IRCAM n'est pas qu'une succession d'œuvres paradigmatiques, de noms de chercheurs pionniers ou de développements de logiciels utilisés par une vaste communauté d'artistes. Cette histoire tumultueuse qui est celle du développement du matériau sonore et de la pensée musicale, repose sur l'effervescence intellectuelle, expérimentale et artistique de l'institution. Les œuvres de Pierre Boulez, fondateur de l'IRCAM, d'Emmanuel Nunes, de Philippe Manoury, de Georges Aperghis, de Jonathan Harvey et de compositeurs plus jeunes sont nées de l'échange intense entre la singularité artistique et les avancées scientifiques dans les domaines de la synthèse sonore, du temps réel, de la lutherie numérique, de la composition assistée par ordinateur, de la spatialisation, de l'acoustique…

Si l'histoire de l'Ircam est saisie par le concept d'écriture autour duquel se noue l'intrigue entre le compositeur, l'assistant musical et la communauté des chercheurs, la présence d'artistes inclassables de l'ère digitale constitue une provocation stimulante ! Tout à la fois improvisateurs, « performeurs », inventeurs d'une nouvelle lutherie, expérimentateurs des technologies numériques, ces artistes repoussent les distinctions entre ce qui est prémédité et composé, et font de l'interaction leur principal credo. Ce champ dynamique, fécond dans le spectacle vivant, est appelé à s'étendre dans les années qui viennent. Les créations de l'Ircam sur les plateaux du théâtre, de l'opéra, de la danse, rencontrent les questions soulevées par cette édition du NIME. L'avenir de l'IRCAM du reste, appelle la mobilité de tout le dispositif technologique, le renouvellement des présences créatrices, la convergence entre création vécue et prospective. Entre le chercheur qui vise des valeurs de connaissance, l'artiste qui pose un monde et des problèmes neufs, et le dispositif technique, le terrain de rencontre privilégié reste le surgissement d'une œuvre. L'ampleur tumultueuse de l'Ircam qui entrechoque deux temps divergents, l'acte qui fulgure et la construction minutieuse qui formalise, l'impatience de ce qui va émerger et la lenteur de ce qui couve, l'intempestif et la fonction, cette ampleur, une synthèse disjonctive et électrisante, constitue la singularité et l'originalité présente de l'IRCAM.

Nous sommes très heureux d'accueillir à Paris les participants de la sixième édition du NIME au moment de l'ouverture d'Agora et des Rencontres Résonances, qui donne l'occasion de discuter, d'échanger, d'expérimenter mais aussi de vivre la création musicale au cours du festival de l'Ircam.

Unique for the scope of its activities that bring together musical creation, research, and transmission, Ircam is one of the rare places where artistic intuition, scientific rationality, and technological experimentation intersect and advance together. Ircam's rich history is not simply a succession of paradigmatic works, of the names of pioneering researchers, or of software developments used by a vast community of artists. Its eventful history, which has led to advancement in both sound material and musical thought, is the result of the intellectual, experimental, and artistic effervescence of the institution. The works of Pierre Boulez (Ircam's founder), Emmanuel Nunes, Philippe Manoury, Georges Aperghis, Jonathan Harvey, and other younger composers were born of the intense exchange between artistic originality and scientific progress in the fields of sound synthesis, real-time, digital instrument-making, computer-assisted composition, spatialization, and acoustics, among others.

If Ircam's history can be captured in the image of a composer, a musical assistant, and the community of researchers working together to write music, the presence of outstanding artists from the digital era represents a stimulating challenge! Improvisers, performers, instrument inventers, and experimenters in digital technology, these artists push the limits of what is planned and what is composed, and make "interaction" their principal credo. This dynamic field, so fruitful in the context of live performance, should expand in the coming years. Ircam creations for the theater, the opera, and dance are often faced with the same questions being addressed this year at NIME. For the rest of Ircam's activities, the future calls for technological mobility, the renewal of creative participation, and the convergence between past and potential creations. For the researcher who values knowledge, the artist who proposes new worlds and new problems, and the technical aspects, the preferred meeting place is still the production of a work. Ircam's cacophonous range of activities, which forces together two diverging tempos—the act that dazzles and the meticulous construction that formalizes it, the impatience of what will emerge and the slow progress of gestation, the incompatible and the functional—this range, a disjointed and electrifying synthesis, constitutes the uniqueness and the originality of Ircam today.

We are very happy to welcome the participants of the sixth installment of NIME to Paris just as Agora and Résonances open their doors, creating a unique opportunity for discussing, exchanging, experimenting, and sharing the experience of musical creation throughout the Ircam festival.

Frank Madlener
Director

Introduction

We are proud to present the 6th edition of NIME, hosted by IRCAM – Centre Pompidou. Entering the sixth year of NIME we sense maturity and ongoing enthusiasm. We are pleased to base NIME 06 on the acquisitions of an exceptional and dedicated community of scientists, technologists, and artists grown over the past few years.

The response to the NIME 06 call for participation was overwhelming: We received 197 paper and poster submissions as well as 99 performance and 44 installation proposals. To accommodate this expansion, this year we have extended the conference to include three and a half days of paper, poster, and demo sessions as well as five concerts, composing a rich and balanced program of 88 papers and posters, 25 performances, and 9 installations.

The significant increase in the number of submissions made the selection process more complex and difficult than ever before. Both selection committees - scientific and artistic – rose to the challenge and ensured an efficient and fair selection process. We feel that this year's program is a compelling manifestation of the conference's maturity and the result of the continuing fertile creativity of research and artistic work in this field. We would like to thank the authors for the impressive quality of their submissions and the reviewers for their excellent insight as well as their active participation in and contribution to this community.

Our commitment as organizers is to compose a program that best reflects the multitude of complementary and multidisciplinary aspects involved in NIME. In this spirit, several focused events have been created in collaboration with various partners.

We are honored to welcome several guest speakers and performers to NIME 06. George Lewis, composer, improvising musician, professor for American music at Columbia University, theorist, and historian, is the ideal speaker to lead a discussion on the field. We have also invited William Gaver, researcher and teacher at Goldsmiths College to provide a complementary perspective, based on human-computer interaction and the applications that concern everyday life. George Lewis and William Gaver will give the two keynote addresses on the first two days of the meeting.

The special session on *Digital Interfaces for the Violin Family* to be held on the fourth day of the meeting brings together the NIME community around a single, specific topic. The papers presented at the special session were subjected to the same peer-review process as the other NIME 06 paper submissions. The plenary presentations are open to the public and are accompanied by a few additional presentations that are not included in these proceedings.

We thought that the violin was the perfect centerpiece for such a session as it articulates both the future and the past of the ongoing avant-garde experimentation in lutherie with new materials and techniques dedicated to human expression and poetry. As a composer and performer who has branched out from the traditional violin repertoire to embrace the world of interactive computer music, our invited guest Mari Kimura personifies this idea for NIME 06.

The violin also strongly evokes the notion of virtuosity, which we would like to emphasize this year. Michel Waiswisz, who has been working with electrical and sensor-based musical instruments since the 1960's, will lead a panel discussion of NIME performers on virtuosity in live electronic music on the third day of the meeting.

Finally, NIME 06 will host three workshops on topics of increasing pertinence: *Improvisation and Computers*, *Motion Capture and Analysis for Dance*, and *Network Performance*.

NIME has the unique opportunity this year of taking place within the broader context of the Agora/Résonances 2006 festival at IRCAM. We have endeavored to bring NIME to a wider audience by creating free access to the special session, the keynote addresses, and the panel discussion. Meanwhile, we introduce NIME delegates to the very active local Parisian scene by arranging studio tours and events in independent venues throughout the city. A Special Guests concert curated by IRCAM embodies the synergy between NIME and Agora.

These proceedings comprise the papers and posters from the conference, as well as an overview of the artistic program and its concerts, installations, club, and gallery events. Further documentation of the artistic program and demonstrations, in the form of recordings and video, as well as electronic versions of the scientific program, will be made available online after the conference.

The monumental task of organizing a conference would not have been possible if it were not for the support of IRCAM and the other partners of NIME 06. We would like to thank very warmly all the people involved and give particular thanks to Emmanuel Fléty and Sylvie Benoit, our closest collaborators in the NIME 06 adventure.

We hope that NIME 06 represents another milestone in the development of the diverse, original, and open NIME community.

Enjoy NIME06!

Norbert Schnell, Frederic Bevilacqua, Michael Lyons, and Atau Tanaka

NIME 06 Artistic Program – a Rough Guide

We saw an unprecedented number and diversity of submissions for the NIME 06 artistic program, creating for the first time, to my knowledge, a situation where no single review committee member was able to view and evaluate every submitted work. We endeavored to break out of the coldness of score-ranking on the online review system by engaging in subjective discussion, and by asking ourselves hard questions about the musical tenets at the basis of NIME, putting to task its history, and hopefully identifying potential new directions. We attempted to put in place programming that matched this approach and the body of work selected, by at once bringing to Ircam works that otherwise may not have been heard in the Espace de Projection, and at the same time taking NIME out into the city of Paris to independent galleries, clubs, and performance venues.

After making an initial selection, we sought to identify emerging themes in the works, and to create strong concert programs revolving around them, matched to the nature of the venues in question. It should be noted that the themes were not pre-determined, but coalesced out of the body of work submitted. In this way, we feel that it is an interesting indicator of the directions that NIME artists are taking us.

For Monday's NIME Performances, the committee focused on Do-It-Yourself (***DIY***) instruments. This, in our view, is a new turn for the NIME community, and will be an interesting - if not controversial - topic of debate. During this evening, no pieces make use of traditional musical instruments. This decision did not come without soul-seeking on the part of the committee, and is a questioning that we hope will continue in inspired discussion during and after NIME 06.

Tuesday's program, "Special Guests" is the only program in the week not culled from the call for proposals. Instead, it is curated by IRCAM, and serves as a bridge to the Agora/Résonances festival within which NIME 06 takes place. Here we find the material that is at the foundation and history of NIME – extended acoustic instruments, responsive machines, and sensor-instruments. Two of us on the committee were happy to be invited to this program, putting to rest any conflict of interest issues.

With *Club NIME*, we hoped to create an official part of the NIME 06 artistic program that nonetheless kept the freshness of the alternative scene. The four Club programs take us to three of Paris' interesting independent venues and in doing so give us a tour of some of the dynamic neighborhoods of the city.

Monday at the Triptyque is an evening of ***hacks*** – be they hacked videogames, circuit-bending, or turntables turned upside down. We are very happy to have eRikm as headliner for the opening evening of Club NIME. His sonic explorations have broken out beyond the local scene, and beyond the turntable in his sonic explorations.

On Wednesday, Club NIME takes us up the Canal St. Martin to the Point Ephémère, a former concrete warehouse along the canal docks that was taken over by a group of artists in the 1990's who then transformed the space into artists' studios. The new venue marks a coming of age of the alternative scene in the form of a subsidized arts center complete with concert, gallery, and artists' residency spaces, restaurant, and café. It is here that Club II presents a unique evening of ***objects*** as instruments electronically re-piped trombones, huge glass disks, and responsive networked tables.

A gallery event on Thursday afternoon scales things back a bit. Ars Longa, partner in the Club NIME organization, is a multifaceted arts non-profit connecting community action and digital art. The gallery setting offers the intimacy to focus on ***acoustics***, be they the resonances of Tibetan bowls, a drum-playing robot, or the Doppler effects of spinning, unamplified mobile phone handsets.

The final program Thursday night brings us back to the Point Ephémère to close with a program that highlights the active ***local*** Parisian scene. From a technology standpoint, France represents for NIME an intense concentration of sensor interface and instrument developers. In Paris we count no less than four outfits - studios, companies, garage operations - creating interfaces for artists projects. This interest and activity in the area extends to local artists who create algorithmic sound/image systems, or who blend active careers in pop music with a taste for interaction.

I would like to thank the Performances and Installations review committees for their energy, through the occasionally grueling discussions, and for the host and partner organizations for making this very full artistic program possible. We hope that the NIME 06 artistic program will be stimulating and memorable. For those who want more, there's always the rest of Agora/Résonances!

Atau Tanaka
NIME 06 Performances and Installations Chair

Table of contents

Session Program, Monday, June 5

Paper Session, Monday morning, June 5

Mobile and Public

Poster Session, Monday, June 5

Sensors, Interfaces, and Mapping

Paper Session, Monday afternoon, June 5

Network and Collaborative

Session Program, Tuesday, June 6

Paper Session, Tuesday morning, June 6

Real and Virtual, Spatial and Graphical

Poster Session, Tuesday, June 6

Gesture Controlled Audio Systems

Poster Session, Tuesday, June 6

Tables and Screens

Paper Session, Tuesday afternoon, June 6

Instrument Design

Session Program, Wednesday, June 7

Paper Session, Wednesday morning, June 7

Brain, Hands, and Expression

Poster Session, Wednesday, June 7

Instruments, Performances, and Installations

Paper Session, Wednesday afternoon, June 7

Augmented Instruments and Accompaniment

Session Program, Thursday, June 8

Special Session, Thursday morning, June 8

Digital Interfaces for the Violin Family

Keynotes and Panel Discussion, Monday to Wednesday, June 5 - 7

Monday noon, June 5

Keynote George Lewis

Monday noon, June 6

Keynote William Gaver

Wednesday noon, June 7

Panel Discussion moderated by Michel Waisvisz

Workshops, Sunday, June 4, and Thursday, June 8

Workshop, Sunday, June 4

Workshop, Sunday, June 4

Workshop, Thursday, June 8

Performance Program

Concert at IRCAM (Espace de Projection), Monday evening, June 5

NIME Performances

Concert at the Triptyque, Monday night, June 5

Club NIME I

Concert at IRCAM (Espace de Projection), Tuesday night, June 6

NIME 06 Special Guests

Concert at the Point Éphémère, Wednesday night, June 7

Club NIME II

Performances at ArsLonga,Thursday evening, June 8

NIME Gallery

Concert at the Point Éphémère, Thursday night, June 8

Club NIME III

Installation Program

papers and posters

Mobile Music Technology: Report on an Emerging Community

Lalya Gaye
Future Applications Lab
Viktoria Institute
417 56 Göteborg, Sweden
+46-703800134
lalya@viktoria.se

Lars Erik Holmquist
Future Applications Lab
Viktoria Institute
417 56 Göteborg, Sweden
+46-703558500
leh@viktoria.se

Frauke Behrendt
Media and Film Studies
University of Sussex
Brighton, BN1 9 RH, UK
+44-7840148789
f.behrendt@sussex.ac.uk

Atau Tanaka
Sony CSL Paris
6, rue Amyot
F-75005 Paris, France
+33-1-44080512
tanaka@csl.sony.fr

ABSTRACT
The new field of mobile music emerges at the intersection of ubiquitous computing, portable audio technology and NIME. We have held a series of international workshop on this topic with leading projects and speakers, in order to establish a community and stimulate the development of the field. In this report, we define mobile music, and map out the field by reporting on the workshop series and accounting for the state-of-the-art.

Keywords
Mobile music, locative audio, mobile computing, ubiquitous computing, locative media, new interfaces for musical expression.

1. INTRODUCTION
Mobile music is a new field concerned with musical interaction in mobile settings, using portable technology. It goes beyond today's portable music players to include mobile music making, sharing and mixing. The core themes of NIME – interaction, interfaces and music, can today be deployed on mobile electronics. While NIME projects have mostly been concerned with stationary concert performance or installations, mobility allows NIME concepts to occupy exterior urban space, and exploit people's movements through it, as well as the heterogeneous space and social dynamics found in those environments. The *International Workshops on Mobile Music Technology* are the first events that focus on this new field. They have played a key role in the development of mobile music since 2004 and can be regarded as one direction for expansion of the NIME community. This report establishes a definition of mobile music, describes the workshop series, and accounts for the state-of-the-art.

2. MOBILE MUSIC TECHNOLOGY
A number of recent technological advances have pushed the envelope of possible human-computer interactions, giving rise to new fields such as locative media and pervasive gaming. Mobile computing enables systems to be used anywhere and on the move. Coupled with context-aware computing and global positioning technology, mobile devices can respond to the user's surroundings and location, situating interaction within everyday settings. Augmented and mixed-reality technologies merge the digital and physical realm, making them cohabit in the same environment. Ad hoc, peer-to-peer and distributed networking allow groups of users – ranging from co-located strangers to dislocated friends – to collaborate spontaneously, across distances and without the need for centralised supervision. Meanwhile, the miniaturisation of consumer electronics and improvements in high-capacity digital storage have given rise to powerful portable MP3 players that could easily contain one's complete music collection. Moreover, mobile phones have brought us ubiquitous network connectivity. Mobile music emerges at the crossroads of these technological advances and their resulting new practices, joining the worlds of ubiquitous computing and locative arts, with mobile consumer electronics, and the sensing and interaction tradition of NIME.

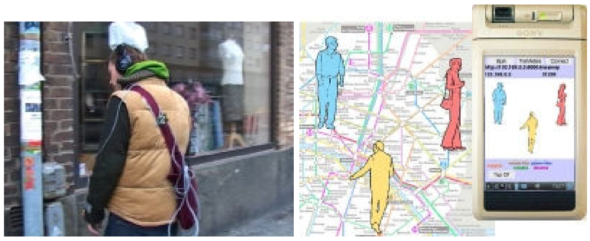


**Figure 1: Mobile music projects:
a) Sonic City; b) Malleable Mobile Music.**

2.1 Mobile Music: Beyond Portability
Mobile music as a term covers any musical activity using portable devices that are not tethered to a specific stationary locale; in particular those where the activity dynamically follows users and takes advantage of the mobile setting, thereby leveraging novel forms of musical experience. Mobile music devices might possess properties such as context awareness, ad hoc or distributed network connectivity, or location sensing, sometimes combined with technology embedded in the physical environment. Therefore, they can be used anywhere and on the move, and take advantage of people's displacements, location, and of the changes of social and geographical context that mobility implies. Examples of mobile music activities include pushing music to people nearby [16], sonifying local Wi-Fi coverage while riding a bike [19], or remixing music tracks with remote friends across peer-to-peer networks [27]. Mobile music goes beyond the iconic Walkman™, and does not need to imply individual use, headphones or passive music listening. It spreads over a large spectrum of musical interactions, ranging from consumption to creation, and with mobility increasingly blurring this distinction [11]. Mobile music resonates with practices of both NIME musicians and everyday users of consumer audio products.

2.2 Reconsidering Musical Interaction with Mobility

Mobile music creates a tension between music and place as well as new relationships between musician, listener, and music. For electronic musicians, the mobile environment offers more than just a new place to transplant NIME techniques. Rather, mobility encompasses specificities that encourage us to reconsider the basic tenets of musical interaction. Mobility implies outdoor environments where space and place become tangible parameters, and also implies always-on itinerant devices: location can become a "sensor" input to music systems, people nearby can become part of an ad hoc networked musical performance. With networked multi-user systems, mobility allows musical engagement beyond eye-to-eye contact. It also asks the NIME musician to consider social aspects in everyday public space, an environment not primarily dedicated to music use and where people might already be involved in a number of adjacent and simultaneous activities.

2.3 Another Dimension to Creative Engagement with Consumer Products

Many mobile music projects draw on earlier popular electronic music movements such as remix- and DJ-culture, file-sharing or playlists. They extend creative ways of engaging with portable consumer audio technology by weaving them into ever-changing geographical and social contexts. One example is tunA [4], where people in close proximity can share music by listening to each other's MP3 playlists, getting a taste of people's musical preferences across various social situations. There is a broad range of possibilities in terms of making music with mobile consumer devices, from ringtones, to mobile soundscape recording or sound art. Widespread platforms such as mobile phones are used as musical instruments and interfaces, encouraging the public to explore new ways of looking at their personal mobile devices. Projects working with such communication technologies invite musicians and lay people alike to participate in performances, group improvisation, sound art or remixing, for example collaborating with strangers in same physical space (e.g. on the bus), or jamming with remote friends while strolling around town.

3. INTERNATIONAL WORKSHOPS

At its early stage, mobile music was rapidly gaining popularity and relevance but lacked a clear sense community and an explicit demarcation as a field. As for any emerging field, it was therefore important to establish a community of people who could share experiences and communicate ideas. A good way to achieve such a goal is focused workshops. For instance, the NIME conference series grew out of a workshop at CHI 2001 [23]. In 2004, we started a series of international workshops on the subject in order to establish and develop the field of mobile music. The workshops have gathered a mix of researchers, designers, musicians, new media artists, and representatives of the industry [15]. They have raised awareness about existing projects as well as helped actors of the field with backgrounds in multiple disciplines to identify common goals and issues, share resources, and introduce one another to relevant technologies, methods and concepts. The purposes and formats of the workshops have varied as the community evolves but activities in common include presentations of projects, in-depth discussions, brainstorming sessions and hands-on activities.

The first *International Workshop on Mobile Music Technology* was organised at the Viktoria Institute in Göteborg, Sweden, in June 2004. The purpose was to gather a number of researchers with a shared interest in mobile music, and to attract additional people who might be interested in making the community grow. This workshop focused on presenting existing projects and defining the field. It had 15 external participants, plus organisers and student volunteers.

The second workshop was organised in May 2005 at NIME 2005 in Vancouver, Canada. This time, the community was better defined, and the workshop time was shared between presentations of new projects, in depth-discussions and hands-on brainstorming activities. It attracted 18 external participants.

The third edition of the workshop was a two-day event that took place in March 2006 at the University of Sussex, Brighton, UK. It gathered nearly 30 participants and focused on the locative media aspect of mobile music, with presentations by invited speakers, feedback sessions about work-in-progress projects, and hands-on activities with the latest mobile music technology.

3.1 Projects

The workshops feature state-of-the-art mobile music projects, in the form of presentations by guest speakers and peer-reviewed papers, posters and demonstration sessions (see Figure 2.a), as well as feedback sessions for works-in-progress. These projects are at the centre of the field's development and demonstrate its diversity and potential. Many use generic mobile platforms such as mobile phones or handheld computers; others use hacked or custom-made technology to better respond to specific needs and requirements. All have in common taking advantage of the mobile nature of mobile technologies and situations as an intrinsic part of their work. Projects can be grouped along the following emerging themes.

Several projects explore *collaborative music making* with mobile technology. Malleable Mobile Music [28] (see Figure 1.b) is a location-based and peer-to-peer networked remixing system. TGarden [24] is an interactive environment for theatrical music making using wearables. Sequencer404 [17] allows multi-user control of a musical sequencer through telephony and Voice over Internet Protocol (VoIP). In CELLPHONIA [8], people engage in a location-based mobile phone karaoke opera. The collaborative public art performance China Gates [10] synchronises a set of tuned gongs with GPS as participants follow different routes. Mobile phones are used for interacting with a sound installation in Intelligent Streets [18]. Finally, IMPROVe [29] is a architecture for collaborative improvisation with sounds recorded with mobile devices.

Some of the projects in the genre of *mobile music making* enable individual users to manipulate sounds and create music by *interacting with environmental factors*: the physical urban environment with Sonic City [12] (see Figure 1.a), and ambient lighting conditions in Solarcoustics [3]. A mobile user-interface platform for such interactions in a personal area network (PAN) was also demonstrated [30].

Another theme is *mobile music listening and sharing*. Some projects address the sharing of playlists and music across peer-to-peer networks, enabling users to listen to their neighbours' music either synchronously (SoundPryer [21] and tunA [4]) or asynchronously (Push!Music [16]). Other projects transform music albums into narratives spread across geographical space (Location33 [9]), or enable the cultivation of public "sound gardens" located in Wi-Fi connections, as an overlay on physical space (Tactical Sound Garden [TSG] [25]).

A third area is dedicated to *HCI and mobile music*. It includes SonicPulse [2] – a project providing an acoustic way of passively monitoring or actively exploring a shared music space, Music Mood Wheel [1] – an auditive interface for navigating music spaces on the move, and Minimal Attention Navigation via Adapted Music [14] – a musical navigation system for pedestrians.

Meanwhile, some workshop participants have taken a more sociological or media-studies approach, looking at the relation between music taste, use and identity [11], soundscapes and people's everyday experience of place [22], mobile phones and its use in sound-art [5], and mobility, sound and urban culture [7]. These contributions have brought insightful humanistic perspectives for the development of the field of mobile music, grounding it on social realities, aesthetics and already emerging practices.

3.2 Group Activities

The workshops included group activities with structured brainstorming sessions in the first two workshops, as well as feedback sessions and hands-on experience of mobile music technology in the third one. These were combined with in-depth discussions on various topics relevant to mobile music and on current issues, opportunities and challenges in the field – e.g. the relationship between mobile devices, space and the body in movement, or how to approach context-aware platforms developed in the field of ubiquitous computing with a NIME perspective.

3.2.1 Brainstorming Sessions in the 1st Workshop

An important function of the first workshop was to map out the field and define future directions. We organised a series of structured brainstorming activities that ran over two days. Participants were divided into three groups, each dedicated to one of the following topics: *mobile music creation*; *mobile music sharing*; *business models and the future of the mobile music industry*. In addition to this, we had pre-defined a number of themes to investigate: *infrastructure and distribution*; *genre and formats*; *social implications*; *ownership*; *business models*; *creativity*; *interaction and expression*; *mobility*; *users and uses*.

Each group would choose four themes from the list, and discussed them from the perspective of their overall topic. For instance, the group on *Business models* discussed the theme *Genre and format,* raising issues such as length of compositions, use of meta-tags or potential revenue from different kinds of formats. After the first day of brainstorming, results were presented to the other groups. The second day was dedicated to defining design dimensions for mobile music applications based on day one's sessions (for example solo vs. collective, foreground vs. background), and to mapping the emerging design space to existing or future projects. The sessions raised a number of issues, including "in-between" states that are neither mobile nor stationary, how musical taste is used to establish personal identity, to the meaning of ownership and where added value could be elicited.

3.2.2 Bodystorming Session in 2nd Workshop

In the 2nd workshop, hands-on activities were kept to one afternoon. They focused on bodystorming of mobile music applications and scenarios. Bodystorming is a method where participants act out a particular scenario of use, taking the roles of e.g. users or artefacts and focusing on the interaction between them [6]. With this method, participants explored various mobile music themes, developed simple application ideas, and physically enacted scenarios of use in order to get an embodied understanding of design challenges and opportunities specific to mobile music.

Figure 2: Mobile music workshops: a) Project demonstration and b) scenario bodystorming

Participants first combined randomly chosen instances of the following categories: *situations* (e.g. driving a car while it snows); users (e.g. school kids); *technological infrastructures* (e.g. Wi-Fi, GPS); *types of music uses* (create, share, organise...). Combinations were assigned to each group and developed into 3 application or scenario ideas per group during short brainstorming sessions. Each group decided on one idea and further developed it through bodystorming. Scenarios were then acted out to the rest of the workshop to stimulate discussion. An example of scenario was a bicycle-taxi working as a peer-to-peer server and broadcasting its clients' music on loudspeakers in Kingston, Jamaica (see Figure 2.b). This scenario generated discussions on mobile ways of sharing and outputting music in public space, and of their social adequacy.

3.2.3 Feedback Sessions and Hands-On Activities in the 3rd Workshop

On the first day of the third workshop, selected work-in-progress projects grouped in parallel sessions received expert feedback during critical and supporting discussions. Through this participants identified crucial issues and presented their findings to the other groups. The second day was hands-on and gave participants access to technologies for mobile music that they might otherwise not have been exposed to. Participants were given tutorials on sensors for mobile music, and on miniMIXA [20], a mobile music software mixer and mini recording studio for hand-held devices. They were also introduced to socialight [26], an audio space annotation platform for sharing location-based media. As a follow-up, participants sketched out possible applications combining such technologies.

3.3 Dissemination of Results

The output from the workshops has been presented in contexts outside of NIME and of the workshop itself. Two of the co-authors moderated a panel discussion at the ACM SIGGRAPH 2005 conference on the subject of *Ubiquitous Music* [13] where the majority of the panellists selected by the conference had previously participated in the workshops. As the largest international conference on digital media and emerging technologies, the SIGGRAPH panel underscored the pertinence of mobility and musical interaction to a wider field. Authors have also given lectures about mobile music in art and design schools. Currently, the results from the first workshop are being edited into a book that will be a key reference emphasising the creative potential of mobile music technologies.

4. CONCLUSIONS AND FUTURE WORK

We have presented the field of mobile music, its current state-of-the-art, as well as a workshop series with a decisive influence on its development. During the workshops, a multitude of emerging key topics concerning the socio-cultural, artistic, technological and economical aspects of mobile music have been identified. The overall experience of these events has been very positive. Out of the participants has crystallised a core group, which is very active in the field. The community continues to grow, with new people being attracted to each workshop and the number of relevant projects increasing consistently. The future of mobile music is now being shaped by a collective community effort and promises interesting future developments. In order to further extend and consolidate the mobile music community and support these developments, we will continue to organise new workshops and will soon publish a website as a resource for mobile music projects and related publications.

5. ACKNOWLEDGMENTS

The 1st workshop was organised by the Viktoria Institute, the 2nd by Viktoria and Sony CSL Paris (in conjunction with NIME'05), and the 3rd workshop was a collaboration between Viktoria, the Universities of Sussex and Salford, the Pervasive and Locative Arts Network (PLAN) and Futuresonic. We wish to thank our workshop co-organiser Drew Hemment, the NIME'05 chairs Tina Blaine and Sidney Fels, as well as the mobile music core group, including among others Arianna Bassoli, Gideon d'Arcangelo, Maria Håkansson, Rob Rampley, Chris Salter and Mattias Östergren. We also warmly thank all the workshop participants, reviewers and student volunteers, for making this series of events successful.

6. REFERENCES

[1] Andric, A., Xech P.-L. Music Mood Wheel: Auditive Interfaces for Mobile Music Devices. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[2] Anttila, A. SonicPulse: Exploring a Shared Music Space. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[3] Barnard, M. Solarcoustics: CONNECT. *2nd International Workshop on Mobile Music Technology, NIME'05,* Vancouver, Canada (2005).

[4] Bassoli, A., Moore, J., Agamanolis, S. tunA: Synchronised Music-Sharing on Handheld Devices. *Adjunct Proc. of Ubicomp'04*, Nottingham, UK (2004).

[5] Behrendt, F. Mobile Phone Music. Sound Art and 'Mobile Devices'. *1st International Workshop on Mobile Music Technology*, Göteborg, Sweden (2004).

[6] Buchenau, M., Fulton Suri, J. Experience Prototyping. *Proc. of DIS'00*, ACM Press (2000).

[7] Bull, M. *The World According to Sound: Investigating the World of Walkman Users. New Media and Society*. Sage London (2001).

[8] Bull, S., Gresham-Lancaster, S., Perkis, T. CELLPHONIA: In The News / Work-In-Progress. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[9] Carter, W., Liu, L. S. Location 33: A Mobile Musical. *2nd International Workshop on Mobile Music Technology, NIME'05,* Vancouver, Canada (2005).

[10] Clay, A., Majoe, D. China Gates. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[11] D'Arcangelo, G. The New Cosmopolites: Activating the Role of Mobile Music Listeners. *2nd International Workshop on Mobile Music Technology, NIME'05*, Vancouver, Canada (2005).

[12] Gaye, L., Mazé, R., Holmquist, L. E. Sonic City: The Urban Environment as a Musical Interface. *Proc. of NIME'03*, Montreal, Canada (2003).

[13] Holmquist, L. E., Tanaka, A. Ubiquitous Music: How Are Sharing, Copyright, and Really Cool Technology Changing the Roles of the Artist and the Audience? *SIGGRAPH 2005*, Los Angeles, USA (2005).

[14] Hunt, R., Apperley, M., Cunningham S. J., Rogers B., Jones, M. Minimal Attention Navigation via Adapted Music. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[15] International Workshops on Mobile Music Technology http://www.viktoria.se/fal/events/mobilemusic/

[16] Jacobsson, M., Rost, M., Håkansson, M., Holmquist, L.E. Push!Music: Intelligent Music Sharing on Mobile Devices. *Adjunct Proc. of UbiComp 2005*, Tokyo, Japan (2005).

[17] Jimison, D., Thatcher, T. Sequencer404: A Networked Telephonic Composer. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[18] Lörstad, H., d'Inverno, M., Eacott, J. The Intelligent Street. *1st International Workshop on Mobile Music Technology*, Göteborg, Sweden (2004).

[19] McCallum, D. N. G. *Warbike - A Mobile Artistic Sonification of Wireless Networks*. Master's thesis in Art and Technology, IT University of Göteborg, Sweden (2005).

[20] miniMIXA. http://www.sseyo.com/products/miniMIXA/

[21] Östergren, M., Juhlin, O. Sound Pryer: Truly Mobile Joint Listening. *1st International Workshop on Mobile Music Technology*, Göteborg, Sweden (2004).

[22] Phillips, A. Composing the Soundscape: Re-Engaging with Place. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[23] Poupyrev, I., Lyons, M. J., Fels, S., Blaine, T. New Interfaces for Musical Expression. *Extended Abstracts of CHI '01*, Seattle, USA (2001).

[24] Ryan, J., Salter, C. TGarden: Wearable Instruments and Augmented Physicality. *Proc. of NIME'03*, Montreal, Canada (2003).

[25] Shepard, M. Tactical Sound Garden [TSG] Toolkit. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[26] Socialight. http://socialight.com/

[27] Tanaka, A., Tokui, N., Momeni, A. Facilitating Collective Musical Creativity. *Proc. ACM Multimedia*, Singapore (2005).

[28] Tanaka, A. Mobile Music Making. *Proc. of NIME'04*, Hamamatsu, Japan (2004).

[29] Widerberg, R., Hasan, Z. IMPROVe: Mobile Architecture for Sonic Socio-Cultural Exchange. *3rd International Workshop on Mobile Music Technology*, Brighton, UK (2006).

[30] Yamauchi, T., Iwatake, T. Mobile User Interface for Music. *2nd International Workshop on Mobile Music Technology*, Vancouver, Canada (2005).

A Framework for Spatial Interaction in Locative Media

Atau Tanaka
Sony Computer Science Laboratory Paris
6 rue Amyot
F-75005 Paris FRANCE
+33-1-4408-0512
tanaka@csl.sony.fr

Petra Gemeinboeck
Key Centre of Design Computing and Cognition
University of Sydney
NSW 2006 AUSTRALIA
+61-2-9351-3030
petra@arch.usyd.edu.au

ABSTRACT
This paper presents the concepts and techniques used in a family of location based multimedia works. The paper has three main sections: 1.) to describe the architecture of an audio-visual hardware/software framework we have developed for the realization of a series of locative media artworks, 2.) to discuss the theoretical and conceptual underpinnings motivating the design of the technical framework, and 3.) to elicit from this, fundamental issues and questions that can be generalized and applicable to the growing practice of locative media.

Keywords
Mobile music, urban fiction, locative media.

1. INTRODUCTION
This paper presents the concepts and techniques created to realize a body of location sensitive mobile media artworks. They are installation works extending beyond the confines of a gallery, to include the urban environment. Conceived for participative use from advanced mobile telephones, they call upon techniques from interactive music applied to new contexts that include the interplay of sound and image, an exchange between participants in the gallery and participants in the streets, and the creation of an abstract narrative from audiovisual media captured on multiple mobile devices.

The paper is organized as follows. In section 2 we describe related works, both by the present authors, as well as others in the field. Section 3 introduces the notions of spatial interaction and locative media that form the conceptual basis of the work and section 4 outlines a typical usage scenario. Section 5 describes a generalized system architecture. The system is ultimately a framework that can be adapted to a series of works in the area. This framework is able to service various embodiments, including the scenario described in Section 3. Section 6 closes with discussion of the first installation work created using the framework.

2. RELATED WORK
The authors have each separately produced prior work that leads up the present collaboration. Tanaka has published on and demonstrated a collaborative mobile music system [18]. Gemeinboeck has exhibited and published installation works in which abstract threads of memory interweave mediated and physical spaces [5].

Related work in the area of mobile music systems can be seen in the projects presented at international events like the Mobile Music Workshop series [11]. Recently, new fields of creative works known as *locative media* are gaining interest. Work in this area is represented in research consortia such as Open PLAN [8], and in festivals such as PixelAche [13]. One of the themes for the 2006 edition of the International Symposium of Electronic Art (ISEA 2006) is the *Interactive City* [10].

Recent artistic inteventions in the field of *locative media* include the annotation of the physical with virtual marks or narratives [14] or the revelation of patterns by tracing the participants' movements through the environment [16]. They tend to take a documentalist approach to integrating geography as a component in artistic practice.

3. CONCEPTS
3.1 Spatial Interaction
The work seeks to establish a connection across and coherence between two distinct types of spaces – urban space, and a gallery exhibition space. The latter is typically characterized as a "white box", controlled environment with artificial lighting to present cultural artefacts. Everything is done to allow concentration and minimize distraction in a gallery. An urban environment, on the other hand is an uncontrollable multiplicity of independent activity. Typified by the term, "concrete jungle", it is characterized by chaotic activity, and multifarious events, indications, and communication that simultaneously make demands on our attention.

3.2 Locative Media
The deployment of mobile, networked, location-aware computing devices, involving participants in mapping processes, social networking or artistic interventions is often associated with the emergent field of locative media. Considering geographical space to be a canvas that allows the inscription of personal narratives, desires, and memories, offers a powerful instrument for communities to (co-)author their environment and to collectively organize and share such subjective experiences. There is, however, a double-edge to the attempt of inversing power strategies of remote control, as the technologies affording these collective location-based interventions paradoxically operate upon the same plane as surveillance. Its reliance on positional precision and location-based context critically link locative media to the arena of cartography and its dominant practices of mapping. And, in Irit Rogoff's words, "mapping as a cultural, political and epistemological activity is deeply imbricated in nations' narratives of their own formation" [15]. This ambivalent notion, of course, bears the potential of opening up a *third space* [1], a space for intervention, in which these power relations can be investigated and negotiated.

3.3 Geography as Musical Interface

This paper seeks to describe location-based techniques in the context of interactive music and multimedia. Interactive music has until now primarily focused on the use of sensor systems as the underlying technologies used to build novel interfaces and new musical instruments. Here we describe the use of geographic localization in this light. As the system is multi-user, it is not only location that enters the system as musical parameters, but the community interactions that take place amongst the users of the system. While prior work by the authors has demonstrated the use of social dynamic to generate musical re-mixes, here we describe a generalized framework that can support the creation of a body of work that is location aware, multi-user, and multimedia.

4. SCENARIO

4.1 The Players

There are two types of participants corresponding to the two spaces the work inhabits: the outdoor mobile environment, and the indoor gallery environment. Players outdoors are armed with specialized mobile phones loaned out to participants. The indoor participants not only experience the resulting audio/visual projections in the gallery, but interact with the piece and the other participants through touchscreen interfaces on pedestals in the gallery space.

By spanning two disparate environments with a single system, we create a 'third space' that is spanned by three mobile participants, three gallery participants and server. The interconnections correlate, process and communicate actions in each domain of this shared space, resulting in the community media dynamic that is the output of the system.

4.2 Mobile Participants

A series of advanced mobile telephones are made available to gallery visitors who are encouraged to undertake a 'mission' in the neighborhood surrounding the gallery. The mobile phone functions as a music listening device with Walkman headphones, and with its built-in video camera, as a portable video camera.

Depending on the specific manifestation, or use of the framework in a specific piece, the *motivation* may take on different forms. This may be game-like, with specific targets as in a treasure hunt, more social, as in orchestrating personal encounters through sound/image avatar mediated representation, or be more abstract or even aimless. The framework supports all these modes, the specific embodiment will define the participants' *tasks* through a kind of community choreography.

4.3 Gallery Participants

The gallery space is arranged to be a sort of "mission command central". The recomposed collage of incoming images from each mobile camera and processed and mixed sounds are projected on a large-scale screen and in the sound system. They display the ongoing weaving process in relation to the path that is invisible for the participants outside the gallery.

The gallery participants are not just passive viewers, but are able to interact with the mobile participants via three small touchscreen interfaces. Made available on pedestals in front of the main projection screen, each touchscreen corresponds to one mobile phone out in the field. The touchscreen display is similar to the mobile phone display, but larger, having greater functionality, and giving a higher level view of the system. While the display follows the camera image received from its corresponding mobile phone (surveying the surveyor…), it also shows the relation of the current path and the one it seeks to follow (the trail of a previous participant). Allowing the gallery participant some level of control, they can choose to highlight or obscure relations before they are interpreted and sent to the participant outside. These manipulations not only contribute to the progress of the *mission*, but are also displayed on the large screen and sound system.

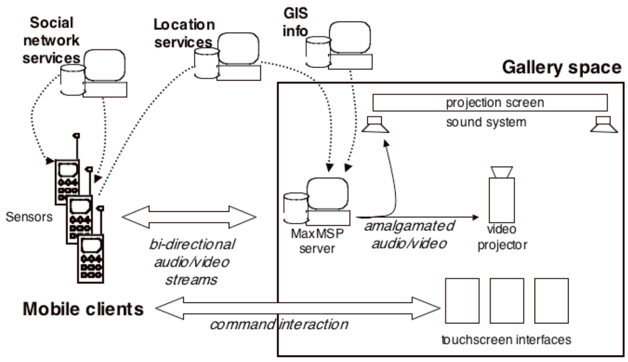


Figure 1. System Overview

4.4 Data Aggregation

The system server aggregates the different audio video sources from the mobile devices, and processes them according to control information coming from the different participants and various external sources.

Mobile participants are able to interact in a crude fashion using the 5-way control buttons on their mobile phone. In a more advanced implementation, sensors onboard the mobile devices can give more fluid information on the gesture and movement of the mobile user. While these various control data can directly affect the audio/visual display local to their device, their primary use is to be recovered by the system as a series of XML messages that alter that participant's part within the aggregated whole on the server.

Likewise, gallery participants actions on the touchscreens serve not only to send instructions to their corresponding mobile user, but also are sent up to the server to contribute to global media processing. The exact processes and mappings depend on the specific implementation.

External control sources include geographic location of each mobile device. While existing systems like the Global Positioning System (GPS) allow accurate location tracking, they necessitate additional hardware. In the current framework, we make use of mobile phone' GSM cell antenna signal reception strength mapped to mobile operators' databases of antenna location to deduce an approximate location. A Location Based Services (LBS) server provides the gateway between the measured antenna signal, the operator database, and the framework server.

Other external information includes Geographic Information Systems (GIS). This allows recovering demographic information correlated to mobile client position. Socio-cultural embodiments of the framework make use of this data channel to extend the *third space* afforded by the framework to include socio-economic data that is location specific.

4.5 System Output

Based on aggregated media and associated control information, the system server amalgamates the disparate media elements, processing, combining, mixing, distorting them according to rules and mappings defined in each specific embodiment.

The result is a series of audio/video streams sent out to several destinations. The destinations include returns to the individual mobile devices and in-gallery touch screen displays. The primary destination is the main video and sound projection in the gallery, representing the sum output of the system. The media being streamed out to the mobile devices and touchscreens can be individually tailored, or may mirror the main output.

Sound output is rendered based on the interaction elements coming in from the control channels. Two types of sound are employed: pre-composed musical tracks, and live environmental recordings. Each mobile user may have selected a music track as their "soundtrack" for their walk. Meanwhile, the microphone on the mobile phone serves to provide live sound capture of the outdoor environment. These elements are correlated to similar elements from the other users, as well as history of previous users experiences to create a collage mix. Depending on the state of the user's sensors or location, as well as those of the others, the final sound output will be an evolving mix of deconstructed musical source elements injected with environmental noises, actively panning and substituting elements from other users and other moments while preserving a coherent sonic flow.

5. ARCHITECTURE

The technical infrastructure of the piece supports both classical client-server architectures as well as peer-to-peer like end-to-end communication. The project server is situated in the gallery space. There are three kinds of server side functions: audio/video (a/v) input/output (i/o), message services, and location services. There are two types of clients: the mobile devices and the small touchscreens in the gallery.

5.1 Clients

5.1.1 Mobile Terminal

The client is a mobile telephone with multimedia capabilities. It has a network connection over 3G (or UMTS) packet switching mobile data infrastructures with a bandwidth throughput of up to 256 kbps.

The terminal is able to operate as an audio and video capture device, streaming live audio-video streams up the wireless network to the server in the gallery. It is also a sound/image display device, with MP3 audio decoding played over standard Walkman headphones, and with a 240x320 color graphics display. The client is able to read audio and video data streams from the network to display in real time. A custom Java program was developed, and has been downloaded to the phone's memory prior to the exhibition.

The user interacts with the system via a 5-way joystick-button system and two programmable soft-buttons. The phone folds switchblade style, concealing the numeric telephone keypad, but leaving the camera, display, audio jacks and interface buttons apparent. This configuration allows the phone to lose it's telephone associations and be used as a nomadic a/v device. (Fig. 2.)

While the primary mode of communication for the mobile client is with the project server, there is the possibility of direct peer communication amongst the mobile devices. A profiling system is used to construct a social networking map when each mobile device joins the system. The social network defines the situations and conditions in which mobile devices would enter into direct communication. This in conjunction with the location services could for example aid two mobile users to cross paths if nearby.

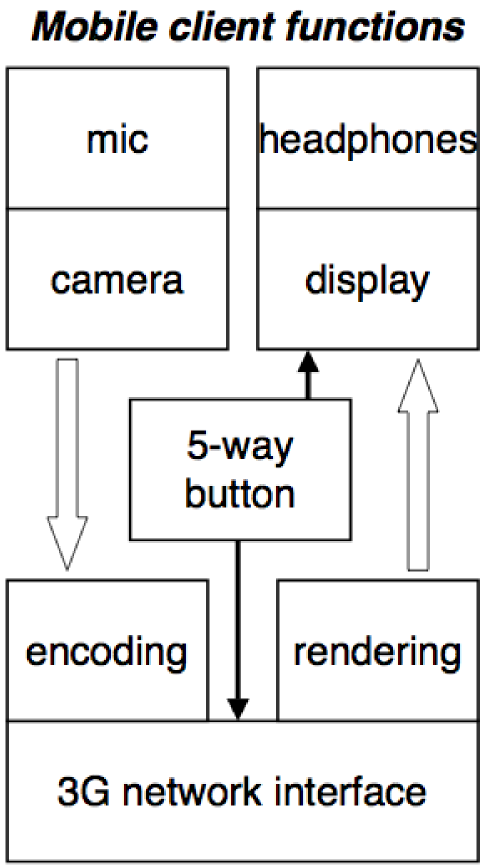


Figure 2. Mobile Client Architecture

5.1.2 In-gallery Touch Screen

The stationary clients are 12.5cm diagonal 768x1024 pixel handheld computers. They are connected via 802.11g (WiFi) wireless networking to the server. One touchscreen interface corresponds to each mobile phone out in the field. The user is able to manipulate the graphic user interface (GUI) on the touchscreen with his fingers. These in-gallery client devices serve as mediators between the mobile terminals and the final gallery audio/video projections.

5.2 Servers

5.2.1 A/V Server

The server is programmed in MaxMSP/Jitter with several ancillary components. It is able to receive parallel audio/video upstreams from each of the three mobile phones.

The server has three distinct a/v output destinations: the mobile clients, the gallery clients, and the main projection display.

Audio to the mobile clients is sent up to an Icecast server [9] running on the same machine that relays the audio stream to the mobile phones as an MP3 live stream.

The incoming video from the mobile phones is mirrored to each corresponding in-gallery client over the LAN via http. The video stream is consolidated and encoded as an H.263 streamed over HTTP to the mobile clients.

The monitor output as well as audio output of the server machine are connected to a local projection system in the gallery. This is the final combined output of the system, sent to a video projector and sound system.

5.2.2 Messaging Server

It also receives control input resulting from the mobile user's actions on the telephone buttons, as well as the gallery participants' activities on the touch screens. These control message enter as OSC messages translated to XML over UDP and TCP socket connections.

5.2.3 Location Server

Location based services are provided as a middleware layer server side between the mobile phone operator and the gallery. Each mobile client can be tracked geographically to a precision determined by the distance between two cell antennas. This location data is aggregated at the gallery server as one of the visual/musical parameters.

5.2.4 Social Network Server

A user profile system allows for recording of user preferences and user-specific metadata. While on commercial systems, this kind of data would typically be used for personalization to feed recommendation engines, in the case of the present framework, the data is used as a means through which different users may enter into direct contact. This allows bootstrapping of a social network overlay to the system where principles from social computing are used to make connections between direct acquaintances, friends of friends, and users of similar profiles. This provides the peer-to-peer communications channel that runs parallel to the client-server media architecture predominant in the framework.

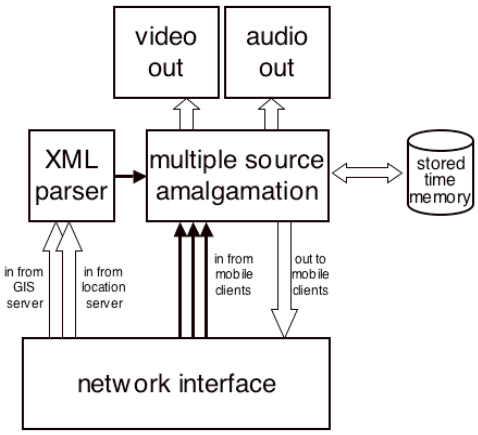


Figure 3. Server Architecture

6. DISCUSSION

6.1 Idiomatic Locative Media

We have described the architecture and general usage scenario of a hardware/software framework that is not specific to one artwork, but that should be useful in the creation of a series of musical or audio-visual works that seek to tap the dynamic of locative media. The question arises about the true generality of the framework, and idiosyncrasies that may influence the resulting works.

The motivation was to create a framework detached from a specific creative embodiment much in the way that a musical instrument has an independent existence and identity from the pieces that are composed for it. A musical instrument, while separate from a work for that instrument, however, does exhibit specificities and idiosyncrasies. Composers who successfully exploit these idiosyncrasies, we say, have found the *voice* of the instrument, and have written music *idiomatic* for that instrument [20]. An instrument can foster a repertoire, or a body of work, that while stylistically diverse, remain identified with that instrument.

We sought a similar goal with the present framework. We did not want the framework to be tied down to needs or peculiarities of a single piece. At the same time, the construction of the framework represents the views of the present authors on issues of locative media. By building a *locative media instrument* of sorts, we hope that others may benefit from the efforts in creating the system to realize other, related works. The notion that we have created a platform even for ourselves to realize more than one piece work is attractive. The resulting pieces, whether done by us or by other composers and artists, may have the indelible identity that associates it with the framework described here, but from the idiomatic composition perspective, this is not seen as a negative quality.

In the following sections we describe some of the theoretical underpinnings that motivated us to construct the framework in the way we did.

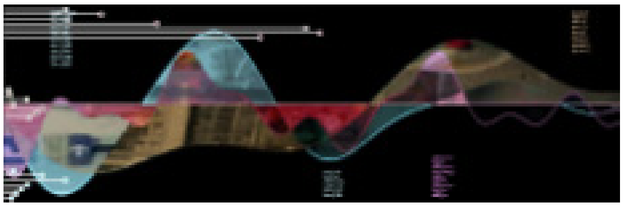


Figure 4. Example of generated imagery

Figure 5. Image displayed on mobile client

6.2 Memory and the Situationist Dérive

The participants' mission is to uncover the traces of earlier participants, that is to say that they are using their mobile instruments to look for and to follow the invisible trails of now past participants and the traces captured along these paths. The scenario reminds of a notion of Benjamin's *flaneur*, in which they put themselves in the shoes of someone else in order to emplace themselves in the city. [2] Only that the observed other is no longer present. The city in the 'eyes' of the device then composes and decomposes itself in relation to the invisible path. As both current and stored positions and images are compared on the server, each participant, similar to the game 'Hot and Cold.' receives indications about their closeness to the path. These visual and acoustic indications, however, can be deceiving as the result of these relational processes can be obscured or manipulated by the gallery participants. Participants, however, don't only leave and receive traces but can also choose to conspire and to co-navigate themselves through the neighborhood. The path reveals itself through puzzle pieces that appear on the screen, weaving threads of the past into the urban present.

6.3 Surveillance

The digital revolution has helped make possible the surveillance society as predicted by Orwell and Huxley [12, 7]. While governments put in place networked capability to survey with advanced image analysis techniques all in the name of security, few today decry Orwell's 1984 and Big Brother except by anecdote. Why?

Perhaps one explanation is that as governments have gained surveillance powers, so has the public. The omnipresence of consumer devices camcorders has created a revolution in *grassroots journalism*. Beginning with the Rodney King video in 1991 denouncing police brutality [6], amateur video has often been the first camera on the spot, be it for the tsunami in Southeast Asia, or the crash of the Concorde [3].

The question of balancing of power, or tension between powers, respectively, and its potential inversion of observer and observed create the second dynamic we wished to put in place in the work.

6.4 Interrelations

The server, storing and processing the trails and traces of the participants, of course, appears to have the ultimate 'knowledge' and thus control over the relations of the individual participants. The tension unfolds when the gallery visitors use their 'control power' to manipulate the sets of relations or the mobile participants, in turn, conspire with one another.

The process of detecting and negotiating the invisible paths not only relies on the instrument's positions but also deploys pattern and image recognition strategies. Analogous to the participants' experience of seeking the invisible trail, the underlying control process likewise becomes a matter of relations and associations, rather than a one-to-one correspondence of geographical positions (coordinates).

7. CONCLUSIONS

We have described a framework for the creation of musical and audio-visual artworks that extend outside of the concert hall or gallery space to embrace and include the surrounding city. The system is conceived as a semi-generalized framework, much in the way of a musical instrument. We have described the qualities and structure of the instrument, and have defended the architectural decisions by a theoretical stance with respect to location-based interaction.

While the conceptual structure calls upon cultural theory, the modes of interaction draw directly upon techniques that are current in the fields of interactive music, media art, and social computing. This provides the composer or artist the possibility to extend their craft beyond the confines of traditional cultural venues to include the dynamics of the urban environment.

8. ACKNOWLEDGMENTS

We would like to acknowledge the participation of Clicmobile S.A. for their gracious contribution of location and social network services. This work has been funded in part by a grant from the RIAM, the French Minister of Industry.

9. REFERENCES

[1] Bhabha, Homi, K. *The Location of Culture*. Routledge, London, 1994.

[2] Bull, M. "The World According to Sound: Investigating the world of Walkman users". In *New Media and Society*, 3(2), 2001, 170-197.

[3] "Concorde: What Went Wrong?" In *BBC News* September 5, 2000.

[4] Gaye, L., Mazé, R., and Holmquist, L. E. "Sonic City: The Urban Environment as a Musical Interface." In *Proceedings of International Conference on New Interfaces for Musical Expression (NIME03)*, Montreal, 2003.

[5] Gemeinboeck, P. "Impossible Geographies of Belonging". In *ACM Multimedia 2005 Proceedings*, November 6–11, 2005, Singapore. ACM Press, New York, NY, 2005.

[6] Gooding-Williams, R. *Reading Rodney King / Reading Urban Uprising*. Routledge, Inc., London, 1993.

[7] Huxley, A. *Brave New World*. Harper & Bros., New York, 1932.

[8] Open Plan. http://www.open-plan.org/

[9] Icecast. http://www.icecast.org

[10] International Symposium on Electronic Art, San Jose, 2006. http://www.isea2006.sjsu.edu

[11] International Workshop on Mobile Music Technologies. http://www.viktoria.se/fal/events/mobilemusic/

[12] Orwell, G. *1984*. Harcourt Brace Jovanovich, New York, 1949.

[13] PixelAche Festival. http://www.pixelache.ac

[14] West, N. "Urban Tapestries: the Spatial and the Social on your Mobile." In Bart Lootsma. (ed.) *La ville à nu / The naked city. Archilab 2004*. 2004.

[15] Rogoff, I. *terra infirma: geography's visual culture*. Routledge, London, 2000.

[16] Rueb, T. *Choreography of Everyday Movement*, 2002. http://mitpress2.mit.edu/e-journals/LEA/ GALLERY/locative/choreography/index.htm

[17] Sannella, M. J. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Ph.D. Thesis, University of Washington, Seattle, WA, 1994.

[18] Tanaka, A., Tokui, N., Momeni, A. "Facilitating Collective Musical Creativity." In *ACM Multimedia 2005 Proceedings*, November 6–11, 2005, Singapore. ACM Press, New York, NY, 2005.

[19] Tanaka, A. "Mobile Music Making." In *Proceedings of International Conference on New Interfaces for Musical Expression (NIME04)*, Hamamatsu, 2004.

[20] Tanaka, A. "Interaction, Agency, Experience, and the Future of Music." In Brown, B., O'Hara, K. (Eds.) *Consuming Music Together: Social and Collaborative Aspects of Music Consumption Technologies*. Computer Supported Cooperative Work (CSCW) Vol. 35. Springer, Dordrecht. 2006.

CaMus: Live Music Performance using Camera Phones and Visual Grid Tracking

Michael Rohs
Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
michael.rohs@telekom.de

Georg Essl
Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
georg.essl@telekom.de

Martin Roth
Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
martin.roth02@telekom.de

ABSTRACT

We demonstrate that mobile phones can be used as an actively oriented handheld musical performance device. To achieve this we use a visual tracking system of a camera phone. Motion in the plane, relative to movable targets, rotation and distance to the plane can be used to drive MIDI enabled sound generation software or hardware. Mobile camera phones are widely available technology and we hope to find ways to make them into viable and widely used performance devices.

1. INTRODUCTION

Mobile phones and mobile handheld devices are by now a ubiquitous commodity used by many. They have become an integral part of everyday interactions. In this work we consider the idea of making mobile handheld technologies into new interfaces for musical expression. Many people already own mobile technologies. Hence, there is no problem of dissemination and social acceptance. Many novel interfaces for musical expression suffer from this problem because they require new and unfamiliar hardware.

Technologically mobile devices are also converging towards one entity. With respect to music this has so far been with an emphasis on playback [4, 7, 20, 26, 27, 28], and mobile music interactions with a strong input component are yet in their infancy [19]. An example of an interaction paradigm based on mobile technology is GpsTunes [25] where walking navigation is supported by variation in musical playback, which in turn is closely related to the Sonic City project, which however doesn't use a mobile device for sensing [12]. "miniMIXA" from SSEYO (www.sseyo.com) is a music mixer for mobile devices. However, it does not use camera-based input.

This work is based on continuous visual tracking and hence requires the mobile device to contain a camera with reasonable frame rates [1, 2, 23, 24]. By moving the camera phone over visual markers, the user has various degrees of freedom to control parameters which in turn are sent to a computer via Bluetooth. The incoming signal is converted to MIDI messages, which then can be mapped to MIDI-enabled sound generating software and hardware.

The visual display of the phone is used to provide navigational guidance and allow for authoring of the geometric setup of the mapping. A flow diagram of the system is shown in Figure 1.

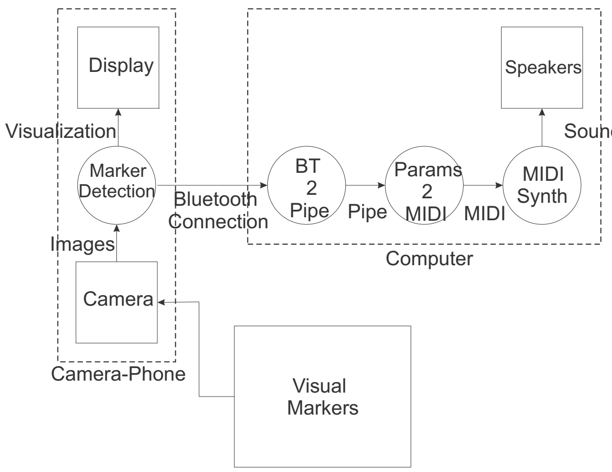


Figure 1: Overall flow diagram of the interface design. The camera phone is used to continuously sense a visual marker grid and display guidance whereas a computer is used to map the data to MIDI and in turn produce sound.

We use a commercial music sequencing program to demonstrate the potential of the mobile interaction for live performance. For this purpose a piece has been written by the third author which can then be remixed live using the CaMus technology.

Other projects have used visual marker recognition to devise new interfaces for musical expression [5, 10, 18]. In all these cases a stationary digital camera is connected directly to a computer and mobile camera phones are not used. Visual tracking for physical mobile interaction has been described in [13, 14, 23]. Hansen et al. [14, 15] track hand-drawn circles, colored objects, and faces, in order to implement *mixed interaction spaces*: physical spaces in which digital interaction takes place. Hachet et al. [13] propose a camera-based interface for two-handed input. One hand holds the device, the other hand holds a card with color codes. The 3D position of the camera relative to the card is mapped to different user interface operations such as pan and zoom, and rotation, and navigation in tree maps.

Figure 2: The user focuses a target that is located at a fixed position in the virtual workspace.

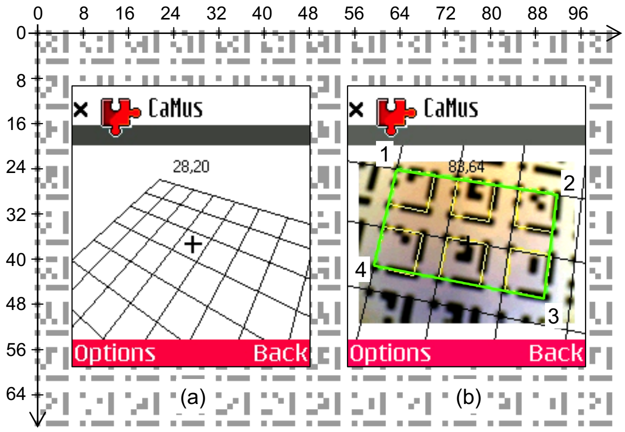


Figure 3: The grid defines an absolute coordinate system. The camera phone computes the coordinates of the cross hair (28,20 in the left screenshot), the distance from the grid surface, and the amount of rotation and tilting. (a) The grid is drawn according to the current camera perspective. (b) The perspective mapping is computed from the maximum area of recognized markers.

2. VISUAL GRID TRACKING AND PERSPECTIVE RENDERING

Interaction takes place on a grid of visual markers, which are derived from visual codes [23]. The grid represents a large workspace, of which different parts can be accessed with a camera phone by simply placing it over the relevant area (see Figure 2). The phone display acts as a window into the virtual workspace. The grid thus enables a spatially aware display [11, 29]. We present one-handed techniques for interactions with objects in space. The space can be shared by multiple camera phones.

The grid defines a coordinate system that provides an absolute frame of reference for the spatial interaction (see Figure 3). Printed on paper, it typically extends over a DIN A4 or A3 sheet. The upper left corner of the grid is the origin, the upper edge is the x-axis, and the left edge is the y-axis of the grid coordinate system. One coordinate unit corresponds to a single black-and-white cell. Each marker has a width and height of 6 cells. Markers are placed two coordinate units apart, which results in one marker for each 8×8 unit area of the grid. The left upper corner stones of each marker are placed at grid coordinates $(8x, 8y), x, y \in \{0, 1, 2, ..., 31\}$.

The markers have a layout similar to visual codes [23], but consist of only two corner stones and two guide bars and a smaller data area. The data area of a single marker has a raw capacity of 12 bits. It is used to store the x index (5 bits) and the y index (5 bits) of the marker within the grid, as well as two parity bits. A grid can thus have a maximum size of 32×32 markers, which is equivalent to 256×256 coordinate units.

In our current implementation on Symbian camera phones with a resolution in view finder mode of 160×120 pixels and with a printed size of 6.8 grid units per cm (i.e. the size of a single marker is about 9×9 mm), the grid is detectable at distances between 2 cm and 10 cm from grid surface to camera lens. This results in a 3D interaction space of $length \times width \times height$ = 37.6 cm × 37.6 cm × 8 cm, if the maximum of 32×32 markers are used. In this space, we can precisely determine the position and orientation of the phone at a rate of up to 10 frames per second with our prototype device (a Nokia 6630 Symbian phone), depending on the complexity of the rendered virtual workspace. In particular, the focus point on the grid surface can be tracked with high precision.

The grid has to allow for a smooth continuous detection of position and orientation during movement. It also has to provide the basis for a perspective rendering of the grid lines as shown in the left screenshot in Figure 3 (labeled a). (Here, the distance between parallel lines is 8 units in the grid coordinate system.) Perspective rendering makes the illusion more convincing, as if the user is looking "through" the device screen onto the background. This is similar to the effect of *symbolic magnifying glasses* [22] and *see-through tools* [6]. To fulfill the first requirement of continuous grid tracking, we have devised a very small visual marker, such that at any position on the surface, and even at close camera distances from the grid, there is at least a single visual marker completely contained in the camera image. However, with smaller sized codes, the perspective mapping between corner points of the code and corresponding grid coordinates is less accurate. Thus, in order to fulfill the second requirement of providing a stable basis for perspective rendering, we use multiple markers in the camera image to establish the mapping. This is shown in the right screenshot in Figure 3 (labeled b).

We need four corresponding pairs of (triple-wise non-collinear) points to establish a perspective mapping between the image coordinate system and the grid coordinate system. Once the markers are recognized, the image pixel coordinates of their corner stones and guide bars are known. This is indicated by the yellow frames around each recognized marker in Figure 3b. (The camera image is not shown during normal use, since it distracts the user.) The closer these points lie together, the less accurate the mapping will be, since a small variation in corner stone pixel positions of successive camera frames results in severe fluctuations of the grid rendering. However, at medium distances, multiple markers are present in the camera image. Hence, to compute the perspective mapping we use the largest possible area to get the most accurate mapping.

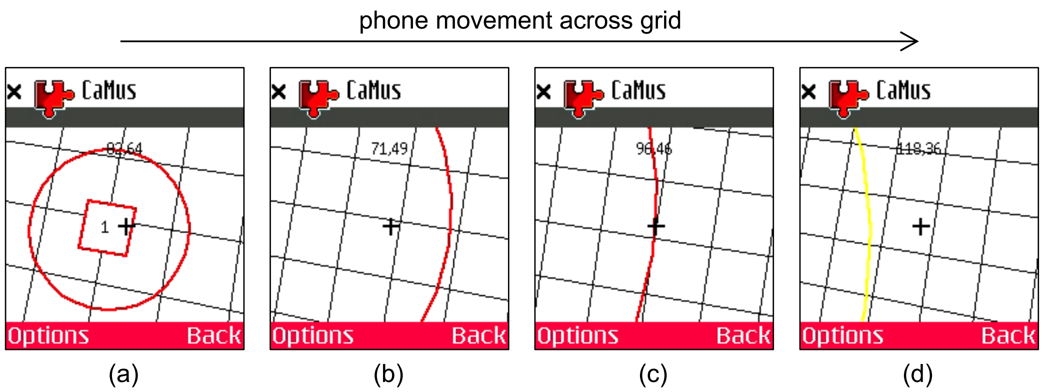


Figure 4: A halo is used to indicate a distant target. Moving from left to right with increasing distance from the target: (a) target is focused, (b) inside the target's range of influence, (c) on the border of the range of influence, (d) outside the range of influence (color changes from red to yellow).

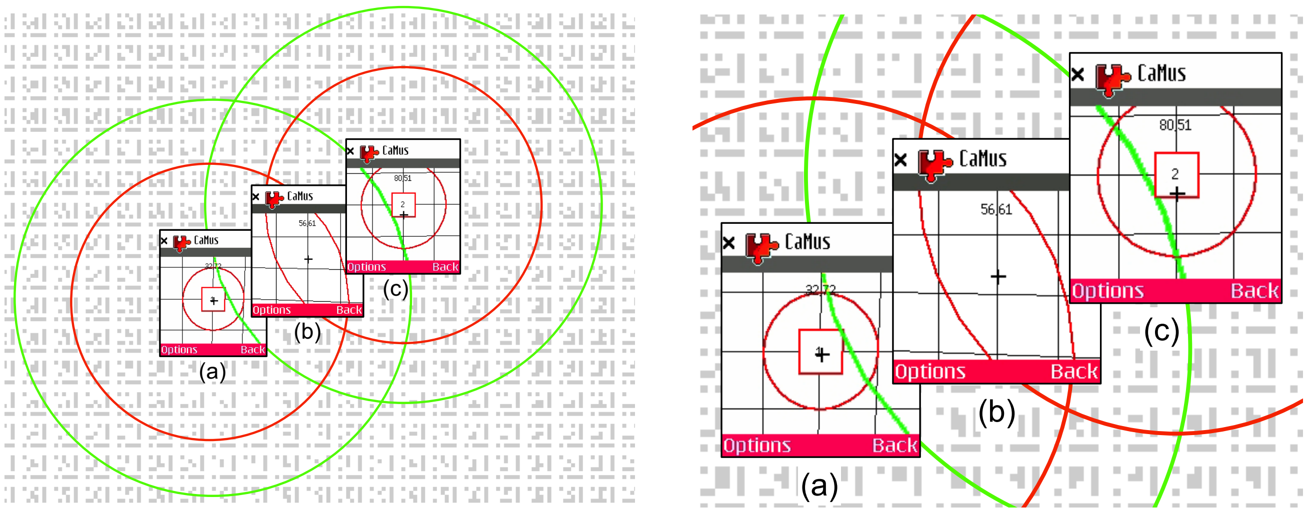


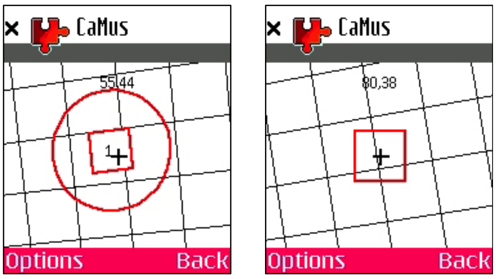
Figure 5: Screen contents at three different positions on a grid with two targets (the right half is a magnification of the left center area): (a) left target focused, (b) between the two targets, (c) right target focused.

In the frame shown in Figure 3b, six markers have been detected. They are highlighted by yellow frames. Instead of basing the perspective mapping on any of the markers' corner points, the points (1) to (4) are used, which represent elements of different markers. The resulting rectangle is highlighted by a green frame. For these corner points, of which we know the image coordinates from the marker recognition step, we establish correspondences as follows. For an upper left corner, its grid coordinates are $(8i_1, 8j_1)$, where i_1 and j_1 are the horizontal and vertical indices that are stored in the marker. The grid coordinates of an upper right corner are $(8i_2 + 5, 8j_2)$, of a lower right corner $(8i_3 + 5, 8j_3 + 5)$, and of a lower left corner $(8i_4, 8j_4 + 5)$. Again, i_k and j_k, $k \in \{1, 2, 3, 4\}$, are the vertical and horizontal index values that are stored in the respective code. From these correspondences, we can compute a perspective mapping (a planar homography) between image coordinate system and grid coordinate system. With this mapping and its inverse, graphical elements that are specified with respect to the grid coordinate system are translated to image coordinates and focus point coordinates are translated to the corresponding point on the grid.

3. VISUAL GUIDANCE

We use the term *target* to denote the basic elements that can be created and placed in the workspace. Each target has a position in grid coordinates and an index number associated to it. Since the small display of a mobile phone can only visualize a small part of the workspace at once, finding targets at different positions can quickly become difficult. Since we want to avoid the need for switching between an overview mode and a closeup mode, we decided to use an extension of the *halo* technique [3] as a way to visualize off-screen targets. This technique supports spatial cognition by surrounding targets with rings that are just large enough to reach into the display window (see Figures 4 and 5). Even if the visible arc is only a small fraction of the ring, it contains all the information needed to intuitively infer the direction and approximate distance of a target. The technique uses very little screen space and has been shown to lead to significantly shorter task completion times compared to arrow-based visualization techniques [3].

Our extension relates to the visualization of the *radius of influence* – and equivalently the circular *area of influence* – that is associated to each target. If the user focuses



a) focus target, press button   b) drag target, while button pressed

Figure 6: Repositioning targets on the grid: (a) focus target and press joystick button, (b) drag target to new location and release joystick button.

a grid point outside the area of influence, then the target has no effect on the output produced. If a point inside the area of influence is focused, then the strength of the effect is inversely proportional to the distance from the center. The goal was to find a visualization that uses the halo technique and does not add additional visual clutter to the interface, yet intuitively tells the user if he is inside or outside the area of influence. Figure 4 shows how the screen contents change as the user moves to the right, starting from the center of a target (a), at which it is surrounded by a minimal halo. Moving to the right, but still inside the area of influence (b), the halo remains to the right of the cursor. If the cursor is inside the arc of the halo, this indicates "insideness." Around the radius of influence (c) there is a boundary region, within which the halo is drawn at that radius at a constant position with respect to the background grid. With further movement, the halo moves with the background grid to the left. When it passes the cursor, its color changes from red to yellow. Color and curvature of the halo now indicate that the user is located outside the area of influence. Using this extension we have reached multiple goals. With a minimal amount of visual clutter, we visualize "insideness" and "outsideness" and the direction and approximate distance of off-screen targets.

The technique scales to multiple targets, as well. Figure 5 shows a grid with two targets. Target 1 is located at grid coordinates $(32, 72)$ and target 2 at $(80, 48)$. The targets are located just outside each other's radius of influence. At position (a), target 1 is focused and the outside of target 2's halo is visible. At position (c) it is the other way round. At the intermediary position (b), the cursor is inside both areas of influence, as indicated by the halos. As proposed in [3], if many targets are present, it is advantageous to merge multiple halos into a single one to indicate a cluster of distant targets.

There are a number of different possibilities of interacting with a target. First, targets can be created at the current cursor position by pressing the phone's binary joystick (also termed multi-way button) in south direction. Pressing it in north direction deletes the closest target.

Targets can be also be freely repositioned on the grid (see Figure 6): (a) a user focuses the target and presses the joystick button, (b) the minimal halo disappears and the target can be dragged to a new location. The target follows the cursor until the button is released.

4. INTERACTION TYPES

Given this technology a number of parameters can be detected. First the x and y position along the two orthogonal axes of the visual marker array. Second is the rotation of the device with respect to the visual marker panel. Distance to the marker array can also be extracted including tilt. In our current implementation we only use position, rotation and distance. This information is then sent to software on the laptop for mapping to MIDI. At this stage already some transformations are applied. In our current implementation, xy-coordinates are converted into relative distances to active markers (see section 3) with a maximum radius of influence, height, and rotational angle in the range of -90 to 90 degrees with 0 degrees being north. Then the respective parameters are linearly rescaled to fit the typical dynamic range of MIDI (0-127).

This information is only sent while the cursor is in the influence radius of a target. In this case rotation of the phone in the horizontal plane 90 degrees left and right of the normal upward position and height of the phone over the sheet with the visual marker array will be sent and are associated to MIDI messages.

An important result is the ability to easily author and perform "cross-mixed" effects. Targets can be placed in relative position to each other. The distance to each other will define how much the respective effects overlap in various spatial regions. By moving inside and outside of these regions effects can be manipulated individually and jointly in a continuous and intuitive way.

An existing target can always be picked up and moved by use of the joystick button. This allows a change in the current effects cross-mix. If a target is no longer desired it can also be removed.

It is important to note that these choices are already arbitrary. Alternative designs are thinkable. For example, the x and the y axis can be used separately to implement axially dependent controls, for example up-down controlling one parameter like a vertical slider and left-right doing the same for a horizontal slider.

Our subjective experience suggests, that the notion of distance is rather intuitive. It is not prone to problems due to a difference in alignment of hand motion direction to marker sheet axis.

Our current interaction mechanisms have two target audiences in mind. First a knowledgeable performer, who is aware of the relative effects of mapped targets. This user can place targets and move them into relative position to compose a parameter landscape. The second audience is a novice user, who can explore this landscape with minimal musical knowledge and achieve musically interesting results by exploration.

5. MAPPING

The targets themselves do not offer an inherent semantics for their use in interaction. The mapping of data associated to them (position, rotation and height) can be freely mapped to sound synthesis parameters, effects parameters or other attributes chosen for control. This is a source of great flexibility of this implementation but at the same time also brings this device close towards the classical mapping problem of new musical instruments where a gesture may not have an intrinsic musical meaning [17].

In our example mappings we chose to map height to overall strength of an effect, relative distance in the horizontal plane to a target for a primary effect parameter, which results in controlling the most salient feature of the effect. If a secondary effects parameter was of interest it was mapped to rotation of the device in the horizontal plane. The first two gestures are similar to the analogy of gravity or strength of influence by proximity, whereas the last gesture resembles the gesture used for knob or dial control.

6. SOUND

Figure 7: An example mapping of CaMus to the commercial sequencing software FruityLoops (by Image-Line Software). Various parameters are mapped to the knobs of various effects boxes via MIDI.

The interface can be used with any software that offers MIDI input. We experimented with the commercial sequencing software FruityLoops (by Image-Line Software, see Figure 7), as well as the research projects STK [8, 9] and PD [21].

Our main sounding environment and application has been on the fly re-mixing of music. The third author wrote a piece within FruityLoops for this purpose. The ability of FruityLoops to link control-knobs of its interface to incoming MIDI messages was then used to map the controls of the camera phone to various pre-selected effects (Figure 7). The respective mappings are depicted in Table 1.

6.1 Sound, Gesture and Visual Display

For each of the effects, a separate target can be freely placed within the plane. The performance consists of two basic features.

The first is the authoring or configuration feature. In this case, targets are placed and moved in the plane. This happens interactively by clicking a button on the camera phone to either place a new one or grab an existing one to drag it. Dragging continues as long as the button is held. As long as a target is present it emits control signals to the laptop and in turn to the MIDI software. The main function of this step is to pick effects which are desired for a performance. One can either fix their relative spatial location before the performance starts or change the relative positions between targets on the fly during a performance.

The second feature consists of performing with one or more interaction target. Whether or not a target is currently being a source of manipulation is determined by the distance of the camera cursor position relative to the target position. If the cursor is within a preset range, all effects of that target will be active and sent to the MIDI software. Because multiple targets can be within range of the current cursor position, multiple effects of the MIDI software can be manipulated simultaneously. At the same time, their respective spatial separation allows for various degrees of influence or configuration of the effect leading to a sort of multi-effect hybridization. For example, one can place two targets with some separation. In the region between the targets both effects will receive high control values, whereas in the regions on one side or the other of the target leads to a joint effect where one effect parameter is low while the other parameter is high. This joint parameter space can be conveniently explored by moving in the plane and easily extends to multiple effects (see Figure 5). The behavior of the effect is intuitive by the "strength of proximity"-analogy.

Effect	Distance	Height	Rotation
Distortion	Distortion	Effect weight	Not used
LP filter	Cut-off freq.	Effect weight	Not used
Balance	Not used	Effect weight	Left/right
Delay	Forward delay	Effect weight	Feedback delay
Reverb	Reverb	Effect weight	Room size

Table 1: Mapping of CaMus parameters to digital audio effects.

The result is an easy to use multi-effects performance interface, which in this particular setting can be used for life remixing of sequenced software using real-time digital effects.

7. CONCLUSIONS

We presented a mobile camera phone based interface for musical expression, which uses visual marker technology to allow hand gesture based performance of music and music remixing.

The main motivation behind developing this technology is a first indication at the potential of commodity handheld mobile devices serving as novel interfaces for musical expression. These devices are already very widely available and hence one can hope that expressive musical uses can more readily reach a large practitioner base.

We have described a camera based sensing technology that was used to design interactions for musical performance allowing for motion in the plane, hand rotations in the plane as well as distance control. By using MIDI to interface with the synthesis software, a large array of sounding sources and specific mappings can be achieved.

Future plans include various modes of collaborations, including the use of phones interacting with each other, the extension of the plain target paradigm to allow a variety of analogies, potentially including sliders and dials. We are currently to implementing a synthesis engine on the mobile phone to allow stand-alone use of the mobile phone for performance. Finally we plan to either augment or substitute the current sensing technology with other modes of sensation [16] and provide additional sensory modes of display, for example tactile.

In terms of camera-based input, one future direction is tracking everyday types of visuals – like faces, posters, or specific colors – and map them to characteristic musical output. Another direction is to capture phone movement with optical flow techniques and map it to filters. This could also be the basis for the recognition of compound gestures that are linked to specific musical output.

8. REFERENCES

[1] R. Ballagas, M. Rohs, and J. G. Sheridan. Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1200–1203, New York, NY, USA, April 2005. ACM Press.

[2] R. Ballagas, M. Rohs, J. G. Sheridan, and J. Borchers. The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing*, 5(1):10–17, Jan. 2006.

[3] P. Baudisch and R. Rosenholtz. Halo: A Technique for Visualizing Off-Screen Objects. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 481–488, New York, NY, USA, 2003. ACM Press.

[4] F. Behrendt. *Handymusik. Klangkunst und 'mobile devices'*. Epos, 2005. Available online at: `http://www.epos.uos.de/music/templates/buch.php?id=57`.

[5] R. Berry, M. Makino, N. Hikawa, and M. Suzuki. The Augmented Composer Project: The Music Table. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 03)*. IEEE, 2003.

[6] E. A. Bier, M. C. Stone, K. Fishkin, W. Buxton, and T. Baudel. A Taxonomy of See-Through Tools. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 358–364. ACM Press, 1994.

[7] W. Carter and L. S. Liu. Location33: A Mobile Musical. In *Proceedings of the Mobile Music Workshop*, Vancouver, Canada, May 25 2005.

[8] P. Cook and G. Scavone. The Synthesis ToolKit (STK). In *Proceedings of the International Computer Music Conference*, Beijing, 1999.

[9] P. R. Cook. *Real Sound Synthesis for Interactive Applications*. A K Peters, Ltd., July 2002.

[10] E. Costanza, S. Shelley, and J. Robinson. Introducing Audio D-Touch: A Tangible User Interface for Music Composition and Performance. In *Proceedings of the International Conference on Digital Audio Effects (DaFX-03)*, 2003.

[11] G. W. Fitzmaurice. Situated Information Spaces and Spatially Aware Palmtop Computers. *Commun. ACM*, 36(7):39–49, 1993.

[12] L. Gaye, R. Mazé, and L. E. Holmquist. Sonic City: The Urban Environment as a Musical Interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003.

[13] M. Hachet, J. Pouderoux, and P. Guitton. A Camera-Based Interface for Interaction with Mobile Handheld Computers. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 65–72, New York, NY, USA, 2005.

[14] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Mixed Interaction Space: Designing for Camera Based Interaction with Mobile Devices. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1933–1936, New York, NY, USA, 2005. ACM Press.

[15] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Mixed interaction spaces – a new interaction technique for mobile devices. Demonstration at UbiComp, 2005.

[16] S. Hughes, I. Oakley, and S. O'Modhrain. MESH: Supporting Mobile Multi-modal Interfaces. In *Proceedings of UIST'04*, Sanfa Fe, NM, 2004.

[17] A. Hunt, M. M. Wanderley, and M. Paradis. The Importance of Parameter Mapping in Electronic Instrument Design. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, pages 149–154, Dublin, Ireland, May 24-26 2002.

[18] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reacTable*. In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.

[19] M. Kaltenbrunner. Interactive Music for Mobile Digital Music Players. In *Inspirational Idea for the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.

[20] G. Levin. Dialtones - A Telesymphony. Description of the Dialtones concert performances `http://www.flong.com/telesymphony/`, September 2 2001. Retrieved on January 23, 2006.

[21] M. Puckette. Pure Data. In *Proceedings of the International Computer Music Conference*, pages 269–272, San Francisco, 1996. International Computer Music Association.

[22] J. Rekimoto. NaviCam: A Magnifying Glass Approach to Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):399–412, Aug. 1997.

[23] M. Rohs. Real-World Interaction with Camera Phones. In H. Murakami, H. Nakashima, H. Tokuda, and M. Yasumura, editors, *Second International Symposium on Ubiquitous Computing Systems (UCS 2004), Revised Selected Papers*, pages 74–89, Tokyo, Japan, July 2005. LNCS 3598, Springer.

[24] M. Rohs. Visual Code Widgets for Marker-Based Interaction. In *IWSAWC'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems – Workshops (ICDCS 2005 Workshops)*, pages 506–513, Columbus, Ohio, USA, June 2005.

[25] S. Strachan, P. Eslambolchilar, R. Murray-Smith, S. Hughes, and S. O'Modhrain. GpsTunes: Controlling Navigation via Audio Feedback. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services*, Salzburg, Austria, September 19-22 2005.

[26] N. Warren, M. Jones, S. Jones, and D. Bainbridge. Navigation via Continuously Adapted Music. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1849–1852, New York, NY, USA, 2005. ACM Press.

[27] D. Waters. Mobile Music Challenges 'iPod Age'. BBC News Online, Monday, 7 March 2005. Available online at `http://news.bbc.co.uk/1/hi/technology/4315481.stm`.

[28] T. Yamauchi and I. Toru. Mobile User-Interface For Music (poster). In *Proceedings of the Mobile Music Workshop*, Vancouver, Canada, May 25 2005.

[29] K.-P. Yee. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–8, New York, NY, USA, 2003. ACM Press.

Pocket Gamelan: tuneable trajectories for flying sources in *Mandala 3* and *Mandala 4*

Greg Schiemer
Faculty of Creative Arts
University of Wollongong
+61 2 4221 3584
schiemer@uow.edu.au

Mark Havryliv
Faculty of Creative Arts
University of Wollongong
+61 2 4221 3584
mhavryliv@hotmail.com

ABSTRACT
This paper describes two new live performance scenarios for performing music using bluetooth-enabled mobile phones. Interaction between mobile phones via wireless link is a key feature of the performance interface for each scenario. Both scenarios are discussed in the context of two publicly performed works for an ensemble of players in which mobile phone handsets are used both as sound sources and as hand-held controllers. In both works mobile phones are mounted in a specially devised pouch attached to a cord and physically swung to produce audio chorusing. During performance some players swing phones while others operate phones as hand-held controllers. Wireless connectivity enables interaction between flying and hand-held phones. Each work features different bluetooth implementations. In one a dedicated mobile phone acts as a server that interconnects multiple clients, while in the other point to point communication takes place between clients on an ad hoc basis. The paper summarises bluetooth tools designed for live performance realisation and concludes with a comparative evaluation of both scenarios for future implementation of performance by large ensembles of non-expert players performing microtonal music using ubiquitous technology.

Keywords
Java 2 Micro Edition; j2me; Pure Data; PD; Real-Time Media Performance; Just Intonation.

1. INTRODUCTION
In the past decade there has been a paradigm shift from desktop to ubiquitous computing. Mobile phones represent a major part of this shift. Pd2j2me, a set of software tools used to create live musical applications that run in the java 2 micro edition [1, 2], was previously described [3]. It was developed as part of a project that seeks to address the challenge of composing music for mobile computing environments. Called the Pocket Gamelan, the project seeks to develop an interactive musical performance interface that allows non-expert performers to perform microtonal music using mobile phones [3, 4, 5]. In previous papers we described how performance scenarios associated with the project might extend the musical legacy of historical tuning systems as well as new tuning systems first explored by composer and theorist Harry Partch [6] and later extended through the work of contemporary tuning theorist Erv Wilson [7, 8, 9, 10, 11]. Wilson's tuning theories are inspirational because they offer a broad map of the microtonal world that is informed by tuning theorists from many musical traditions. Extensible scales produced by his combination product set theories, and the related Euler-Fokker genus, have a particular appeal for a project focused on the exploration of microtonal music through extensible interfaces.

Two new microtonal works for mobile phones have been created and performed. Microtonal files were created for the Pure Data (Pd) composition environment using recently added features of Scala tuning software designed to export and document tuning data. Purpose-built tools were used to translate Pd files into j2me, a format suitable for java phones. Prior to its realisation using multiple phones, each performance was auditioned using Pd files running on a single desktop machine.

1.1 Pocket Gamelan – pd2j2me
Pd2j2me was implemented as a library of j2me classes that allows composers working in the Pure Data composition environment to create music that can be performed by an ensemble of mobile phones. Since pd2j2me was first developed, two further performance scenarios have been implemented using communication between bluetooth-enabled mobile phone handsets.

A new composition interface has been devised for each work in order to explore the musical possibilities of each scenario. The interface for each application was developed in Pd then exported to the java environment using pd2j2me. Some java was hand coded during the final stages of translation between Pd and j2me. It is our objective, however, that all translation will eventually be automated. The most recent version of pd2j2me was submitted as part of a thesis on 19th December 2005 [12].

2. NEW BLUETOOTH PERFORMANCE SCENARIOS
In performing both the works discussed here, players swing mobile phones on the end of a cord in a circular trajectory, as shown in Figure 1a. Each mobile phone is mounted in a pouch made of semi-transparent fabric attached to a cord.

Figure 1a. Mobile phones are used as flying sound sources in *Mandala 3* and *Mandala 4*.

As the phone is swung by its cord it produces audible artefacts such as Doppler shift and chorusing which are produced as a bi-product of movement. The performance concept originated from purpose-built mobile instruments developed by one of the authors more than two decades earlier [13, 14]. One of these instruments, the UFO, is shown in Figure 1b. UFO is an acronym for Ubiquitous Fontana Oscillator, in acknowledgement of sound sculptor Bill Fontana who used flying speakers in 1976.

Figure 1b. One of 16 UFOs from the Tupperware Gamelan built for *Mandala 2* in 1981.

Bluetooth communication allows the audio algorithms used in *Mandala 3* and *Mandala 4* to be altered during performance as shown in Figure 1c. During hand-held operation a phone keypad may be operated easily through the fabric as shown in Figure 1d.

Figure 1c. Flying phones are controlled by hand-held phones.

Figure 1d. Pressing a button sends a bluetooth message.

2.1 *Mandala 3*

In the first of these works, entitled *Mandala 3,* three mobile phones (Nokia 6230) interact with one another via a fourth mobile phone (Nokia 7610) which acts as a dedicated server, as shown in Figure 2a. Each 6230 is used both as a sound source and bluetooth controller while the 7610 is used to relay control messages to the other phones. This configuration allows all three sound sources to be affected by the action of any player.

Figure 2a. A mobile 'server' configuration allows any player in the ensemble to affect sound on other phones.

Mandala 3 uses a microtonal scale ascribed to 8th century theorist Al-Farabi [15]. It was first performed at the 13th Australasian Computer Music Conference 2005, Queensland University of Technology, The Loft, Creative Industries Precinct, Kelvin Grove, Brisbane, July 12th 2005.

2.2 *Mandala 4*

In the second of these works, entitled *Mandala 4*, mobile phones communicate with one another on an ad hoc basis. Four Nokia 6230 phones are used both as sound sources and bluetooth controllers; each player in turn operates the phone as a bluetooth controller in order to affect sound on one of the other flying phones. The configuration is shown in Figure 2b.

Figure 2b. An ad hoc connection system makes connections only when required.

Each phone is colour-coded as indicated by the colour of its LCD screen. When a single coloured button appears on the screen of a hand-held 6230, a player may choose to respond by sending a control message that affects a flying sound source on another phone. The colour of the on-screen button indicates which phone is affected. The composition specifies a sequence of cues communicated by players during the performance. These cues ensure that each player is ready with phone in hand before the button appears on the screen. As soon as it appears, a player may respond by pressing any key on the 6230. This selects new tuning modes as described in section 4.

Mandala 4 uses one of Wilson's combination product set scales called the Euler-Fokker Genus [15]. This was first performed in the Wild Dog Concert as part of UK Microfest, Riverhouse Barn, Walton-on-Thames, October 15th 2005.

3. DEVELOPMENT ENVIRONMENT – FROM SCALA TO J2ME

Each of the mobile phone performances described above is initially developed in a desktop environment. This allows microtonal aspects of each work to be heard and sequencing of the mobile phones to be simulated. The development path is as follows:

- Scala produces Pd file and exports tuning data to Pd
- Composer works in Pd and simulates sequencing of mobile phones
- pd2j2me produces j2me code
- j2me code is exported to Wireless Toolkit and optionally, refined in Eclipse
- j2me code is compiled and emulated using Wireless Toolkit producing JAR/JAD files
- JAR/JAD files are downloaded to each handset using Nokia PC Suite

The development environment is shown in Figure 3.

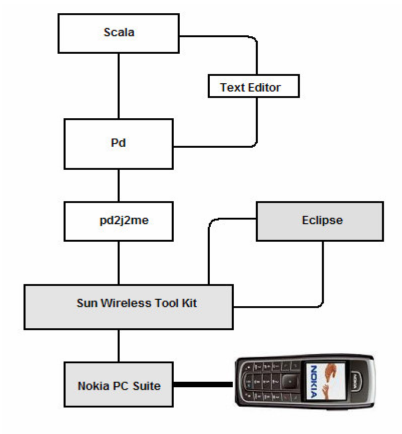


Figure 3. The development path from Scala to deployment to the mobile phone.

3.1 Scala

Scala was chosen for its tuning tools and extensive knowledge base of tuning systems [15]. These include tools to export scales and information related to tuning into other applications. Scala scripting tools extended by Scala's author, Manuel Op de Coul, for the Pocket Gamelan project to allow scale documentation to support file formats used in Pd. Tuning data exported to Pd is stored as a table of linear factors. Linear factors are normally formed by dividing the fraction in a just interval. Scala's tools allow tuning intervals normally measured in cents to be expressed as linear factors. Scales expressed as linear factors simplify tuning operations in Pd because frequencies can be calculated for each scale degree by multiplying a reference frequency by each factor. Extended Scala commands were subsequently released in Scala version 2.2o.

The extended Scala commands gradually evolved into an automated text-based method of generating the Pd interface used in composing these works. As a Pd file was refined or a new canvas added, it was initially saved in Pd. The Pd file was then manually modified using a text editor, and assembled as part of a Scala command file. A new Pd file is then rendered by running the Scala command file. This allowed Pd canvas coordinates and arguments to be initialised consistently, unaffected by the vagaries of the mouse or GUI. In this way Scala managed development of the Pd interface as it grew in complexity. Eventually Pd project management may be automated in Scala using the 'spawn' command.

Scala command files to render Pd applications for *Mandala 3* and *Mandala 4* can be downloaded separately [16].

3.2 j2me

The j2me environment includes:

- pd2j2me – this was discussed previously in 1.1.
- Eclipse – this is used where a particular need is not yet supported by pd2j2me.
- Sun Wireless Toolkit – this compiles j2me source code and simulates the running of the application; it also generates JAR and JAD files which contain the application and application data.
- Nokia PC Suite – this uploads the JAR and JAD files to a target Nokia phone.

4. WORKS

4.1 *Mandala 3*

The Pd canvas in Figure 4a shows the 7-note Dorian scale used in *Mandala 3.*

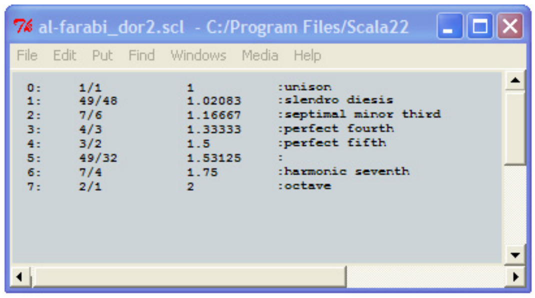


Figure 4a. Al-Farabi's Dorian scale used in *Mandala 3*.

The tuning of Al-Farabi's Dorian scale lends itself to melodic treatment of the kind that can be realized using the Pd interface shown in Figure 4b. This interface plays up to three voices simultaneously. Each voice is a continuous oscillator running on a mobile phone. In the j2me environment each voice is represented by one of three phones, phone A, B and C respectively. Phone functions used in *Mandala 3* will now be explained with reference to the Pd interface shown in Figure 4b.

Single voice melodic phrases are generated on each phone in real-time. Each generator produces a stream of numbers that represent the pitch class of a seven note scale. These address an array containing linear factors used to implement the Al-Farabi tuning.

Pitch classes are generated by using a modulo-7 counter to produce scale steps. Bitwise inversion is then performed on each scale step to vary the melodic contour. Toggle switches in the canvas labelled 'BitwiseInvert1' allow the user to preset various contours. Ascending scales are generated by setting all switches off, descending scales by all switches on and various melodic contours using various combinations of on and off. Pitches may be played over a range of four octaves and start on different scale degrees, or modes, using three groups of horizontal radio buttons labelled 'Keyboard', 'Octave' and 'Mode'. These groups of buttons serve as an input keyboard for users to rehearse with the sequence as well as a status indicator during playback.

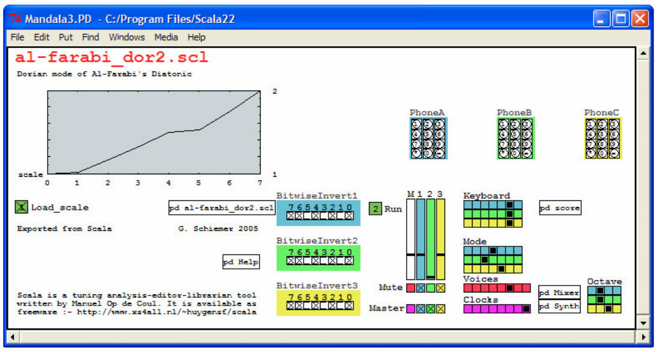


Figure 4b. Pd interface for *Mandala 3*.

The rhythmic characteristics of melodic phrase are defined by changing the clock period for each note event. The note length of the next event is read from a table of preset values. Events in each phone are sequenced by a local clock. Though each local clock runs asynchronously they are started from the echo server via bluetooth during initialization. There are no discrepancies between the timing of sequences on phones even after a period of twenty minutes. This makes it possible to create elaborate three-voice hockets in which each independent part is rhythmically synchronized.

On the Pd interface, three mute buttons situated below three vertical sliders are used to activate or deactivate each voice. A voice is inactive if its vertical slider is set to zero and its associated mute button is deactivated. The Pd interface also shows two eight-state horizontal radio buttons. The first, labelled 'Voices', is also used to activate or deactivate voices; this allows all eight combinations of voices to be selected in a single key operation. The second, labelled 'Clocks', allows each sequence to advance or pause independently or in tandem with other voices. Any one-of-eight antiphonal combinations can be selected in a single key operation.

In performance, players select various pre-set combinations by pressing buttons on their phones. This makes it possible for Player A to interact first with Player B and without Player C, then, sometime later, with Player B and with Player C; and so on. On the Pd interface, phone keys are mapped to various control functions using the three twelve-key phone canvases to configure mapping. In performance, the tempo of every phone may be increased or decreased by a factor of 2. This is done by selecting buttons '4' '5' and '6' on any phone; '4' is half speed, '5' is normal speed and '6' is double speed. A tempo change command issued on one handset is relayed by the echo server and takes effect on every handset. This allows any player to select the overall tempo of all sequences. A more complete discussion of mapping for all functions is beyond the scope of this paper.

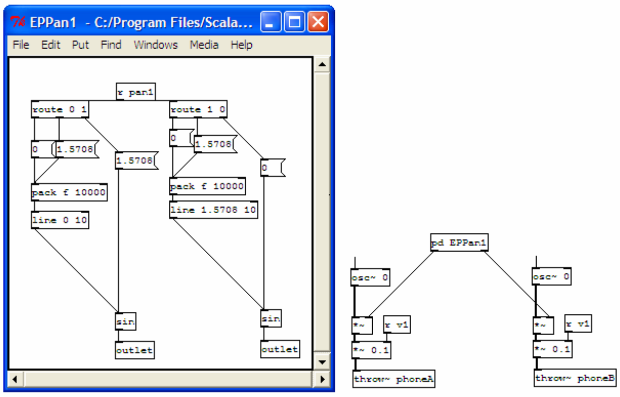


Figure 4c. Pd algorithm allows panning between two phones as shown in sub-patch EPPan1.

The tone generator in each phone also has a second oscillator which doubles at the octave. This allows cross fading between one tone and another an octave apart and results in a gradual timbral change while a sequence is playing. Cross-fading between tones in a single phone is implemented using an equal-powered algorithm based on sine and cosine in the same quadrant. Equal powered cross-fading can also be executed as a form of panning between phones as shown in Figure 4.c. This creates the illusion that sound moves from one phone to another.

4.2 *Mandala 4*

The Pd canvas in Figure 4d shows a 10-note scale used in *Mandala 4*.

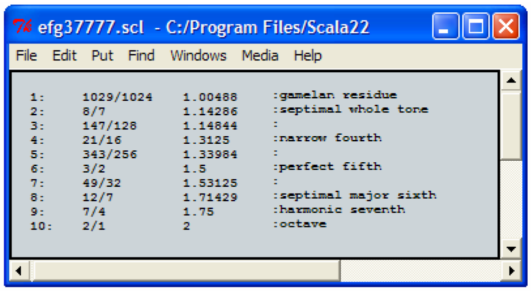


Figure 4d. Erv Wilson's Euler-Fokker Genus scale based on harmonics 3 and 7 used in *Mandala 4*.

The Pd interface shown in Figure 4e allows control sequences played on all four phones to be programmed and auditioned. Each phone produces four oscillators. Throughout the performance the amplitude of each oscillator is continuously varied using a concatenated series of line functions. The sixteen vertical sliders on the right of Figure 4e allow the composer to monitor changes in amplitude that occur during performance. The composing interface also allows pitch class, octave and transposition to be pre-programmed.

When the performance begins, players assemble in the centre of the stage area to synchronise automated sequences that run throughout the performance. The synchronization sequence takes about 30 seconds and is described in Section 5.2.

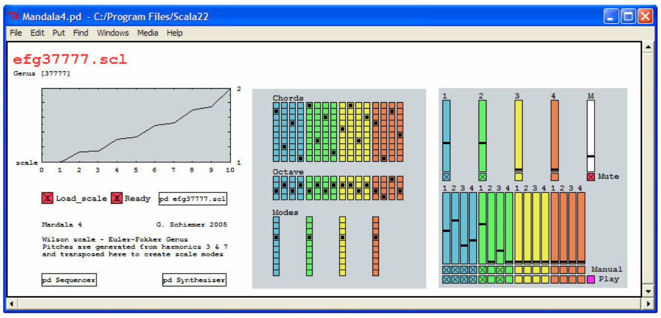


Figure 4e. Pd interface for *Mandala 4*.

Once all three devices have been synchronised, pre-programmed sequences on all phones begin to play synchronously. Entry and exit of sequences on phones is pre-programmed as shown by the cues in Figure 4f. These sequences ensure that:

- a single cue is given for every event; these provide an aural reference for players during the performance
- each player provides four cues that the other players may use throughout the performance
- every combination of phone solo, duo and trio as well as all four phones is heard.

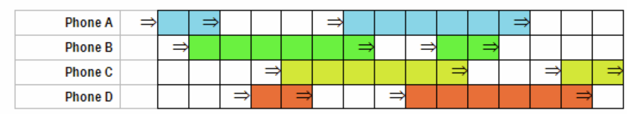


Figure 4f. An automated event sequence determines when sound is produced on each phone. Every 15-30 seconds an event occurs, i.e. a single tone either starts or stops.

In equal temperament, modulation causes pitches to shift to different degrees of the same scale. Here, where just intonation tuning is involved, non-uniform interval sizes cause pitches to shift to positions not audible in the original (i.e. untransposed) scale [16]. In effect, transposition behaves like modulation.

Vertical radio buttons labelled 'Modes' shown in figure 4e, allow intervals of the 10-note scale to be transposed using the same intervals. This is accomplished through the multiplication of linear factors.

In this context, scales played in various transpositions beat with other transpositions of the same scale. This produces beating and other timbral effects which are the bi-products of microtonal intervals produced by modulation.

5. BLUETOOTH TOOLS

While *Mandala 3* and *Mandala 4* both involve wireless communication between several phones, in each case the development process yielded two different ways of managing a bluetooth network: an echo server model implemented as a bluetooth *piconet* and an ad hoc connection system. The development of these was principally driven by a significant limitation in the Nokia 6230's bluetooth implementation: it can only be connected to one other phone at any time.

5.1 Echo server piconet – *Mandala 3*

A bluetooth *piconet* is created when a device is connected to one or more other devices [17]. One device must be the master. In *Mandala 3,* three slaves (6230) communicate bi-directionally with the master (7610), as shown in Figure 2a. Unlike the 6230, the 7610 allows up to seven bluetooth channels to be addressed.

Initially the master device searches for the three phones within a ten metre radius; unsolicited participants are excluded. The phones are then identified with a number – or phone ID – that represents the order in which they are discovered.

Pressing a key on any of the three performing phones generates a message consisting of its phone ID and the key pressed. This message is then echoed by the master to each performing phone. The phone ID and the value of the key pressed determines the response of each phone.

5.2 Ad hoc connection – *Mandala 4*

In *Mandala 4*, the ID of each phone is determined prior to the performance and stored by the phone as its bluetooth friendly name, i.e. the name by which it appears to other devices. Also pre-determined are the times at which one phone attempts to connect to another.

Phones are initially synchronised by Phone A, which searches for the other three phones as soon as the application is run. Phone A then connects and sends a different delay value to each of the other phones which in turn starts a timer. Delay values are pre-computed so all phones begin the piece at the same time.

Once the synchronization sequence is complete, the automated sequences described in section 4.2 begin playing. In the background, each phone then begins a bluetooth discovery sequence searching for other phones taking part in the performance. This allows each phone to identify the bluetooth radio address of other phones to which it must connect during the performance. The bluetooth radio address of each phone discovered is then stored in memory. This allows connections to be made quickly at the appropriate time.

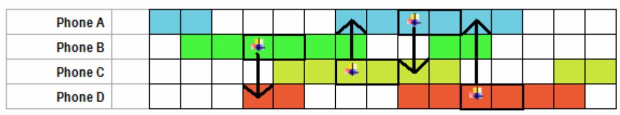


Figure 5. Bluetooth connects a hand-held phone to a flying phone allowing the player to modify its tuning.

Figure 5 shows four windows in the playing sequence during which one phone will automatically connect with another. The player of the phone which initiates the connection is then able to perform operations on a flying phone.

When phone B initiates connection to phone D, as shown by the direction of the arrow in Figure 5, a red button appears on the screen of phone B. This indicates to player B that a new tuning mode may be selected on phone D. Player B may then choose to select a new mode by pressing any key. Mode selection uses the transposition algorithm described in section 4.2. The same thing happens in the remaining three windows shown in Figure 5, where phone C connects to phone A, phone A to phone C and phone D to phone A.

6. CONCLUSION

Work done so far demonstrates the potential of musical applications used with mobile phones. All development has been done using open source software. The tuning resources of Scala, already accessible in many desktop applications like Csound, Artwonk and Metasynth, can now be exported to Pd where pd2j2me is currently used to export applications to java phones. In time, these musical resources can also be extended to other music composition environments like MaxMSP, Supercollider, Algorithmic Composer and AudioMulch, to name a few.

Currently, phones require wired connection for downloading musical applications. In the future, we envisage scenarios in which servers allow tunable applications to be downloaded by multiple clients via a wireless connection. These scenarios provide a broader musical framework in which musicians in future will work with sensor networks based on ultrawideband (UWB) communications [18]. Such networks will dramatically extend the capabilities of bluetooth beyond those explored in *Mandala 3* and *Mandala 4* and address shortcomings of bluetooth implementation in the current generation of phones; we would like all UWB devices to have the capability to multicast in an ad hoc network without the necessity to use an echo server. Moreover, for live musical performance, connection initiated at the discretion of the player is preferable to connection initiated by an automated process; the same is also true of musical games for multiple players [19]. Ongoing development of work begun in *Mandala 3* and *Mandala 4* calls for a new generation of phone that is fully compliant with MIDI, can synthesise streaming audio files and recognise human performance gestures other than the pressing of buttons [20]. This will allow mobile phones to be a generic performance tool widely used by musicians. The frequency hopping algorithm on which mobile phone technology is built has its origins in *Ballet Mecanique*, one of the earliest forays by a twentieth century composer into musical instrument design [21]. Future development of this technology can still be driven by communities of musicians just as such communities drove the development of MIDI more than two decades ago. We see new sensing networks based on UWB potentially as interactive musical instruments that are as diverse as the tuning systems used by musicians over many centuries and in many civilizations.

7. ACKNOWLEDGEMENTS

This project was funded by an Australian Research Council Discovery Grant for 2003-2005. Manuel Op de Coul wrote extensions to Scala source code and offered support during the testing of new scripts.

8. REFERENCES

[1] Sun Microsystems, Java 2 Micro Edition 1999

[2] Connected Limited Device Configuration (CLDC) 1.0, Mobile Information Device Profile (MIDP) 1.0.

[3] Schiemer, G. and Havryliv, M. “Pocket Gamelan: a Pure Data interface for mobile phones” *Proceedings of the New Interfaces for Musical Expression Conference*, *Vancouver, Canada* pp. 156-159 2005

[4] Schiemer, G. and Sabir, K. and Havryliv, M. 2004 “The Pocket Gamelan: A j2me Environment for Just Intonation” *Proceedings of ICMC2004 University of Miami, Florida, November 4th-9th* pp. 654-657 2004

[5] Schiemer, G. Schiemer, G., Alves, W., Taylor, S. and Havryliv, M. “Pocket Gamelan: building the instrumentarium for an extended harmonic universe” *Proceedings of the International Computer Music Conference*, *Singapore* pp. 329 -332 2003

[6] Partch, H. Genesis of Music Da Capo Press 1949

[7] Wilson, E. Musical Instrument US Patent Office Patent Number 3,012,460 1961

[8] Wilson, E. Musical Instrument Keyboard US Patent Office Patent Number 3,342,094 1967

[9] Wilson, E. “The development of Intonational Systems by Extended Linear Mapping” 1975 (1)

[10] Wilson, E. “Bosanquet – A Bridge – A Doorway to Dialog” Xenharmonikôn 3 (13 pages) 1975 (2)

[11] Wilson, E. “D'Alessandro Like a Hurricane” Xenharmonikôn 9 pp. 1-38 1986

[12] Havryliv, M. 2005 “Playing with audio: the relationship between music and games” Master of Creative Arts thesis, University of Wollongong, 2005

[13] Atherton, M. Australian Made - Australian Played University of New South Wales Press, Sydney pp. 210-211 1991

[14] Schiemer, G. “Improvising Machines: Spectral Dance and Token Objects” *Leonardo Music Journal 9* MIT Press pp. 107-114 1999

[15] Op de Coul, M. Scales Archive in Scala available at http://www.xs4all.nl/~huygensf/scala/ 1992

[16] http://www.uow.edu.au/staff/crea/schiemer/.html

[17] Schiemer, G. and Havryliv, M. “Pocket Gamelan: Bluetooth scenarios for mobile phones” *Proceedings of ACMA’2005 Queensland University of Technology, Kelvin Grove, July 12th-15th* pp. 66-71 2005

[18] Nekoogar, F. “Ultra-wideband Communications: Fundamentals and Applications” Prentice Hall, Sydney, 2005

[19] Havryliv, M. and Narushima, T. “Metris: a Game Environment for Music Performance” *Proceedings of Computer Music Modeling and Retrieval, Pisa, Italy* 2005

[20] Schiemer, G. and Havryliv, M. “Wearable Firmware: The Singing Jacket” *Proceedings of ACMA’2004 University of Victoria, Wellington, July 1st-3rd* pp. 66-71 2004

[21] Price, R Further Notes and Anecdotes on Spread-Spectrum Origins see IV. “Shortly before Pearl Harbour: The Lamarr-Antheil Frequency-Hopping Invention” IEEE Transactions on Communications Vol. COM-31, No.1 pp. 89-91 1983

Interactive Public Sound Art: a case study

David Birchfield, Kelly Phillips, Assegid Kidané, David Lorig
Arts, Media and Engineering
Arizona State University
Tempe, AZ 85287
1.480.965-3155
{ dbirchfield, crowbar, assegid.kidane, david.lorig } @ asu.edu

ABSTRACT
Physically situated public art poses significant challenges for the design and realization of interactive, electronic sound works. Consideration of diverse audiences, environmental sensitivity, exhibition conditions, and logistics must guide the artwork. We describe our work in this area, using a recently installed public piece, *Transition Soundings*, as a case study that reveals a specialized interface and open-ended approach to interactive music making. This case study serves as a vehicle for examination of the real world challenges posed by public art and its outcomes.

Keywords
Music, Sound, Interactivity, Arts, Public Art, Network Systems, Sculpture, Installation Art, Embedded Electronics.

1. INTRODUCTION
With the advent of inexpensive and often readily available delivery mechanisms, the scope of public art has expanded to include a wide range of methodologies and outcomes. For example, there have been a number of radio broadcast and more recent internet projects, including our own work [10, 9], that reach a wide public audience. Newer ubiquitous technologies, such as cell phones, have also served as a platform for public sound art [4].

However, for the purposes of this study, we restrict ourselves to a conventional definition of public art and examine the challenges posed by the realization of physically situated works that are directly accessible and freely available to public audiences. Public art in this scope is housed outside of traditional art settings and is intended to engage a public audience that might not otherwise seek art experiences. Furthermore, as this work does not live in a virtual or broadcast context, it is dependent on site-specific circumstances in the physical world.

There has been extensive prior work with art that seeks new forms outside traditional contexts. For example, Chico McMurtrie's recent work, *Growing Rain Tree*, serves as an example of engaging kinetic public art that speaks to a broad audience and integrates expressive technologies [6]. The work has proven to be robust, but it does not face the challenges of an outdoor exhibition environment.

Sound artist and composer Max Neuhaus has undertaken a number of outdoor sound installations [8, 7]. While these pieces generate environmentally sensitive sound experiences in outdoor contexts, they do not provide a mechanism for audience interaction.

Artists such as Ned Kahn have utilized sound as a medium in permanent installations [5] that leverage naturally occurring sound production mechanisms. This work has been influential in our own conception of approaches to public sound art. However, this work is not audience interactive and does not address the logistical challenges posed by the use of electronic sound production mechanisms.

In our own recent work we have realized installations that utilize sound, digital media, and embedded electronics [2, 1]. Although this work has been exhibited in diverse contexts including galleries and indoor public spaces, it was not conceived for a public art context and does not address those unique challenges.

This paper describes an approach to physically situated public art that requires the design and realization of new frameworks for interactive music making. In Section 2 we identify challenges for the design and realization of interactive public sound art. Section 3 examines a case study of these points through our recent work, *Transition Soundings*. Finally, we evaluate our results and provide conclusions.

2. PUBLIC ART CHALLENGES
The realization and presentation of interactive sound art in the public sphere poses important challenges for artists.

Sound art is finding increasing acceptance in the culture of galleries and museums as artists, curators, and directors have sought to find new strategies for the presentation of this work. Nonetheless, even in the relatively controlled atmosphere of these venues, audiences can find sound art to be a challenging experience as it undermines many of the norms and conventions of exhibition attendance. Furthermore, interactive art has proven to be sometimes vexing and frustrating to even the most experienced audiences in a museum [3]. In this light, exhibition conditions outside a traditional gallery setting can be viewed as having distinct advantages and disadvantages. On the one hand, audiences will not be bound by expectations of traditional venues. On the other hand, the artist can make no assumptions about the audience's familiarity with the domain of interactive sound art, and artists must generate work that is sensitive to this context.

Public art exhibitions in outdoor environments demand an acute environmental awareness. Sound art work in particular must be embedded in the sonic context without overwhelming the existing soundscape. Extensive work has been undertaken to identify sources and attributes of noise pollution in our communities [11, 12]. This research reveals that society's perception of the role of the source can play a large part in the identification of sound as a nuisance. Public sound art, as an unusual element in the environment, must be particularly sensitive to both noise levels and qualities of sound to avoid becoming a source of distraction or annoyance for the public.

Public art must be designed and realized with careful attention to unique exhibition dangers. Specifically, weather and vandalism pose risks to this work, and electronic components are particularly vulnerable to both factors. Legal liability issues including public safety and regional building and electrical code requirements are also brought to bear on art presented in a public area rather than a gallery.

Finally, artists must address important logistical challenges including maintenance, durability and longevity of their work. Public art administrators typically expect works to have a serviceable life that is measured in years rather than weeks. This presents a unique challenge for interactive electronic works, given that many emerging technologies are still in their infancy and can require close monitoring to assure reliable operation. Careful consideration must be given to material selection and design in all respects to ensure an operable life of the work with minimal maintenance requirements.

3. *TRANSITION SOUNDINGS*: A CASE STUDY

Over the course of nine months during 2005, we designed, built, and installed a public art work, *Transition Soundings*, as a commissioned piece for the City of Tempe Cultural Services. Our piece is part of a larger series of works through a commissioning project entitled Artist Adorned Transit Stops. The work is currently installed at a bus transit stop in Tempe, Arizona. The work is a dynamic, interactive sound installation that is environmentally sensitive and is rooted in metaphors drawn from transit networks in our community. In this section we use this newly realized work as a case study to examine challenges and strategies for interactive public sound art.

3.1 Description of Audience Experience

As a user approaches the transit stop, they find a large wall positioned behind the waiting bench, ten feet wide and extending from just above the ground to over their head. The face of the wall is reminiscent of a typical transit map as it is multi-dimensional, attractively colored, and contains a network of elements that are arranged in clusters and paths. However, this wall houses a large matrix of sound producing elements and nearly fifty embedded sensors that allow for interactive sound-making exploration.

As the user comes in closer proximity to the face of the wall, their presence is detected, and a burst of sound ignites from the spot mirroring where they are standing. Just as a stone dropped in a pond will ripple waves across its surface, the sparkling sound washes across the face of the wall, shifting in color and tone as it emanates from the original source in all directions. As the transit user moves in front of the wall, freely exploring the way in which hand-waving and movements toward and away from the wall yield different sound shimmers, another user approaches the stop. This second user approaches the wall and the two play together, exploring how their movements can trigger simultaneous musical patterns that sparkle across the sound wall and heighten their awareness of the urban soundscape that previously went unnoticed.

Figure 1. *Transition Soundings* onsite

Later in the evening, this same transit user returns to the stop, and approaches the piece again. Remembering the interaction of earlier in the afternoon, the user waves their arms in front of the sound wall. Sound bursts still ripple across the surface of the wall, igniting from the locations where their arms travel. However, as darkness has now fallen and the sounds of Tempe have shifted, so too have the sounds of the piece. Now clicks and rushes emerge from the wall as they glide across its surface and into the sounds of the night.

The work seeks to engage transit stop users on two levels. First, the work encourages them to be active, up off the bench and moving - engaging with both the work and with one another – as they explore the sonic possibilities of the piece through playful interaction. Second, the work encourages them to reflect upon larger issues regarding the interconnectedness of our communities, the role of transit in uniting us, the role of technology in our society, the sonic world around them, and the importance of environmental sensitivity.

3.2 Motivation and Aesthetic Ideas

In the conception of an interactive installation for public exhibition, we were mindful of the challenges of this environment. In addition, we sought to create an experience which is immediately communicative and engages a broad public audience. As a commission relevant to transit networks in an urban desert environment, the work is motivated by concepts that are pressing for the public community it is intended to serve. Specifically, we engage the transit network, environmental concerns, and site-specific circumstances.

Relevance to transit system

The proposed work is inspired by the form and function of transit networks. This inspiration is present in the visual layout of the piece, and in the way that sounds propagate across the network, linking remote portions of the sound wall through hubs and direct routes of travel. We hope that our re-imagination of the transit network encourages the public to reflect on the ways in which transportation networks build links across our physical world, and how we can better unite our communities through networks formed of social, cultural, and intellectual ties.

Environmental relevance

Our work seeks to address issues that are central to urban communities that are situated in the desert environment. It is also inspired by the metaphor of water, and the interaction behaviors of the piece mimic the way in which ripples will emanate from a disturbance in the water. We sought to generate a sonic and technology-based equivalent of this visceral and universal experience that references water resource concerns in the metropolitan area and the western United States.

The sun is a dominant environmental feature in Tempe, and a highly practical source of power. Although we could have arranged access to the urban grid, we chose to integrate solar energy panels in our work to raise awareness of renewable, reusable energy sources. Clearly this energy source serves as a logical and sustainable solution for the requirement of powering the embedded electronics.

Site-specific relevance

We sought to integrate site-specific sonic and geographic considerations into the design for *Transition Soundings*.

The installation site is oriented along a major east/west thoroughfare. We have implemented a topological design that mirrors the growth of the greater metropolitan area of which Tempe is a part. Just as the areas to the west of Phoenix are more densely populated than those to the east, we have created a greater density of sound resonators toward the west side of the sounding wall that will thin as sound travels to the east.

Given that sound plays a central role in the piece, we worked to develop and refine a sound language that both enriches and reflects the ambient soundscape of the site. We first spent time listening and analyzing the sonic characteristics of the site, and physically tested prototype sounds in the environment to ensure that the sounds were attractive and meaningful. In addition to the musical notes that were present in the submitted prototype, we developed a broad palette of sounds including chirps, rhythmic clicks, flutters, and abstract sound events reflecting the diversity of sources in the urban environment. The sonic palette of the work is described in detail in *Section 3.3.3*.

3.3 Design and Realization

At the heart of the piece is a sounding wall six feet high by ten feet wide that houses a network of sensors and sound producing elements. This wall is comprised of twenty-seven modules that have the same basic design and function. There are two types of modules that differ only in the layout of their speakers as either hubs (clusters) or straight paths. Each module contains two proximity sensors, one light sensor, ten piezo-electric speakers, and one microcontroller computer (Atmel Mega 8) with supporting circuitry for sensing, sound output, and networking. Adjacent modules are connected to one another such that they form a network across the entire wall that allows sounds to wash across its surface following the structure of hubs and paths. In this section we describe the realization of this work and discuss our solutions to the challenges posed by public art.

3.3.1 Physical Implementation

Transition Soundings is installed in an outdoor environment that required us to be particularly mindful of the challenges of vandalism, harsh weather conditions, and maintenance concerns in designing the physical body of the work. In addition, we wanted to create a work that is visually inviting to audiences as well as functionally interactive.

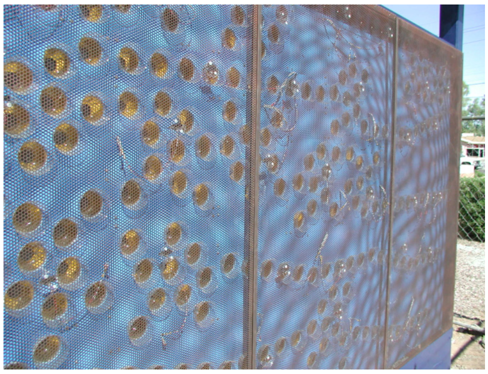


Figure 2. Close view of interactive sound wall onsite

Vandalism - To protect against unauthorized access to the inner electronics, we created a layered design that conceals most structural fastening points and exposes a minimum of security screws and locks. The outer layer of the facade is perforated steel, with small enough holes to prevent most types of damage by potential vandals. Sensitive elements are some distance from the front of the enclosure, while the perforations allow sound to easily emanate from the speakers and provide visibility to the sensors and resonators embedded on the inner panels.

The solar panel is one of the most expensive single elements in the piece. To guard against theft and vandalism, this panel is mounted well out of reach of passers-by, and is seated in a solid steel frame with security bolts. This design is similar to numerous solar panel mounting schemes employed on devices throughout the city.

Weather - Rain poses a risk to any outdoor installation, especially in the later summer months. The front panel, while perforated, has enough solid surface to combine with the shelter provided by the top-mounted solar panel and deflect most rainwater from entering the piece. The resonators, and their mounting panels, provide full weather resistance to any water that does pass through the front panel, and all electronics are safely mounted behind this barrier. Any rainwater that splashes off the inside panels is harmlessly routed downwards and drained past the waterproof battery enclosure and out the bottom of the structure. Each proximity and light sensor is protected by its own transparent weather enclosure that is easily replaceable and mounts directly to the face panel. The back access panels of the piece are fully sealed when closed.

Heat is a special problem in the Phoenix area, where summer temperatures frequently exceed 100°F and may reach 115°F or more. All of the electronic components in the work are guaranteed by their manufacturers to function in temperatures up to 135ºF. In practice, components can often withstand temperatures in excess of this mark. We have also temperature tested the selected plastic resonator cups and have determined that they retain their integrity at temperatures up to 150ºF. As a consequence, it is only necessary to keep the internal temperature of the enclosure at the same level as the ambient environment. By designing weatherproof vents in the enclosure at the bottom and top, we can evacuate excess heat from the electronics by simply creating an upward draft.

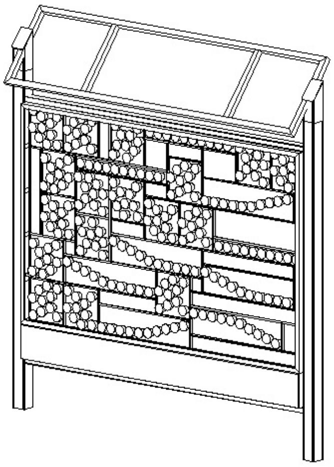


Figure 3. Diagram of physical structure

Materials for the structure have been chosen with heat, rain, and ultraviolet resistance in mind. We selected solar charging panels that are intended for exposure to the elements. They are similar in structure and function to solar panels that are currently in use in city traffic and lighting structures at many locations in the southwest United States.

Maintenance

The electronics of the work are designed in a modular fashion such that even if an element is damaged, it does not jeopardize the overall function of the piece. If one of the electronic modules should fail for any reason, it can be readily extracted and replaced. This internal wall of sensor/speaker circuits and controllers is enclosed by three solid back panels that keep the electronic elements out of sight and reach of the public. These panels as well as the front panels are hinged so that complete access can be had to any part of the piece.

Funds were budgeted to cover repair and replacement of any damaged components. During the initial six-month installation period, we have scheduled weekly visits to the site to inspect for any damage, and should the installation be extended we have a plan for continued maintenance.

3.3.2 Interaction Design

The work's interaction framework addresses concerns of both audience engagement and environmental sensitivity. A distributed network of proximity and light sensors blankets the front panel of the installation. Each module has either one or two infrared proximity sensors with a sensing range of up to 80cm extending out from the front panel. The resulting sensor density is one sensor per 2 square feet on average, and proportional to the speaker density as illustrated in Figure 5. Each module also contains one photovoltaic sensor to detect ambient light levels.

In order to engage all types of audiences, we needed to develop an intuitive and expressive framework for interaction that is suited for adults and children of all levels of education and experience. Furthermore, we have sought to engage both individuals and groups of audiences by facilitating simultaneous and collaborative interaction. The work facilitates multiple interaction paths to ensure that complex and varied musical structures can emerge. For example, if a user makes a broad, full-bodied sweeping gesture across a section of the wall, it will erupt with a diverse sound explosion that similarly races across the wall. Conversely, if a user instead holds a stationary hand in one isolated spot, a small region of the wall will be activated as it cycles through a sequence of sound transformations.

We were also concerned with accommodating transit riders, who have little choice about sharing the environment of the piece and whose level of interest may vary widely for reasons completely external to the work. For this reason the wall will only sound if actively engaged by the audience.

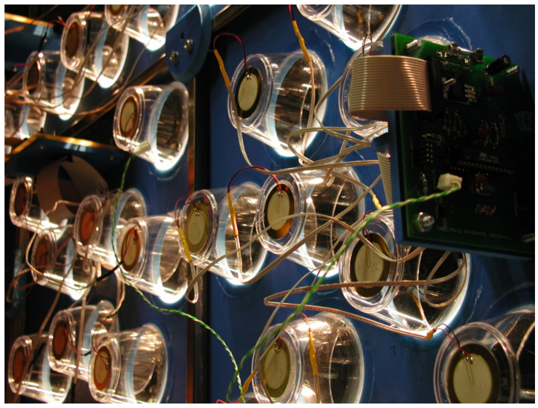


Figure 4. Partial inside back view of embedded electronics

3.3.3 Interactive Sound Generation

We have given particular consideration to two aspects of sound design for this work. First, we considered the ambient sound levels of the site to ensure that our work would not pose a noise pollution hazard. Second, we worked to design a strategy for interactive sound that would yield complex musical results with sounds and timbres that would engage a broad public audience.

Sound Levels

Given that the work is in an open environment, we first considered the sound design in light of the existing site-specific soundscape, to ensure that our piece would not dominate the ambient environment. The installation site is at the corner of a major intersection that has heavy traffic during peak hours and high variance due to the passage of cars and trucks according to cyclical traffic patterns. We first took sound level measurements at the site in both the afternoon (12:30PM on a Tuesday) and in the evening (8:30PM on a Tuesday) and found the following conditions

60-80 decibels (day) and 55-65 decibels (night)

We then measured a prototype sound module and found a rise above the ambient sound environment for the sounding wall when active is:

12 decibels at proximity of 2 ft. (simulating a listener who is interacting with the piece)

7 decibels at proximity of 8 ft. (simulating a listener at the transit stop, but not necessarily interacting)

2 decibels at proximity of 15 ft. (simulating a listener who is not at the transit stop)

These sound level readings reveal that the sound of the wall is well within the range of the ambient and traffic sound levels of the sites under consideration. The sound wall is audible for transit stop users without being intrusive, and does not compete unpleasantly with the ambient sounds of the site. In addition, it is evident that the wall is not a source of noise pollution for the community around the stop as the work is very quiet at a distance and is only audible when a user is interacting with it.

Interactive Musical Expression

Given the challenge of designing an interactive sonic work for a broad audience of users, we have endeavored to create interactive musical structures that are intuitively meaningful to listeners, and indicative of the nature of transit networks and the surrounding environment. Although we wanted to provide diverse and variable musical paths through the work for transit passengers who might visit the site on a daily basis, we also wanted to ensure some measure of shared experience for the broad spectrum of the public who might experience the work. All sound for the piece is generated via 1-bit digital to analog conversion using the embedded Atmel microchips. Algorithms for sound generation and DSP functions are programmed directly on the microchip.

First we discuss the behaviors of sound within an individual module. As described in *Section 3.1*, as a user moves or gesticulates in front of any of the proximity sensors, a wash of propagating sound originates from the location of the triggered sensor. Because each module functions independently, sounds can originate in parallel as multiple modules can be simultaneously activated by a user.

We have designed four classes of sound events:

(1) sustained square wave
(2) pitched pulse train with three-eight pulses
(3) clicking pulse train with three – eight clicks
(4) noise with a subtle central pitch

These sound classes were designed to integrate and reflect on the sonic environment and abstractly narrate a conceptual sonic day at this site. Combinations of harmonic square waves (Class 1) embody the notion of a bright, dawning day. Pitched pulse trains (2) reflect the vibrant activity of the transit network at peak hours. Clicking pulse trains (3) link to the evening hours when insects begin to emerge. Finally, the pulsing noise bursts (4) fuse to sound like cicada colonies on a summer evening.

There are ten stages for each of these events that dictate the frequency and duration of an event. For example, stage 1 of a pitched pulse train specifies frequencies of 200Hz with four short pulses. Stage 2 specifies frequencies of 400Hz with seven short pulses. When a given proximity sensor is triggered, the sonic event for that module will propagate through each of the ten ordered speakers. For example, a short noise burst can originate in speaker 0 of a straight line. Identical noise bursts will sequentially emanate from speakers 1-9. With each trigger of a given proximity sensor, the module stores an interaction history counter that flags the number of triggers. As a module moves from zero to a maximum of forty triggers, the originating sound will move through a fixed sequence of sound events, frequencies and durations. After a period of three minutes, each module will reset this history flag so that new users who approach the work will find an empty history. This timing value was selected to approximate an average transit wait period. This reset assures a relatively shared experience for all users and serves as a practical function to provide robust performance.

Now we describe how sound propagates through the network. Module junction points occur where the topology of the speaker network has 'touching' speakers. See Figure 5 for a diagram of speaker module and example junctions. As a sound travels within an individual module, when the sound reaches a junction, via an embedded serial bus, a ping will be transmitted to the adjacent module. The receiving module will in turn propagate the sound in the same fashion. In this way, sound will mushroom from the point of origin across a region of the wall. A radius feature that depends on the interaction history counter limits the number of linked modules that will carry the originating sound. Specifically, when a new sound class is revealed, the radius will range to a maximum of four. With each additional interaction this radius diminishes until the radius is at a minimum of one. The variable radius allows for a mixture of local, regional, and global sonic events that emerge depending on how users engage the interactive wall.

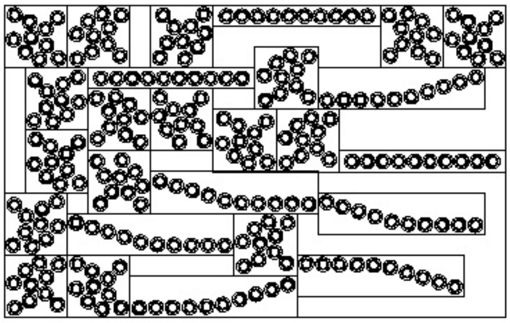


Figure 5. Diagram of front face speaker network

This propagation algorithm yields diverse musical events as users interact through varied gestures in different regions of the wall. Furthermore, the musical experience of the wall will vary greatly depending on whether individuals or groups engage it. In addition to impacting the sheer number of sonic events, each sound class event yields different combinatorial structures. For example, when a square wave sound is passed from one module to another, the frequency of the sound will select from a frequency array. Frequencies are organized into regions that yield simple harmonic relationships in groups with some dissonant transitions that yield moments of musical tension and release through interaction. This structure is simple enough to be readily perceived by novice users, but the interaction algorithm provides numerous paths to explore varied musical outcomes. Similarly, as clicking pulse trains pass from one module to another, the speed and number of clicks is varied to yield gradually accelerating and decelerating events.

The sound generation algorithm is also sensitive to the ambient light levels. When bright light is detected by the photovoltaic sensor, the interaction history advances slowly through the four sound classes. However, when light levels are dim or dark, the progression from pitched square waves to noise bursts is very rapid. This design allows for a varied experience for users who approach the work in the morning or night. The algorithm thus highlights the sounds associated with the appropriate time of day. For example, in darkness, users will quickly hear noise bursts that approximate the sound of night insects after even a brief interaction.

4. EVALUATION

Transition Soundings was installed in the city of Tempe in September, 2005 and is currently scheduled to remain on exhibit through April 2006. A potential extension of the installation period will be negotiated at that time. We have described a number of challenges posed by public art and have discussed our approach taken in the realization of this work. Here we assess the outcomes.

A critical challenge is to design work that is communicative and engaging for a diverse public audience that is not necessarily seeking an art experience. Given the open nature of the exhibition circumstances, audience evaluation is a challenging task. However, through both informal user studies with our colleagues and direct observation of public audience participation, we can draw some preliminary conclusions. Our initial user studies proved very encouraging. Our colleagues were able to approach the work and intuitively interact through movements and group gestures. Their comments indicate that the resulting sonic washes are timbrally rich and that the spatial sound movement is engaging. They report that the pulse train sounds are most appealing in this context, but that the harmonic richness of the cascading square waves is very rich. Most are intrigued by the sonic fusion of the noise bursts, but few users engaged the work long enough to reach that stage without additional direction.

As of this writing the work has not yet been widely publicized and thus our evaluation of a larger audience sample is inconclusive. Our brief observations reveal that users often do not realize that the work is interactive, and we suspect that our attempts to avoid problems of noise pollution have lead to a situation where the work does not fully attract public users to engage the work. Furthermore, as work is situated on a busy street, we postulate that users might feel self-conscious about engaging in extended interactions.

We took great care to design sonic feedback that is site-specific. We have been pleased with the overall sound design and have received positive feedback from users regarding the relationship of the musical structures. However, while the work is in no way a source of noise pollution, we have been disappointed that it is overwhelmed by the ambient sounds and traffic noises that define the site. During the evening hours the overall sound level of the piece is ideal, but traffic noises mask much of the sound during peak hours and users are unable to decipher many details of the work.

A critical design challenge in our work was to address issues of durability, maintenance and reliability in the work. After several months of uninterrupted display we are extremely pleased with the robustness of the work. Our solar charging power plant provides reliable power throughout the day and night. We have not experienced any failure of electronic components. Despite several rain storms and above average fall heat, we have not observed any ill effects. Finally, we have not observed any effects of vandalism or tampering with the work.

5. CONCLUSIONS AND FUTURE WORK

We have described the challenges posed by physically situated interactive public sound art works, and have discussed these points within the context of our development of a new public art work, *Transition Soundings*. We have been pleased with the outcome of the interaction framework and the strength of the conceptual underpinnings for this project. Our funding partners have expressed great interest and excitement about the innovations of the work and its uniqueness within the field of public art.

We expect that the piece will ultimately greatly benefit from location in a less active sonic environment. In our future work, we plan to find a permanent installation site in our region that is publicly accessible but sonically more appropriate. For example, we have identified a number of transit stops that are situated along local streets that are less heavily traveled.

6. MEDIA DOCUMENTATION

Extensive documentation of the design, development and final realization of the piece can be found online at http://ame2.asu.edu/faculty/dab/transitionsoundings.php. Still images, video clips, and diagrams are published that detail each stage of the design process and outcomes.

7. ACKNOWLEDGEMENTS

We are grateful for generous support of the City of Tempe Cultural Services and the Arts, Media and Engineering program at Arizona State University. Their funding and support has made this work possible.

8. REFERENCES

[1] Birchfield, D., "Interactions: an interactive multimedia installation", ACM Multimedia, Singapore, 2005.

[2] Birchfield, D., Lorig, D. and Phillips, K., "Network Dynamics in Sustainable: a robotic sound installation", Organised Sound, 10 (2005), pp. 267-274.

[3] Boxer, S., "Art That Puts You in the Picture, Like It or Not", New York Times, April 27, 2005.

[4] Freeman, J., Telephone Etude 1: Shakespeare Cuisinart, http://www.jasonfreeman.net/Catalog/electronic/telephone1.html, 2001.

[5] Kahn, N., Wavespout (Breathing Sea), http://nedkahn.com/water.html#wavespout, 1993.

[6] McMurtrie, C., Growing Raining Tree, http://www.cronos.net/~bk/amorphic/machines_tree.html, 2003.

[7] Neuhaus, M., Time Piece Graz, http://www.max-neuhaus.info/soundworks/permanent/, 2003.

[8] Neuhaus, M., Times Square, http://www.diacenter.org/ltproj/neuhaus, 1977-1992.

[9] Neuhaus, M., Freeman, J., Ramakrishnan, C., Varnick, K., Burk, P. and Birchfield, D., Auracle: Live interactive sound over the web, http://auracle.org, 2004.

[10] Ramakrishnan, C., Freeman, J., Varnick, K., Birchfield, D., Burk, P. and Neuhaus, M., "The Architecture of Auracle: A Real-Time, Distributed, Collaborative Instrument", New Interfaces for Musical Expression, Hamamatsu, Japan, 2004.

[11] Schafer, R. M., The Tuning of the World. Random House, New York, 1977.

[12] Truax, B., Acoustic Communication. Ablex Publishing, 2001.

Building Collaborative Graphical interFaces in the Audicle

Ge Wang
Department of Computer Science,
Princeton University
35 Olden St.
Princeton NJ 08540 USA
gewang@cs.princeton.edu

Ananya Misra
Department of Computer Science,
Princeton University
35 Olden St.
Princeton NJ 08540 USA
amisra@cs.princeton.edu

Perry R. Cook
Department of Computer Science
(also Music),
Princeton University
35 Olden St.
Princeton NJ 08540 USA
prc@cs.princeton.edu

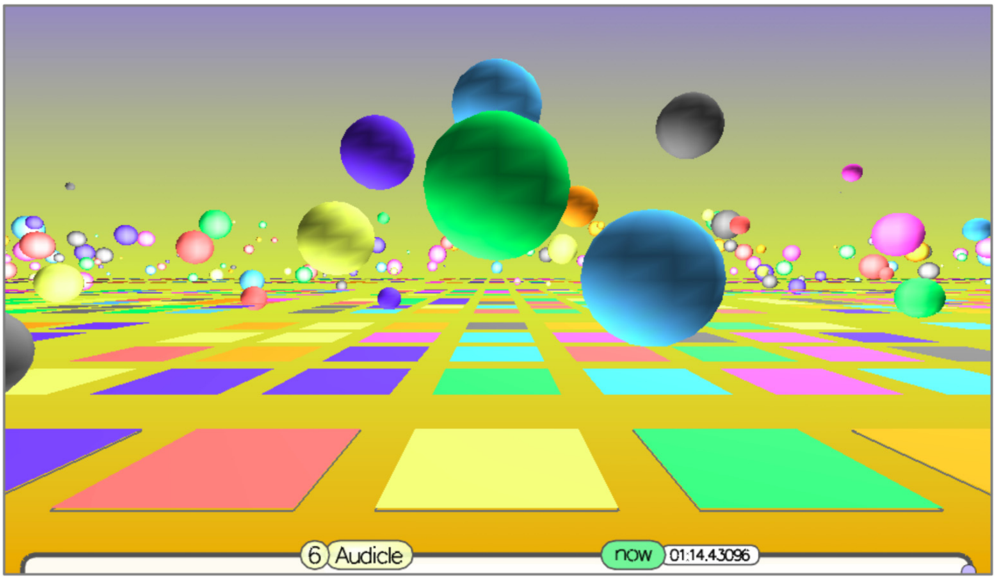


Figure 0. Multiple Bouncing Spheres interfaces visualized from a centralized viewpoint. Each human player manipulates spheres over a portion of the squares. The ensemble is synchronized by computer.

ABSTRACT
Emergence is the formation of complex patterns from simpler rules or systems. This paper motivates and describes new graphical interfaces for controlling sound designed for strongly-timed, collaborative computer music ensembles. While the interfaces are themselves minimal and often limiting, the overall collaboration can produce results novel beyond the simple sum of the components – leveraging the very uniqueness of an ensemble: its strength in numbers. The interfaces are human-controlled and machine-synchronized across a dozen or more computers. Group control, as well as sound synthesis mapping at each endpoint, can be programmed quickly and even on-the-fly, providing a second channel of real-time control. We show examples of these interfaces as interchangeable plug-ins for the Audicle environment, and also document how they are used in a laptop ensemble.

Keywords
Graphical interfaces, collaborative performance, networking, computer music ensemble, emergence, visualization, education.

1. INTRODUCTION
Emergence is the formation of complex patterns from simpler rules or systems. In this paper, we explore minimal, easy-to-learn graphical interfaces that can, as a group, form sound and music that is more interesting and complex than that made by any single component, in a tightly coupled and collaborative environment.

This was motivated by the desire to provide new interfaces for new computer music performance ensembles such as PLOrk: Princeton Laptop Orchestra, currently being developed and instructed by Dan Trueman, Perry Cook, Scott Smallwood, and Ge Wang. In addition to more self-contained, sophisticated instruments, we wished to provide the ensemble with interfaces that require minimal setup and learning time, and with which the participants can immediately and directly influence the overall sound as part of the group. Furthermore, we wanted the option of tightly synchronizing all participating interfaces / machines.

Given this motivation, the research goals are defined as follows.

The interfaces should be:

- simple enough to pick up and use, yet complex enough to generate interesting music/sound as a group
- amenable to collaboration in a tightly-timed setting; for example, a server should be able to synchronize all interfaces with desired musical timing; collaboration is the unifying aspect of all the interfaces we present.
- as direct and as immediate as possible
- as precise as possible, even at the cost of resolution
- easily programmable (i.e. mapped to sound/graphics)

To implement the interfaces, we used the Audicle programming environment [6,7] as the platform, leveraging its existing framework for blending high-performance graphics with precise real-time audio. Extending the Audicle API (in C++), we were able to add and experiment with new graphical interfaces as new faces of the Audicle. We also added a mechanism for accessing and controlling the interfaces directly using ChucK [5], the language for which Audicle was built. In this way, we can write ChucK code to control sound synthesis using data from the graphical interfaces, decoupling the interface from the sound synthesis. Furthermore, ChucK makes it possible to change sound synthesis and interface mapping on-the-fly.

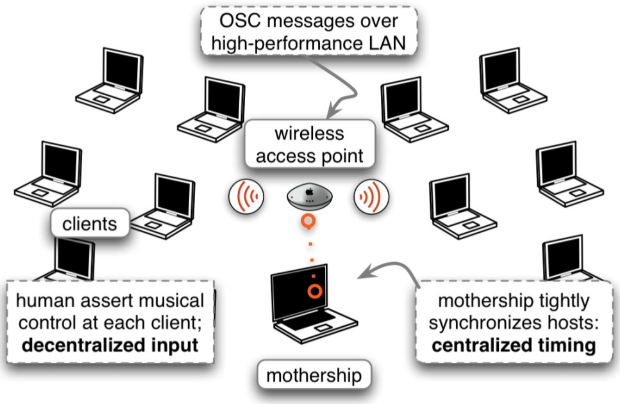


Figure 1. A collaborative interface network model.

The interfaces are synchronized over a wireless local-area network (Figure 1), using Open Sound Control [9]. One or more *mothership* host runs any application that broadcasts messages and synchronization messages to each of the end points. Our current mothership programs are written in ChucK.

2. RELATED WORK

The various graphical interfaces developed in this ongoing investigation derived mainly from three areas: audio/visual interfaces (Figure 2), such as the Painterly Interfaces created by Golan Levin [3] and musical video games created by Toshio Iwai [2,10]; GUI-based frameworks such as the MIDIGrid [1] and ixi software [4]; mainstream puzzle games, such as Chu-chu rocket, Lemmings, and Domino Rally [11,12,13].

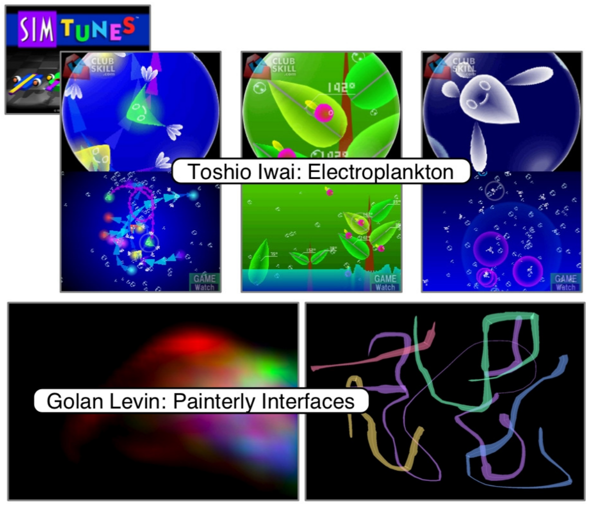


Figure 2. Audiovisual interfaces.

All of these works have significantly influenced and inspired interfaces in this paper – directly (such as in the case of the MIDIGrid) or aesthetically. The contribution of our work is placing interfaces similar to these within a tightly-timed synchronization framework, and finding paradigms to leverage this new collaborative aspect.

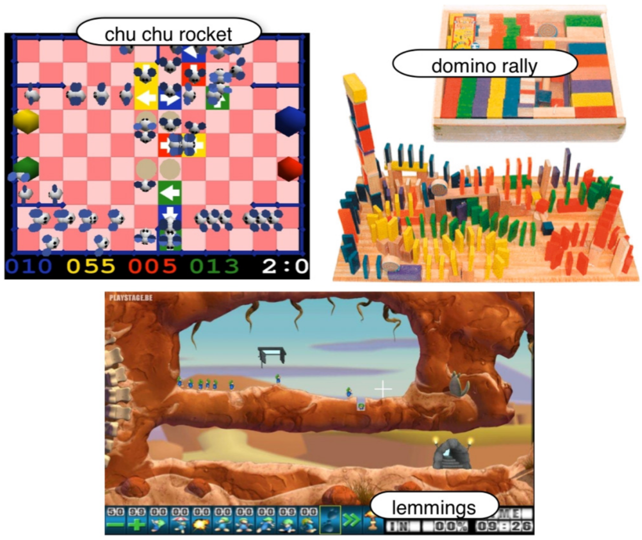


Figure 3. Puzzle games.

3. INTERFACES

Some example interfaces designed for single-server multiple-client setups are described below. As shown in Figure 1, humans assert musical control at each client, while the server centrally synchronizes all the clients.

3.1 Color Squares

In this interface (Figure 4), each client has an N x M grid and a finite color palette. Every color can be mapped to a separate event or to silence, with a potentially different mapping for each client. The user at every client machine selects a color using the mouse or the keys 1-9 and a-z (0 and es are reserved for the silent color), and applies it to any number of squares on the grid.

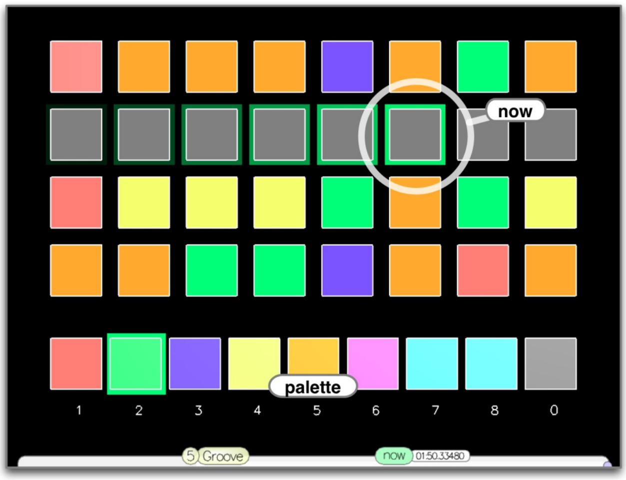


Figure 4. Color Squares

The server program moves through the grid in any desired manner, including sequential, random, or in any other fashion, in any timing pattern as specified in code in the server ChucK program. The client is aware of when a particular grid element is activated and what color was last placed on it, and can respond to this information in any programmatic manner, such as by playing a sound associated with the color of the currently activated grid element. Sample ChucK code for the server and client sides are presented here.

3.2 Spheres

Spheres (Figure 0) is a three-dimensional interface extending the color squares metaphor. It consists of spheres that bounce from a height, with or without energy loss, and colored square covering the "ground". Each time a sphere hits the ground, it can trigger an event based on the color and mapping of the square it touches. Multiple views of the space allow the user to control where a sphere bounces as well as its starting height. The bouncing location (or square) controls which event the sphere triggers, while the height translates to how often it bounces and thus how often the triggered event is repeated.

3.3 Mouse Mania

The Mouse Mania interface draws from the Chu Chu Rocket game described earlier. Each client or host has a virtual entity or "mouse" and a grid that acts as a map for the mouse to follow. Each grid element can have a user specified color and shape. As in the Color Squares interface, the color of a grid element can be mapped to an event that is triggered when a mouse reaches it. In addition, the grid element's shape can control the subsequent movement of the mouse, including its direction, speed, or special dance moves possibly associated with repetition of the related musical event. A mouse need not be confined to a single host; another option is for the server to own many mice that run from host to host, changing the spatial dynamics of the piece.

3.4 Dominotes

This interface uses the visual metaphor of dominoes to control events being triggered in rapid succession. Each user constructs domino rallies and connects sub-rallies to a central launching station when they are ready to be played. The launching station, at the discretion of the server, pushes all adjacent or connected dominoes, triggering a chain reaction. Each domino toppling, as well as special items such as rockets in the dominoes' path, can be mapped to any parameters or events at the discretion of the client. Toppled dominoes can be made upright automatically or manually by the users' selecting any subset of their dominoes. Forks in a domino rally allow each client's musical sequence to follow multiple paths in parallel.

3.5 SaladTossers

This interface is based on the idea of musical "recipes" and consists of salad ingredients, dressing, and a mixing bowl for each client. Ingredients can map to musical events as specified by the client. The user creates a salad by inserting desired quantities of each ingredient into the mixing bowl and tossing it. The tossing causes events to be triggered repeatedly; events associated with ingredients that make up a larger portion of the salad are triggered more often and thus have greater density in the resulting sound. As more ingredients are added to the salad, events are triggered more often. Further, a variety of dressings are available for adding to the mix, each dressing being associated with a different filter or sound processing effect. Finally, there is a "consume" option which gradually empties out the contents of the bowl and thus reduces the overall event density until there is silence. This interface is expected to be especially useful for creating textures where one may prefer to closely control the density of events rather than specifying the exact times at which events are triggered.

3.6 More

The above are some examples of simple interfaces that can produce complex music over a network of collaborators. It is possible to program more such graphical interfaces using the open-source Audicle framework. In addition, the mapping suggestions and time-based behavior described above are optional for each graphical interface and can be easily modified by changing the ChucK code on the client and server sides. Thus, these interfaces are flexible on the visual and auditory levels as well as in the interactions between the two.

4. CASE STUDIES

The Color Squares interface was used in the debut concert of PLOrk: Princeton Laptop Orchestra in a piece called “Non-Specific Gamelan Taiko Fusion Band”. The setup involved one conductor, one mothership laptop, one inkjet printer (from which scores were printed during the performance), and 15 laptop stations, each equipped with powered hemispherical speakers and running Color Squares. The stations were divided into four groups, each with a different sound synthesis mapping.

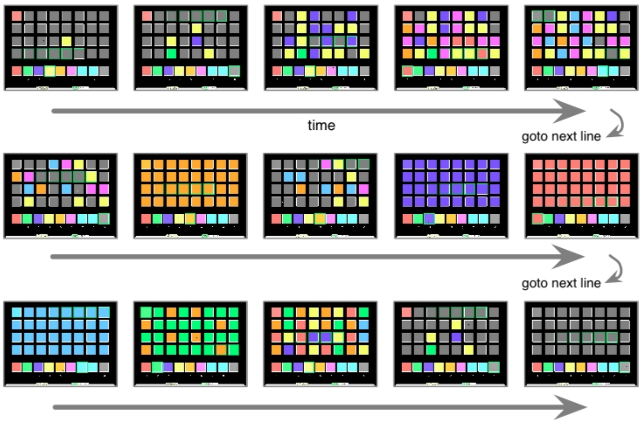


Figure 5. Example score over time.

During the performance (Figure 6), the conductor held up colored print-outs specifying differing densities and timbres (colors) to different groups over time. The score (Figure 5) shows some typical textures and transitions. The study successful demonstrated that the approach fulfilled its major goals. The students learned the interface in seconds and were able to immediately affect change in controllable ways. The resulting performance can be heard at:

http://plork.cs.princeton.edu/listen/debut/nsgamelan.mp3

Figure 6. PLOrk in action.

5. CONCLUSION AND FUTURE WORK

We have demonstrated a variety of graphical interfaces for creating music collaboratively. The simplicity of these interfaces allows new users to grasp the rules quickly and start making music right away. At the same time, the strong central synchronization facilitates collaboration, giving rise to more complex pieces than would be expected from the basic rules for the clients or server. Thus, these interfaces produce a form of *emergent music*. As our investigating continues, we hope to expand on this beginning exploration into collaborative graphical interfaces.

6. ACKNOWLEDGMENTS

We would like to thank Dan Trueman, Scott Smallwood, all the member of PLOrk, Philip Davidson, and Joshua Podolak for their invaluable help and ideas.

7. REFERENCES

[1] Hunt, A. and R. Kirk. “MidiGrid: Past, Present, and Future” *In Proceedings of the International Conferences on New Interfaces for Musical Expression*. May 2003.

[2] Iwai, T. “Images, Music, and Interactivity - the Trace of Media Art” *Keynote Speech. International Computer Music Conference*. June 2004.

[3] Levin, G. *Painterly Interfaces for Audiovisual Performance.* M.S. Thesis, MIT Media Laboratory, August 2000.

[4] Magnusson, T. “ixi software: The Interface as Instrument” *In Proceedings of International Conference on New Interfaces for Musical Expression*. June 2005.

[5] Misra, A., G. Wang, and P. R. Cook. “SndTools: Real-time Audio DSP and 3D Visualization” *In Proceedings of the 2005 International Computer Music Conference*. September 2005.

[6] Wang, G. and P. R. Cook. “The Audicle: A Context-Sensitive, On-the-fly Audio Programming Environ/mentality” *In Proceedings of the International Computer Music Conference*. Novermber 2004.

[7] Wang, G., A. Misra, P. Davidson, and P. R. Cook. “Co-Audicle: A Collaborative Audio Programming Space” *In Proceedings of the International Computer Music Conference*. September 2005.

[8] Wang, G. and P. R. Cook. “ChucK: A Concurrent and On-the-fly Audio Programming Language” *In Proceedings of the International Computer Music Conference*. October 2003.

[9] Wright, M. and A. Freed. “Open Sound Control: A New Protocol for Communicating with Sound Synthesizers” *In Proceedings of the International Computer Music Conference*. September 1997.

[10] http://electroplankton.nintendods.com/

[11] http://www.pressmantoy.com/

[12] http://www.vintage-sierra.com/puzzle/tim.html

[13] http://en.wikipedia.org/wiki/ChuChu_Rocket

The Frequencyliator – Distributing Structures for Networked Laptop Improvisation

Pedro Rebelo
Sonic Arts Research Centre (SARC)
Queen's University Belfast
Belfast BT7 1NN, Northern Ireland
+44 (0)28 90974829
p.rebelo@qub.ac.uk

Alain B. Renaud
Sonic Arts Research Centre (SARC)
Queen's University Belfast
Belfast BT7 1NN, Northern Ireland
+44 (0)28 90974829
alain.renaud@qub.ac.uk

ABSTRACT
The culture of laptop improvisation has grown tremendously in recent years. The development of personalized software instruments presents interesting issues in the context of improvised group performances. This paper examines an approach that is aimed at increasing the modes of interactivity between laptop performers and at the same time suggests ways in which audiences can better discern and identify the sonic characteristics of each laptop performer. We refer to software implementation that was developed for the BLISS networked laptop ensemble with view to designing a shared format for the exchange of messages within local and internet based networks.

Keywords
Networked audio technologies, laptop ensemble, centralized audio server, improvisation

1. INTRODUCTION
The desire to use machines to share musical data during a performance has been present since the earliest developments of computer technology. It not only suggests a reflection on the exchange of musical ideas between human performers but also an exploration of something computers became very good at: data transfer. The development of reliable high-speed networks and ever increasing computer power have made truly effective networked music events possible. We investigate the potential emergence of new forms for musical interaction that are a direct result of networked structures. BLISS (Belfast Legion for Improvised Sights and Sounds) [1]– a laptop improvisation ensemble together with other collaborators act as a "testbed" for the development of technologies for networked music ensembles (NMEs).

2. BRIEF HISTORY OF NMEs
John Cage's "Imaginary Landscape No. 4 for twelve radios" can be seen as an early NME experiment. The piece, "used radio transistors as a musical instrument. The transistors were interconnected thus influencing each other." [2] Although the levels of interactivity were limited to the dialing of radio-stations, gain and tone-colour, the desire to investigate the possibilities of cross-influence in networked instruments is evident in the piece.

NIME 06, June 4-8, 2006, Paris, France.

It is really with the development of microcomputers that more interactive NMEs began to emerge. From 1978 to 1982, The League of Automatic Music Composers originally composed of Jim Horton, Tim Perkis, and John Bischoff started using networked computers to exchange messaging data between each other with the goal of influencing their playing. The group, which used Commodore KIM-1 computers, developed ways to increase the level of interdependency between players by, for example, using frequencies in one computer to generate notes in another.

The technologies and playing techniques pioneered by the League of Automatic Music Composers were important since it was the first time that a group of electronic musicians attempted to include computers in a live music environment. The League represents a singular project in a period when computer music research was almost solely focusing on non real-time applications.

In 1987, the League of Automatic Music Composers became The Hub [3]. The ensemble included Chris Brown, Scot Gresham-Lancaster, Mark Trayle, Tim Perkis, Phil Stone, and John Bischoff. The principle of The Hub was also based on interdependency with a more elaborated system of communication such as a form of audience participation and the enabling of remote collaboration. Later on, the ensemble started using MIDI as the main communication protocol. The Hub, was initially the name of the central computer used to store and distribute messages amongst the performers. The group quickly developed a standardized interface with the aim of allowing any type of computer available at the time to connect to the network. In 1987, The Hub had its first networked performance through a collaboration with composers Nic Collins and Phil Niblock. A live performance took place between two spaces in New York which were linked by a basic modem connection through which messages were exchanged.

In 1997, the group started experimenting with remote performances over the internet although these presented significant technical challenges. The Hub still performs today on various occasions.

The existence of pioneering NMEs such as The Hub led to a wider spread practice in networked performance. The SoundWire project, led by Chris Chafe at Stanford's CCRMA [4] has developed a set of applications, which allow high quality networked performances over high speed internet. It is one of the first applications to take advantage of the bandwidth offered by next generation IP networks such as Internet2. The project has also developed ways to evaluate network reliably through the use of audio signals.

Standard internet infrastructure is increasingly being used for collaboration and interaction. This is the case of Prométhée

Numérique (2002), a radio art piece structured around a web-site which is accessible by remote web users [5].

The development of new message protocols such as Open Sound Control (OSC) allows much more flexibility for exchanging control messages between players. The flexibility provided by OSC as a "transport-independent, high-level application protocol" [6] is very suitable as a basis for the development of present and future NMEs.

Such technologies and practices has led to a new classification of various types of networked setups as illustrated by Barbosa [7]. These include: Local Interconnected Musical Networks, Musical Composition Support Systems, Remote Music Performance Systems and Shared Sonic Environments.

3. IMPROVISATION AND THE NETWORK

3.1 Performing Roles

In order to develop musical interactions within the context of networked improvisation, we have explored the dynamics between free or non-idiomatic performance [8] and pre-determined temporal structures. Whilst rejecting specific musical languages or idioms such as those characteristic of Jazz-based improvisation, free improvisation relies nevertheless on certain structures and roles. Amongst these are the constraints dictated by each musical instrument. A quartet consisting of drums, bass, piano and saxophone suggests certain relationships between the performers (which can of course be challenged and transgressed). These relationships are less apparent in a quartet composed solely of computer musicians. The desire to turn the computer into a universal instrument which is capable of all sounds at all times has produced a performance practice in which performative roles are not clearly identifiable. Although one can see this as a reflection of a wider social trend that has replaced the guitar hero with the cooler and certainly less visible DJ, the relative anonymity and dispersed responsibility that characterizes many computer-based ensembles could be seen as musically restrictive. While a solo laptop performer is capable or articulating dramatic shifts in texture, loudness or timbre, at literally the press of a button, once another performer is introduced, certain types of shifts become increasingly difficult to articulate. While instrumental performers often recur to visual cues and gestures for communication in a group context, the laptop performer rarely counts on this mode of communication to coordinate events. The posture of most laptop performers favors the intense gaze towards the computer screen at the expense of interpersonal cues. In contrast, instrument-based improvising ensembles have developed a performance practice which is very much based on the identification of roles (even though each musician will have a number of different roles during a performance), and in the communication of form through gesture and visual cues.

3.2 Musical Structures

We have attempted to develop an environment in which an ensemble performing solely on software instruments can resource to musical structures that are very much taken for granted in an instrumental ensemble. A non-exhaustive list would include the following:

- Potential for common pulse to emerge through rhythmic interaction.
- Synchronization of events that require negotiation and agreement from two or more performers.
- Balanced spectral structure with possibilities for both masking and extreme spectral separation.
- Identification of performative roles which suggest performer-to-performer as well as audience-performer interactions.
- Up beats or the ability to anticipate and precede events across the ensemble.

These structures can be identified in a number of improvisatory contexts (idiomatic or not) and often provide a collective platform for the development of musical form. The generic nature of the computer as a musical instrument often conflicts with the specificity that characterizes most musical performance situations. While maintaining a free improvisation approach to musical materials we have attempted to design a framework within which this "freedom" is constrained in order to facilitate certain types of performance interactions.

3.3 Networked Interactions

The musical structures identified above are difficult to recreate intuitively with NMEs due to the lack of common performance practice. Although the instrumental paradigm has been used to describe the software used in performance by a computer musician, it is rare that this software behaves like an instrument (unless it is designed to imitate an acoustic instrument). The constraints that define acoustic musical instruments are often unobserved in the development of a "software instrument". While this might offer the potential for novel instrumental roles, the desire for technological sophistication often obscures consideration for musical contexts. The limits in range, amplitude and articulation that characterizes acoustic instruments are replaced by flexibility, unlimited access to all soundfiles in a hard drive, and open-ended spectral behavior. Without wanting to disregard the possibilities that these software instruments suggest, it is worth considering the role of constraints in a situation that is often characterized by notional freedom.

The system has been used on a local network with four performers in concert. The first application is based on a score that has been written "offline". The score, which describes elements such as tempo variations, filter banks and bandwidth separation has been interpreted and entered into the system. This approach provides a central point of reference to the ensemble, which is not computerized.

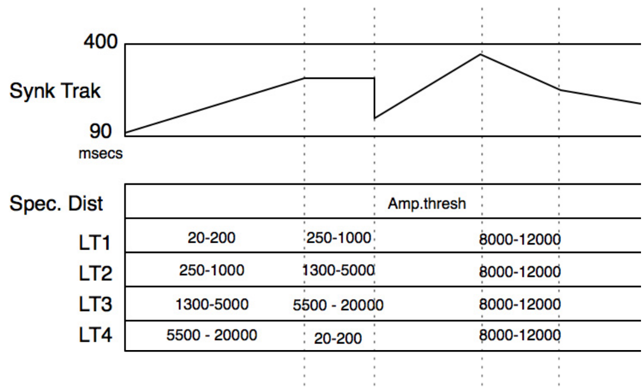


Figure 1: Excerpt from the score for bliss.net

Figure 2: The BLISS Ensemble Setup

4. THE FREQUENCYLIATOR

4.1 Concept

The Frequencyliator is a software tool being developed as the outcome of the initial experiments made with the BLISS ensemble. The concept is to create a framework for laptop improvisers to easily collaborate and exchange musical ideas over a local or remote networked setting. The primary goal of the Frequencyliator is to recreate such interactions through the implementation of a basic structure that is imposed to the ensemble. Basically, a server issues various messages to the ensemble, whilst each audio output is routed through the server for processing.

4.2 Design

The system is being developed with Max/MSP [9]. The application will include a series of message and/or DSP based modules:

- Shared timeline (Message)
- Bandwidth allocation mechanism (DSP)
- Countdown (Message)
- Sync event (Message)
- Spectrum analyzer (DSP)

All the modules will be connectable with each other on a local and global level. The messaging part of the system will be based on the Open Sound Control (OSC) protocol while the audio signals will use a low latency audio to UDP adapter such as the one developed by the SoundWire project [10]. Even though it is crucial to keep a server/client model to allow a master clock to act as a central point of synchronization for the ensemble as well as for connectivity reasons, the system will allow performers to send messages and audio content to each other. The software is designed around modules which implement a particular type of musical interaction. This interaction is then articulated through a function which can be pre-determined or generated in real-time, by an algorithm or manually by any of the performers. Each performer can create a new function which is then automatically suggested to the ensemble and approved or denied based on the percentage of positive and negative votes within a specific time limit. This promotes a basic yet dynamic way of exploring the notion of negotiation within an ensemble. Below is a more detailed description of the modules currently implemented.

4.2.1 Shared timeline

The Shared timeline is a central structure within the system. The timeline resembles a proportionate score that defines the form of a piece. Its generation can be pre-determined and implemented manually or generated in real-time by an algorithm. For example, a function can define variations in tempo whilst a secondary envelope can send frequency interpolation messages to a filter bank. Timeline messages can be sent to the entire ensemble (global) or to a selected member of the ensemble (local). Variations in timeline parameters can be suggested on an ad-hoc basis locally. The potential for common pulse is achieved through a common rhythmic structure, which is sent as messages to each performer, not unlike a conductor's beat. The timeline is implemented as a *qlist* MAX object. Each section of the qlist sends internal messages that influence the server's behavior (e.g. centre frequency and bandwidth for frequency distribution). External messages such as pulse and general section change messages are also included in the list.

4.2.2 Bandwidth allocation mechanism

This module allocates a given frequency bandwidth for each laptop plugged into the system. This is implemented with a range of filters that process the audio signal from each performer. Allocated bandwidth parameters are sent to each player as messages. The variation in bandwidth allocation will generally be controlled by the timeline but can be driven by interactive behaviors such as swapping bandwidths as triggered by amplitude thresholds. The filtering is achieved through the use of MSP objects *reson~* for bandwidth separation and *fffb~* for filterbanks. This mechanism allows for spectral structures across the ensemble (e.g. distribution of a complex spectrum through a series of filterbanks for each performer) as well as the identification of individual roles.

4.2.3 Countdown

The countdown mechanism is purely based on messages sent to each performer to allow for anticipation of events. A typical message would be a 10 second warning that a section of the piece is about to finish. This mechanism provides a reliable way to automatically warn each member of the ensemble when a new section is about to begin, usually meaning each performer will be assigned a new role within the ensemble, not unlike the transition from tune to solos in Jazz.

4.2.4 Sync event

This type of event is accessible by each performer and involves sending trigger events ("bang messages") either to the entire ensemble or to one or several carefully chosen performers. Each performer can choose to activate or de-activate the reception of such messages as well as the mapping to a local event.

4.2.5 Spectrum analyzer

This module detects and distributes characteristics of the audio signal of each performer. An FFT analysis on each signal captures partial distribution, spectral centroid, onsets as well as perceptual parameters such as brightness and noisiness through the use of Tristan Jehan's *analyzer~* MSP object [11]. The analysis data is formatted and shared amongst the ensemble, allowing for the synchronization of events on onsets, or "spectral grabbing" which consists of one performer grabbing an FFT frame from another signal in order to create similar spectral content. Information from this module can be used to influence the behavior of other modules (e.g. timeline). BLISS has used this module to detect amplitude thresholds as analyzed after bandwidth allocations. The crossing of these thresholds causes a change in bandwidth allocations, effectively resulting in a "stealing" of bandwidths. This type mechanism adds a level of interactivity within the ensemble that resembles game structures present in works such as John Zorn's Cobra [12].

4.3 Interface

A standard interface rendering the events being broadcast by the server displays each parameter to the performers. Each performer uses the Frequencyliator as a subpatch window that is added to their software instrument. The suggestions are sent on an ad-hoc basis by any given member to modify elements as the piece develops can be "dragged and dropped" into the structure of the piece via this interface.

5. CONCLUSION AND FUTURE WORK

This server-based approach to structured improvisation provides a rich platform for networked ensembles.

At the moment, the interface only allows a unidirectional communication of messages from the server to the clients, whilst the clients send the output of their signals back to the server. This limitation will soon be overcome by introducing bi-directional communication for both messages and audio data. This approach will most certainly lead to an increased level of interaction since it will allow, for a performer to send messages to another performer or even to the server.

The system will be optimized for remote participation. The frequency separation mechanism introduced in this context provides an excellent way to identify distinctively the signal output of each performer. We can therefore envisage the participation of other laptop performers from remote locations. This approach is technically possible as initial audio tests as part of a separate project between SARC and CCRMA have proven to be very conclusive.

Finally, another goal is to standardize the interface of the Frequencyliator and make it available to anyone who wishes to participate in such a performance.

6. ACKNOWLEDGMENTS

We would like to thank Chris Chafe for his dedication to making the network heard, and to the other members of BLISS involved in this project: Tom Davis, Jason Dixon and Chris McClelland.

7. REFERENCES

[1] Bliss Web www.sarc.qub.ac.uk/~bliss Accessed on January 23, 2006.

[2] Pritchett, J. The Music Of John Cage. Cambridge University Press, Cambridge, UK, 1993.

[3] Brown, C and Bischoff, J, Hub Origins http://crossfade.walkerart.org/brownbischoff/. Accessed on January 23, 2006.

[4] Chafe, C, Wilson, S, Leistikow, R, Chisholm, D and Scavone, G, A simplified approach to high quality music and sound over IP. In Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), (Verona, Italy, December 7-9,2000).

[5] Tanaka, A. *Composing as a Function of Infrastructure.* Errant Bodies Press, 2003

[6] Wright, M, Freed,A and Momeni, A, OpenSoundControl: State of the Art 2003. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, (Montreal, Canada, May 22-24, 2003

[7] Barbosa, A, Displaced Soundscapes: A Survey of NetworkSystems for Music and Sonic Art Creation. *LEONARDO MUSIC JOURNAL, Vol. 13, pp. 53–59, 2003.*

[8] Bailey, D. Improvisation: Its nature and practice in music. Da Capo Press, 1993

[9] *Cycling 74' MAX/MSP* http://www.cycling74.com/ Accessed on January 29, 2006.

[10] *SoundWire Project page* http://ccrma.stanford.edu/groups/soundwire/ Accessed on April 12, 2006.

[11] *Tristan Jehan's Max/MSP object page* http://web.media.mit.edu/~tristan/maxmsp.html Accessed on January 29, 2006.

[12] Zorn, J. *Cobra* Tzadik Cat. # 7335, 2002

A VR Interface for Collaborative 3D Audio Performance

Martin Naef, Daniel Collicott
Digital Design Studio,
The Glasgow School of Art
Glasgow, UK

[m.naef | d.collicott]@gsa.ac.uk

ABSTRACT

This paper presents a novel interaction paradigm to support musical performance using spatial audio. This method reduces the interface bottleneck between artistic intent and spatial sound rendering and allows dynamic positioning of sounds in space.

The system supports collaborative performance, allowing multiple artists to simultaneously control the audio spatialization. The interface prototype is built upon standard virtual reality software and user interface technology. Tracked data gloves are used to manipulate audio objects and stereoscopic projection to display the virtual 3D sound stage.

Keywords

Spatialized audio, virtual reality interface, 3D audio performance.

1. INTRODUCTION AND MOTIVATION

Contemporary artists have explored the power of electroacoustic spatial audio over the last few decades. Composers such as Edgar Varese or Karlheinz Stockhausen have used multi-speaker systems to immerse the audience in artificial sounds, extending upon earlier ideas of positioning the musicians freely around the audience. Today, an increasing number of artists are regularly using multi-channel speaker arrays to immerse the audience even further. Still, spatial audio is far from becoming mainstream.

While today's digital systems support matrix mixing to enable many sound sources and complex speaker arrangements, the user interface severely limits the performer and composers. There is little support to position sound sources in space beyond the traditional 2D joysticks on high-end mixing desks with surround capabilities; 3D capabilities are almost unheard of. Instead, a performer has to manually distribute sounds to speakers, typically using traditional channel-fader interfaces. Dynamic sound sources require complex automation, rendering live performance almost impossible. Compositions and recordings are limited to playback systems with closely matched speaker arrangements.

This situation arises because of the high dimensional requirement placed upon the interface: the intrinsic physicality of spatialized sound sources is best represented by four dimensions: three for position and one for source volume. If, for the sake of the discussion, we supposed the artist used eight speakers in a performance, they would be attempting to control an intrinsically four dimensional representation with eight channel faders.

Analyzing this interaction in more detail, the input device space has eight dimensions, with faders being operated for the most part sequentially; representation feedback is poor (the slider positions themselves) and related through a counter-intuitive mapping to the underlying physical representation, which is itself four dimensional. In effect we have created an 'interaction bottleneck': dimensionally impoverished control devices act upon an inefficient and counter-intuitive mapping, the only feedback and representation being the input device itself. Such an interface could not easily support multiple dynamic sound sources.

The project presented here tackles the bottleneck in two ways. Firstly, by separating data and control representation; the concept of a sound source in space is abstracted from the actual mixing process onto the output audio channels. Secondly, via a richer input device and simple mapping: the abstract sound sources are moved in space using a 3D input device; the mapping is now intuitive, directly relating the visualization to the sound spatialization, with source volume being represented by the orientation of the geometric sound object.

A first performance prototype was implemented to test the concept, provide a user interface and support collaborative performance. The underlying technology is derived from existing virtual reality technology and adapted to meet the requirements of music performance. Future work will aim to extend the capabilities of the initial prototype beyond limitations imposed by a VR system that is optimized for realism in simulations instead of artistic expression.

The remainder of the paper is organized as follows. Related work is acknowledged in Section 2. Section 3 then presents the system overview including all the components. Section 4 presents the interaction paradigm. Section 5 presents the extension of the interface for collaborative performance. Conclusions are drawn in Section 6 and future research directions are identified.

2. RELATED WORK

This work combines the fields of spatial music performance with virtual reality technology. It builds upon technology and interaction paradigms developed independently in the two fields.

There is a wealth of research literature concerning novel musical interface devices; [14] includes a comprehensive overview. Many artists have successfully employed VR-gloves and non-contact sensing in live performance; examples of early innovation being the work carried out at STEIM, and by Jaron Lanier and Tod Machover.

In the areas of mapping [3],[5] and visualization of sound, experimental research is relatively sparse, with mapping research also tending to focus on instrumental richness, rather than the intelligibility of the mapping.

NIME 06, June 4-8, 2006, Paris, France.

Historically, the use of VR as an expressive medium has concentrated on visual art, e.g. [8]. To date, there has been little experimental research on the use of VR interfaces for sound control [1],[10]. The DIVA system [6] was among the first to use VR for musical performance. A VR evaluation framework has been built at the Helsinki University of Technology, introducing concepts such as a virtual air guitar [9].

Spatial perception and rendering of sound is well understood, ranging from amplitude panning approaches [16], Ambisonics [4] to large speaker arrays for rendering wave fields or head-related transfer function methods predominantly used with headphones. Virtual reality toolkits usually include some form of spatial sound rendering, often based upon standalone audio servers or DSP hardware (e.g. Lake Huron). These VR systems employ sound as a tool to achieve a realistic simulation and increase the feeling of immersion. Little emphasis, however, has been given to tackling the difficult interaction issues surrounding the visualization and control of sound spaces for artistic purposes.

This paper addresses this area of interaction design, offering intuitive post-processing of the sound stream for spatialization.

3. SYSTEM OVERVIEW

The system consists of three major components: the scene-graph that stores the positions and parameters of the audio sources including their visualization; the user interface to modify the scene; and the audio rendering system (see Fig. 1). The audio rendering system as well as the “glue”-code required to combine the elements is provided by the blue-c API [12], a virtual reality toolkit originally designed for collaborative and tele-presence applications.

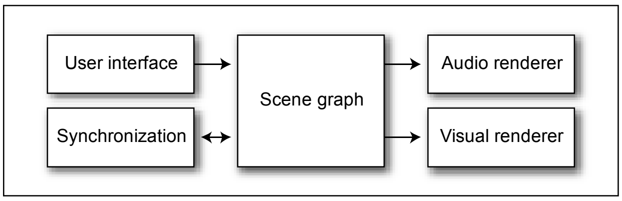


Fig. 1. System overview.

3.1 User Interface

The user interface module handles all user interaction with the scene graph and its embedded audio objects. The interface module enables the user to pick and move objects using tracked virtual reality data gloves. Rotating objects changes the volume of the point source; this user interface is described further in section 4. The interface also handles object locking to avoid concurrency problems when the system is used collaboratively.

3.2 Audio Rendering System

The audio rendering system spatializes the audio source objects using a volume panning approach, deriving the data from the scene graph. All audio sources are rendered using the blue-c API sound rendering system [13] that supports spatialization of a large number of sound sources with arbitrary speaker configurations. The audio system supports audio file playback either from memory (e.g. for short loops), streaming from disk (e.g. longer audio tracks) or from live sources (e.g. microphones or synthesizers). The audio rendering system was chosen mostly for convenience reasons as it performs well and is directly integrated into the virtual reality software development toolkit. Plugging in a different audio renderer or transmitting audio source positions to a different spatialization server (e.g. a Lake Huron system) would be straight forward.

3.3 Hardware Environment

The prototype implementation uses a standard virtual reality environment with a single wall-type stereoscopic projection surface, head- and hand-tracking, gloves with bend sensors for all fingers (see Fig. 2b) and a 14 speaker audio rendering system. Additional tests were conducted on a stereoscopic workbench environment with a slightly smaller 8 channel audio system (see Fig. 2a). Both systems are driven by a standard PC.

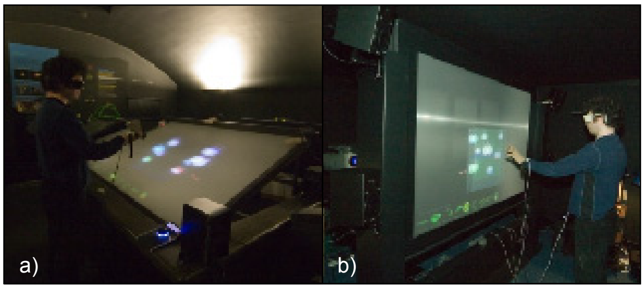


Fig. 2. Pictures of the hardware setup used for testing.
a) Workbench display system. b) Wall display system.

Although technically not required, picking in 3D is significantly easier with stereoscopic projection. Similarly, using a 6DOF tracking system allows for much smoother and more intuitive interaction than using a mouse that is inherently 2D.

The hardware environment was chosen based on availability at the lab. It is obviously not well-suited to live performances due to portability restrictions. The software, however, is flexible enough to run on a variety of platforms, including laptops.

3.4 Scene Graph: Representing 3D Audio

The scene graph is subdivided into a static and a dynamic section. The dynamic section (Audio group in Fig. 3) includes the sound sources and their visual representation. This dynamic section is synchronized and distributed among the different machines in the collaborative setting (see section 5). The static section (Stage and UI groups in Fig. 3) is used to provide the performer with guidance elements, such as a reference coordinate system, the position of the speakers, or an abstract representation of the performance environment and user interface elements.

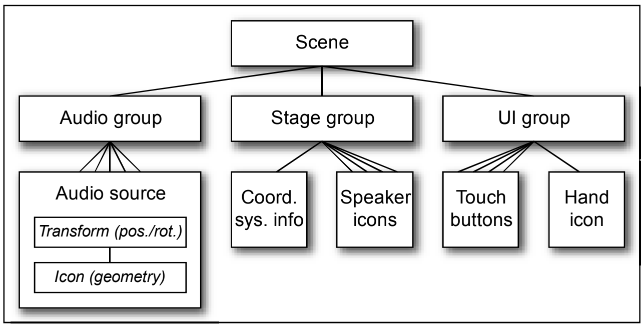


Fig. 3. Scene graph structure.

Each sound source is represented as a simple geometric object. The associated structure in the scene graph consists of the position transformation node at the top, an additional rotation transformation node and attached geometry. The position given by the transformation node is transferred to the audio renderer to control spatialization of the respective audio object. The rotation transformation affects only the small pointer visually. The roll component is transformed into a gain value. The current prototype only implements gain control; the two additional rotation axes, however, could be used to control source-dependent effects (e.g. reverb send, filter frequency, etc.). Care has to be taken not to overload the user interface, though. The use of a purely geometry-based approach exploits the scene synchronization features of the underlying VR toolkit without requiring additional modification.

4. THE PERFORMANCE INTERFACE

The performance interface concept is derived from a 3D visualization and interaction environment "AutoEval" [1] originally developed for design review in the automotive industry and adapted to the needs of 3D audio manipulation.

4.1 Glove Interface

All editing is conducted using a tracked virtual reality glove. For the prototype implementation, we used a Polhemus Fastrack 6DOF magnetic tracking system and an Immersion CyberTouch glove with vibration devices on the fingers and palm to provide haptic feedback.

Editing operations are initiated by picking an audio object with the tracked glove. Picking virtual objects in 3D is often difficult for the untrained user, especially for those with limited stereoscopic depth perception. The interface system therefore supports the user with additional cues: if the user touches an object, the object is visually highlighted and small vibration motors inside the VR glove (similar to those in mobile phones) provide a haptic sensation. A touched object can be picked by pinching the thumb and index finger.

The audio object is moved by picking and dragging the object to the desired location. The attached audio source is continuously updated during move operations, enabling the performer to "animate" sound in real-time.

Volume is changed by rotating the hand while the object is grabbed, which is essentially the same gesture as turning a physical knob. While the move operation follows a 1:1 mapping, twisting the knob is accelerated by a factor of two. Tests revealed that a scaled mapping reduces fatigue without significantly sacrificing precision. Additional editing modes are available to avoid accidental side effects.

4.2 Latency Issues

In this system, latency is introduced at various stages. The audio rendering engine itself can run with the smallest possible buffer size allowed by the audio interface. Significant latency, however, is introduced through the user interface handling that runs synchronized to the visual rendering system, with frame times typically between 16 to 22 ms. Magnetic tracking systems also introduce a delay due to their limited update rate and required noise filtering. All factors included, the time delay between an actual event and its effect on the audio output may well go beyond 50ms. It is therefore clear that this interface is not suited to control a percussion performance. Preliminary tests, however, suggested that the system is fast enough for all practical uses.

4.3 Working Volume

Previous research suggests that the optimal actual working volume of the hands is relatively small compared to the volume defined by the fullest reach and is maintained at a fixed distance relative to the body. The standing position was chosen to enhance performance aesthetics; in this position, fine motor control is best achieved with a hand position 50-100mm above elbow height and within the 'normal working area' [15] which equates to approximately one forearm's span from the body. Hence the working volume chosen is a cube of approximately 0.4m centered directly in front of abdomen, and the audio scene is scaled accordingly. In this region the most accurate motor control is achieved and additionally, muscle fatigue is greatly reduced. In contrast, typical wall-type VR display environments are designed for a large interaction volume, and consequently offer less precision for a small interaction volume as used for this prototype. The workbench type display fits the preference for a smaller working volume better and proved to be less tiring to work with.

5. COLLABORATIVE PERFORMANCE

The system supports collaborative performance in two ways. By using the live audio streaming sources, a performance can be split along functional lines: one performer is responsible for the spatialization of the sounds from the other musicians, much in the same way that a front-of-the-house mixing engineer takes care of the band's sound (Functional separation in Fig. 4). This option is not further discussed here.

The system also supports concurrent editing of the 3D sound stage by several users (Parallelization in Fig. 4). For this type of operation, the virtual sound stage is distributed and synchronized among several computers, one per spatial audio performer. 3D audio objects can be edited independently on all connected computers, while a locking system ensures that no two users try to modify the same object at the same time.

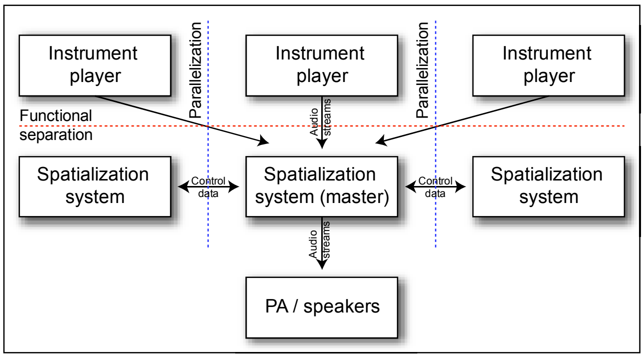


Fig. 4. Collaborative performance. Functional (sequential) parallelization between instrument players and spatialization vs. parallelization within the spatialization post-process.

5.1 Synchronization between Machines

The system enables several users to edit the sound stage concurrently. All performers use their own computer, each displaying the stage from an arbitrary viewpoint and polling the input device. Each machine holds an independent copy of the stage group and user interface, and the software ensures that all machines share the same position and state for all sound sources. This essentially provides multiple instances of the user interface to a single audio rendering system.

The synchronization is based upon the blue-c Distributed Scene Graph (bcDSG) [11] that synchronizes the scene graph data structure across multiple machines and manages concurrency issues including locking to make sure no two users can modify the same object concurrently. Although the bcDSG was designed with graphical applications in mind, the synchronization system did not require any adaptation since all audio states have a direct scene graph representation. Building a distributed application therefore required only little additional development effort over a single-user solution.

5.2 Distributed Audio Rendering

In a typical live performance situation, only a single computer actually processes audio data; the other machines are used for visualization and interaction only. If desired, every machine in the distribution group could run their own local audio renderer with an arbitrary speaker setup, providing individual monitoring for each performer.

6. CONCLUSIONS AND FUTURE WORK

The work presented here forms the first step towards an intuitive performance system for spatial audio and music performance. We demonstrated the usefulness of virtual reality tools in the context of music performance, and introduced a first concept prototype for visualizing sound sources in a 3D environment.

The interaction paradigm has proven its effectiveness over previous multiple-fader techniques; due to the improved efficiency of the interface, one performer is now able to sequentially alter the spatialization of multiple sources, whereas previously this sequential or 'time-sharing' capability had been consumed by poor interaction. In addition, the collaborative interface allows parallel spatialization of sound sources with multiple performers. Thanks to scene distribution features inherent in the underlying VR toolkit, enabling a collaborative performance only required minimal additional development effort.

Additional work will be required to increase the dimensionality of the control interface. The current system only supports position and gain parameters, whereas a fully fledged performance system should include effect control. Quantitative and qualitative HCI testing will be used to determine which interaction paradigm(s) represent the most expressive musical interface.

7. ACKNOWLEDGEMENTS

Thanks to Alistair MacDonald for fruitful discussions about 3D audio performance, to ETH Zurich for providing the blue-c API, and the developers of the original AutoEval system at the Digital Design Studio for inspiration of the user interface paradigm.

8. REFERENCES

[1] Anderson, P., Kenny, T., Ibrahim, S. "The role of emerging visualisation technologies in delivering competitive market advantage." *2nd International Conference on Total Vehicle Technology*, Institute of Mechanical Engineers, University of Sussex, Brighton, UK, pp. 87-97, 2002.

[2] Choi, I. "A Manifold Interface for Kinesthetic Notation in High-Dimensional Systems." in *Trends in Gestural Control of Music.* Battier and Wanderley, eds., IRCAM, Paris. 2000.

[3] Garnett, G., Goudeseune, C. "Performance Factors in Control of High-Dimensional Spaces." In *Proc. 1999 Int'l Computer Music Conf.* San Francisco. 1999.

[4] Gerzon, M. "Ambisonics in multichannel broadcasting and video." *Journal of the Audio Engineering Society 33, 11*, 859–871. 1985.

[5] Hunt, A., Wanderley, M., Paradis, M. "The Importance of Parameter Mapping in Electronic Instrument Design." In *Nime 2002 Proceedings*. 2002.

[6] Huopaniemi, J., Savioja, L., Takala, T. "DIVA virtual audio reality system." In *Int. Conf. Auditory Display*, Palo Alto, CA, 111-116, 1996

[7] Jot, J.-M. "Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces." *Multimedia Systems 7, 1*, 55–69. 1999.

[8] Keefe, D., Acevedo, D., Moscovich, T., Laidlaw, D.H., LaViola, J. "CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience." In *Proceedings of ACM Symposium on Interactive 3D Graphics 2001*, pages 85-93, March 2001.

[9] Mäki-Pataola, T. "Experiments with Virtual Reality Instruments." *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression [NIME05]*. Vancouver, BC, Canada, 2005.

[10] Mulder, A. "Design of Virtual Three-Dimensional Instruments for Sound Control." PhD Thesis, Simon Fraser University. 1998.

[11] Naef, M., Lamboray, E., Staadt, O., Gross, M. "The blue-c distributed scene graph." In *Proceedings of the IPT/EGVE Workshop 2003.* J. Deisinger and A. Kunz, Eds. 2003.

[12] Naef, M., Staadt, O., Gross, M. "blue-c API: A Multimedia and 3D Video Enhanced Toolkit for Collaborative VR and Telepresence." *Proceedings of ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry 2004*. VRCAI 2004, Singapore, June 16-18, 2004.

[13] Naef, M., Staadt, O., Gross, M. "Spatialized audio rendering for immersive virtual environments." In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2002*. H. Sun and Q. Peng, Eds. ACM Press, 65–72. 2002.

[14] Paradiso, J. "Electronic Music Interfaces: New Ways to Play." *IEEE Spectrum*, *34(12)*, 18-30, 1997.

[15] Pheasant, S. "Bodyspace: Anthropometry, Ergonomics and the Design of the Work." pp52-57, CRC; 2nd edition. 1996.

[16] Pulkki, V. "Uniform spreading of amplitude panned virtual sources." In *1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* 1999.

Using the Touch Screen as a Controller for Portable Computer Music Instruments

Günter Geiger
Music Technology Group
Universitat Pompeu Fabra
Ocata, 1
Barcelona, Spain
ggeiger@iua.upf.es

ABSTRACT

Using mobile devices as instruments in computer music is one of the goals of the "Pure Data anywhere" project [5]. An obstacle we encounter is controllability, because most of the devices do not offer the necessary interface, such as MIDI or USB, in order to be controlled by external controllers. Also, attaching external controllers to the devices would make them less portable.

This paper investigates the possibilities of using the touch screen, an interface that is part of mobile devices like Personal Digital Assistants (PDA's). It describes usage scenarios that have been implemented for the PD anywhere system. As most traditional PDA applications use the touch screen in the same way as a mouse would be used, emphasis is put on the difference between mouse and touch screen interaction for instruments.

We are going to describe interaction models, that were found useful and intuitive and enable the touch screen to become a fairly sophisticated controller for expressive real time music on a PDA.

Keywords

touch screen, PDA, Pure Data, controller, mobile musical instrument, human computer interaction

1. INTRODUCTION

Most mobile devices offer a touch screen which is used together with a pen in order to take the role of the mouse for application control. Apart from some additional buttons, the touch screen is the main interface to the system. Therefore it seems to be natural to use it as a controller for musical instruments that are running on the device. However, just using the touch screen as it is used in standard applications is too limiting. The WIMP (windows, icon,menu,pointer) based paradigm has proved to be difficult to handle for live musical performance, especially if the instrument should be highly interactive and feel natural. It is hard to express musical ideas by moving sliders and pressing buttons, more so if they are very small, as is common on PDA's.

Therefore we have tried to surpass these limitations and go beyond the traditional control paradigms, by using dynamic gestures on the touch screen to control software instruments running on the device itself.

Our goal in defining new interaction principles is to construct an interface that resembles traditional instruments, and especially portable instruments, in several ways.

First, it is one piece, the mobile device, and not a collection of controllers and synthesis engines.

It should stay a portable instrument, which amplifies its applicability to different contexts. Portable instruments play an important role in music culture, they can be moved easily, can be used in different social events and situations and deliver music outside the concert hall or studio. Having too many cables and addons would make the setup of the instrument more compilicated and its usage more error prone.

The instrument should have an interface that maximizes control and gives immediate feedback. It should, actually, feel like an acoustic instrument, but produce sound that acoustic instruments can't produce. It should also offer control on other than note level, such as score level or sound processing level [13].

It should be a learn-able and master-able instrument [7], and the player should be able to have direct feedback on how he advances in mastering the instrument [9].

Under these premises we try to develop a model of interaction with the touch screen.

1.1 Pure Data anywhere

Pure Data anywhere (PDa) is a port of the Pure Data (PD) computer music system [10] to Personal Digital Assistants. PDa is based on the Linux port of Pure Data and runs on any PDA that supports Linux. The graphical user interface is based on tcl/tk. PDa implements sound calculations with fixed point number values, therefore it runs in realtime on processors without floating point unit, such as those used in portable devices.

More information about the implementation can be found in [4].

PDa is currently used and tested on several models of HP iPaq's (namely those who run the familiar linux distribution), on PDA's that natively support Linux such as the Sharp Zaurus, and to a lesser extent (without the graphical frontend) on several of Apple's iPod models using the ipodlinux system. For the iPod there is even a costum graphical frontend called pdpod [8] available.

2. CHARACTERISTICS OF THE TOUCH SCREEN ON MOBILE DEVICES

In order to understand what the touch screen could offer, we are trying to highlight the features of touch screens.

The touch screen on a PDA feels very direct in its interaction, because of the connection of visual and haptic cues as well as the connection of the left hand holding the device and the right hand controlling it. This gives relatively good control over where the device is, and where the stylus is in relation to the device. This is very different from traditional touch screens, because these devices are not mounted on a screen anymore but freely movable, adopting more behaviors of touch tablets.[1]

This is also the reason why moving on the PDA touch screen is fast. Moving from one point to another involves the movement of both hands and arms. The haptic feedback and reference points about the screen dimensions as well as its limited size make it easy to navigate even without seeing the screen.

The touch screen can be operated either with a stylus (which gives more precision, but takes away immediacy) or directly by touching with the fingers. If we use two different fingers with the touch screen we can jump from one area of the screen to another area almost immediately and with relatively high precision. This mode of usage is seldomly used in traditional applications, but it makes perfect sense for musical applications when control has to be discontinuous and fast, or where one wants to achieve special effects.

We already mentioned the portability and the small size of touch screen equipped mobile devices. An important factor for a successful controller [14].

On the touch screen it is easy to move around, especially by small amounts, making the touch screen relatively a higher resolution device than for example the mouse, where it is hard to move just by one pixel. It is easy to remember positions haptically, because one can feel the borders of the screen.

The high precision that can be achieved when using a touch screen is also due to the short ways the stylus or fingers have to travel in order to change parameters, the whole range of the device is about 8cm, whereas the precision is between 320 and 640 units.

Additional haptic cues can be put onto the device, helping the instrumentalist to orient himself without having to look at the screen of the device.

3. THE TOUCH SCREEN AS SOPHISTICATED CONTROLLER

In this section we will work on ways of interaction with the touch screen using the possibilities outlined before.

Wessel [14] defines the principal interaction with digitizing tablets as “scrubbing”, “drag and drop”, “catch and throw”, and “dipping”. Our definitions partly overlap with these, they are not meant as complementing, but as a different point of view on interaction with a device very similar to a digitizing tablet.

We can identify several different forms of interacting with the touch screen, which partly overlap and merge into each other. The fact of being able to use several of the principles at the same time increases the possibilities of interaction and makes the controller more sophisticated. We start with a list of four interaction principles.

3.1 Region based triggering

Region based triggering will fire up distinct events based on where the touch screen was hit. In the WIMP paradigm this corresponds to a button press.

For musical purposes we can enhance the button press. The first enhancement is the power with which the button gets pressed. While interacting in a natural way with the touch screen we found out, that the hit does not take place on one single spot of the screen, but it takes the from of a small line. The speed with which this line gets drawn is an estimate for the power of the hit, and hence can be used as a control parameter for dynamics. Also the length of the line can be used for controlling purposes.

Simultaneity can be approximated by drawing from one region to another region, or by interaction with two fingers at almost the same time.

3.2 Gesture recognition

On a higher and more general level, there is gesture recognition. Gesture recognition allows us to control events from an alphabet of gestures. Not only the outline of a gesture can be used, but also its timing information. This adds another level of freedom to each gesture.

Gestures can be pretty complex, and if we include information about absolute position gesture are a very general way to define interaction on the touch screen.

The meaning of gestures in new musical instruments has been described by other authors [11] [12].

3.3 Border crossing

Another, newly defined interaction scheme we call border crossing. An event gets triggered when the pointer on the touch screen crosses a border. Additional parameters depend on speed of the crossover. This pattern draws from the interaction of plucking a string. Together with the sound feedback and the haptic feedback of the screen border, it is easy to learn the positions of borders by experimenting, and the human cognitive system seems to remember these positions pretty well.

Because of the precision and speed of the stylus on a touch screen, additional trigger information such as “double-triggered” (the border gets crossed two times in a very short time) events can be used to determine parameters on the note level.

Information such as the angle with which the border gets crossed can be employed as a continuous parameter.

3.4 Continuous parameter control

As an extension to the region based triggering, or as a stand-alone technique, the touch screen offers continuous parameter control. This means, that control data gets send according to the position of the pen on the touch screen. This allows us to combine piano-like event triggering with continuous control of the sound. This way the evolution of events can be controlled, depending on the mapping of the x and y values.

Continuous parameter control works best when the control input switches between states. One of the event based paradigms, region based triggering, gesture recognition or border cross triggering can be used to produce events, and for further control continuous parameters are used, until the pen (or finger) gets lifted from the touch screen.

3.5 On Virtuosity

One of the reasons why virtuosity can be reached with traditional instruments is their inflexibility. If one starts to

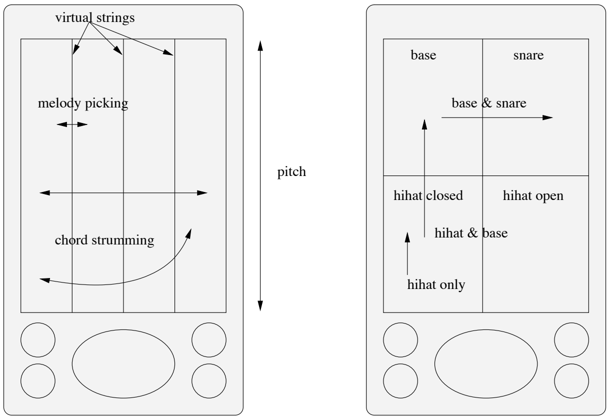


Figure 1: The guitar and drum set screen layout

learn a traditional instrument, she can be pretty sure that the instrument will not change substantially, hence it pays off to put an effort into learning to play it. Paradoxically, the flexibility of most computer music controllers makes it hard to reach higher levels of virtuosity on an instrument.

Unlike work specific to single compositions [2], one of the goals of this article is to come up with a set of interaction principles with the touch screen, that can be regarded as fundamental, studied and practiced as such in order to lead to a higher level of control of the instrument (or, more general, the interface).

We might fail for the first run, but experience and time will show which of the outlined principles preserve and which have to be adapted.

This, of course, assumes that touch screens will stay the main way of interaction with mobile devices.

4. MAPPING

Once refined our ways of interaction, we can work on the mapping of these control inputs to our instrument. At this time we have left the domain of the "controller" and entered the layer of the computer music instrument, generally hidden in the software of the device. Having good mapping strategies allows us to combine the principles of interaction in order to build a pleasing and enjoyable instrument.

The examples here should be seen as a proof of concept and an application of the interaction principles. Our mappings try to be simple in the hope that it will be easy figure out how the instruments react without needing a detailed description of synthesis techniques.

4.1 The virtual guitar

Figure 1 shows an application of the "border crossing" principle, implementing a virtual guitar. Interface resembles the FMOL program [6], but the interaction and sound production is different. The interface behaves similar to a real guitar, it is used to trigger events while crossing (picking or strumming) the virtual strings. Together with the event we can extract two parameters, which are the pitch and the velocity. The pitch is controlled by the vertical position while crossing the string. Simultaneity is achieved by strumming over more than one string, where the curve of the strum influences the type of chord that can be played (e.g. major/minor).

We have chosen on purpose a traditional instrument to simulate, but on a more abstract level of the interface, we can extract a decent amount of parameters from one strumming action. In the 3 string guitar example this would be 3*2, although not completely independent. Our abstract instrument would need something that makes tuned chords (= tuned parameters) useful. Also the velocities of the single strings are tightly coupled. Vibrato effects can be implemented by interpreting movements after "picking", using continous parameter control.

The virtual guitar is an example where additional haptic feedback on the touch screen improves play-ability further. This additional haptic feedback can be achieved by glueing transparent film on the surface, where the borders of the film correspond to the guitar strings and will be felt when strumming over them.

4.2 The virtual drum set

The second screen configuration in Figure 1 shows an example setup of a virtual drum kit. In this setup, events are not triggered by crossing borders, but by hitting certain areas of the screen.

The gesture arrows show how a simple two voice drum pattern would be played with this setup. The smallest rhythmical entity, the closed hi-hat is hit constantly, followed by a "hi hat - bass drum" and "bass drum - snare" combination. More complex drum sets could be built according to the rules for region based triggering outlined above.

According to need, other region layouts can be devised, such as the hi hat in the center, surrounded by other fields. Events that occur together frequently should have neighboring regions.

The virtual drum set mode can also be enhanced by haptically marking the regions.

4.3 The free and unquantified mode

The theremin, being the first controller in the league of new instruments, is one of the few instrument controllers where people have reached the level of virtuosity. It is therefore instructional to take a closer look at it and compare it with the touch screen.

The control of the theremin are two decoupled one dimensional sensors. One of the virtuosity criteria is the ability to hit a specific note and stay in tune. This is done via vibrato and by remembering relative positions which correspond to intervals.

When using a touch screen as a 2 dimensional controller, just using x for volume and y for pitch, one has to fight with similar problems as a theremin player, like finding the right pitch.

Another problem of the touchscreen in this mode is the freedom of movement. The finger can only move in one plane, volume and frequency are tightly coupled, making it harder to hit the notes correctly and perform a decent frequency vibrato.

On the other hand, on the touchscreen frequency and volume vibrato can be combined easily.

The frequency solution is not sufficient, therefore the frequency should be coupled with detection of the speed with which the stylus is moved.

5. THE IMPORTANCE OF FEEDBACK

In digital instrument design, feedback is a very important factor if we want the instrument to be playable. The touch screen, not being designed as an instrument, lacks kinesthetic feedback.

Nevertheless the tactile feedback gives good information about the position of the pen or finger on the touch screen.

The described scenarios have a very tied coupling of event and sound feedback, this is on purpose because the sound is the only dynamic feedback given to the user.

Another well integrated form of feedback is the visual feedback. Unlike with the mouse or other controllers, visual feedback can be displayed where the interaction occurs. This might be helpful for playing the instrument, but the history of instrument playing shows us that only few instruments actually depend on visual feedback. We encountered that having to look at the screen actually takes away a lot of the fun when playing the instrument.

The most important factor that has to be mentioned when playing mobile devices is the feel of the instrument. Having a portable device and being able to control it precisely motivates to go on with it and to learn how to have the total control.

6. FUTURE ENHANCEMENTS

The general way of playing a hand held device is by holding it in one hand and playing the touch screen with the other hand. This interaction works well for the touch screen, unfortunately for the hand that is holding the device it is hard to reach the additional buttons on the device. The only button that is normally reachable is the record button.

Having the ability to input additional information with the hand holding the device would greatly enhance the flexibility of the instrument. A solution for this would be the design of a jacket around the hand held, which serves at the same time as a protection and a input device with several buttons (one for each finger)., similar to the DataEgg[3], an input device developed by NASA for astronauts.

These additional, but more static inputs would make it easier to change states or presets of the instrument or generally to be used as an additional controller.

7. CONCLUSIONS

We have shown that the touch screen of mobile devices can be a versatile and musically meaningful instrument controller. Describing patterns of gesture interaction and explaining instruments implemented with these patterns we have demonstrated their usefulness.

We found playing with an instrument implemented in a portable computer stimulating and interesting, as well in private settings with headphones as in concert setting.

This opens up the road for new applications that convert a PDA into a portable musical instrument and make this kind of computer based music instruments easily accessible and usable, not only in complex concert or studio setups, but also in any other setting, by just pluggin in the headphones and playing along.

8. REFERENCES

[1] Bill Buxton, *Human Input to Computer Systems: Theories, Techniques and Technology*, http://www.billbuxton.com/inputManuscript.html, 2006.

[2] Carlos Cerena, *Gesture control of musical processes - a MAX environment for the Lightning*, Organised Sound **5** (2000), no. 1, 3–7.

[3] Gary Friedman, *Introducing the dataegg*, http://www.e2solutions.com/dataegg/index.html, 1992.

[4] Günter Geiger, *Pda - real time signal processing and sound generation on handheld devices*, Proceedings of the 2003 International Computer Music Conference (San Francisco), International Computer Music Association, 2003.

[5] Günter Geiger, *Pure data anywhere*, http://gige.xdv.org/pda, 2005.

[6] Sergi Jorda, *Faust music on line: An approach to real-time collective composition on the internet.*, Leonardo Music Journal **9** (1999), 5–12.

[7] Sergi Jordà, *Digital lutherie: Crafting musical computers for new musics performance and improvisation*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, 2005.

[8] Martin Kaltenbrunner, *Pdpod*, http://ipodlinux.org/Pdpod, 2005.

[9] Sageev Oore, *Learning advanced skills on new instruments.*, New Interfaces for Musical Expression (Vancouver), University of British Columbia, 2005.

[10] Miller Puckette, *Pure data: another integrated computer music environment.*, Second Intercollege Computer Music Concerts (Tachikawa, Japan), 1996, pp. 37–41.

[11] M. Wanderley and M. Battier, *Trends in gestural control of music.*, http://www.cdemusic.org, 2000.

[12] Marcelo M. Wanderley, *Gestural control of music*, International Workshop Human Supervision and Control in Engineering and Music, September 2001.

[13] Marcelo M. Wanderley and Nicola Orio, *Evaluation of input devices for musical expression: Borrowing tools from hci*, Computer Music Journal **26** (2002), no. 3, 62–76.

[14] David Wessel, Matthew Wright, and John Schott, *Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors*, Conference on New Instruments for Musical Expression, 2002.

Using MIDI to Modify Video Game Content

Jukka Holm
Nokia Research Center
Visiokatu 1
33720 Tampere, Finland
jukka.a.holm@nokia.com

Juha Arrasvuori
Nokia Research Center
Visiokatu 1
33720 Tampere, Finland
juha.arrasvuori@nokia.com

Kai Havukainen
Nokia Technology Platforms
Visiokatu 1
33720 Tampere, Finland
kai.havukainen@nokia.com

ABSTRACT
This paper discusses the concept of using background music to control video game parameters and thus actions on the screen. Each song selected by the player makes the game look different and behave variedly. The concept is explored by modifying an existing video game and playtesting it with different kinds of MIDI music. Several examples of mapping MIDI parameters to game events are presented. As mobile phones' MIDI players do not usually have a dedicated callback API, a real-time MIDI analysis software for Symbian OS was implemented. Future developments including real-time group performance as a way to control game content are also considered.

Keywords
Games, MIDI, music, rhythm games, background music reactive games, musically controlled games, MIDI-controlled games, Virtual Sequencer.

1. INTRODUCTION
Music has an important role in contemporary video games. It can help to make a right kind of atmosphere for gaming, and emphasize actions on the screen. It is common that the background music is adaptive i.e. it changes according to game events, between different parts of the game, and is synchronized to the game actions. As an example, when an avatar is moving in a safe area the music may be slow and relaxing, but during an enemy attack it becomes faster and more aggressive.

The development of an adaptive music soundtrack and sound effects for a modern video game is an expensive and time-consuming process. Due to this, many games just loop the same relatively short music files over and over. Some developers have started using songs from popular artists as background music in their games. Repetition has also its cost: The gamer may become bored with the non-adaptive soundtrack and turn it off after a while. The study by Cassidy et al. [19] suggests that the best player experience emerges when a player can choose a game's background music to something that he or she prefers. The scope of that study was limited to driving games.

Since 1990's, we have also seen the rise of musically oriented games. As Blaine points out in [14], the majority of these are so called "rhythm games" that prompt a single player or a group of players to perform rhythmic actions in time with a predetermined musical sequence. The game genre has also led to the development of new low-cost interfaces such as drum, guitar, and dancemat controllers that make the games more enjoyable to play. Many rhythm games suffer from the same problem as non-adaptive game soundtracks: As the number of music files is limited, the games may have quite short lifecycles. To improve the situation (and earn more money), some developers have started offering game upgrades such as catalogues of popular songs.

This paper describes a novel way to use music in games. Instead of adaptive background music that reacts to the game events, the authors propose the concept of games that react to their background music. Throughout this paper, this kind of games are referred to as "background music reactive games." In [17] and [18], the authors have also used the term "musically controlled games." The idea works both in the case of rhythm games and non-adaptive background music soundtracks.

The contents of this paper are as follows: Chapter 2 discusses some relevant previous work, Chapter 3 presents an overview of the idea, and Chapter 4 justifies using MIDI to control game content. Chapter 5 lists some interesting MIDI parameters and ways to use them to modify games. Chapters 6 and 7 discuss authors' test platform "AudioAsteroids" and MIDI analysis software for Symbian mobile phones. Finally, Chapter 8 draws some conclusions and Chapter 9 suggests some topics for future work.

2. PREVIOUS WORK
When thinking about "music games", most people are probably referring to "rhythm games." In PlayStation game "Parappa the Rapper" (NanaOn-Sha 1997) [9] and other similar rhythm games, the player has to trigger musical events by pressing specific buttons to the beat of the music. In "Gitaroo Man" [3] for PlayStation 2, the player must also follow the pitch of lead instrument with the joystick. In "Mad Maestro" [7] for PlayStation 2, the "appropriate" tempo of each song determines the speed with which the player must press the buttons. In all of these games, the sets of songs are fixed and the programmers have had to define the occurrence of each sequence in every song. In other words, the generation of game levels from pieces of music is not automatic.

In [14], Blaine discusses alternative game controllers that have been used with music games. These controllers border double-function as musical controllers. Experimental hybrids between games and new types of multi-user musical controllers, including Jam-O-Drum, Jam-O-Whirl, and Jam-O-World, have been discussed in [16].

One example of a video game in which music becomes the result of players' actions, that in themselves are unrelated to the theme of making music, is Shockwave game "BLiX" [10]. "Rez" (Sonicteam 2001) [11] for PlayStation 2 is a third-person

NIME 06, June 4-8, 2006, Paris, France.

shoot ’em up game with drum samples instead of conventional weapon sounds. When the player shoots, the triggering of samples is quantized so that they always match the background music rhythmically. It could be said that in games such as BLiX and Rez, player’s actions on top of the background music produce a certain soundtrack along each play session. However, these games cannot be played with background music of player’s own choice.

A couple of games that respond to the pitch of player’s whistling, singing, or humming have been implemented. In many cases, the pitch is rounded to the nearest semitone. Hämäläinen et.al. [4] discuss games controlled by singing. One of their examples is a pitch-controlled Pong. The authors also discuss the technical and psychoacoustic issues of pitch detection. In karaoke game “Staraoke” (Intervisio 2003) for Windows 98/XP, the pitch of background song’s melody forms a path. The player must guide his or her character through the path by singing the melody correctly. Pitch is represented on the vertical axis and time on the horizontal axis. [2]

Of all music games on the market, the one that most closely resembles ideas presented in this paper is Playstation game “Vib-Ribbon” (NanaOn-Sha 1999) [8]. At the time of its release, Vib-Ribbon was welcomed as a refreshingly new kind of game. It is best described as an obstacle track game, in which the background music (any song from any audio CD) affects the appearance of obstacles, the points in time when these obstacles appear, and the spawning of certain additional objects. Player’s character seems to walk along the stylized waveform. One issue with Vib-Ribbon is that the correspondences between characteristics of music and obstacle track are not that obvious for casual players. The obstacle track just appears different with different pieces of music.

In addition to audio-control, a small number of games that use MIDI controllers for input have been developed. David Bagno’s “Musical Space Invaders” and "Music Scale Teacher" [6] for Windows 98/XP and Macintosh are note teaching games played with a MIDI keyboard. “Musical Invaders” [5] for Windows 98/XP is a music learning game that responds to real-time MIDI input. The goal is to play the notes as they appear on the screen. By doing this, the player performs a melody at the same time. Players can also load their own MIDI files and play them as game levels.

3. OVERVIEW

Starting a “background music reactive game” differs somewhat from traditional games. In the beginning, the player must first select a music file or collection of files to be used in the game. The music is analyzed for relevant musical parameters (see Chapter 5) either in real time, in larger buffers, or the whole song can be processed before the game starts. The resulting control data is then sent to the actual game engine, which maps it to selected game parameters. After this, the player starts the game and tries to play through as many files as possible. Depending on the game type and implementation, each file may be considered as one unique game level.

When music is analyzed to produce control parameters for the game, novel ideas can be found. Even a very trivial game can be made interesting if the player can affect the difficulty level by changing the background music to his or her favorite tune. A player may end up saying things like: “I passed the game with Queen’s Show Must Go On, but Steep’s Rise is far too difficult for me!” In the case of mobile phones, even ringing tones could be used to control game parameters.

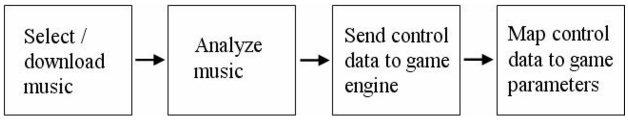


Figure 1. Starting a background music reactive game.

The concept also offers many imaginative possibilities for the game designers. As an example, consider a submarine combat game. It is possible to use music to control the behavior of the enemy so that it can shoot only when a certain note in the background song is played. Also elements not essential to the gameplay can be controlled by the music, for instance, the underwater plants may sway in the rhythm of the background song. In conventional game implementations, these actions would be controlled by a random generator or artificial intelligence (AI). Through music-control, the actions do not appear random.

Examples of musically controllable game elements and characteristics include e.g.:

- Speed and difficulty level of the game;
- Location of game objects (enemies, ammo, guns, bonus objects, buildings, etc.);
- Number, size, type, color, and shape of objects;
- Time and frequency of appearance of new objects;
- Movement (e.g. speed, direction, rhythm, starting point, trajectory) of objects;
- Properties of avatars (e.g. skills and endurance);
- Context of gameworld (e.g. location of game events, time of day); and
- Camera angle.

4. BENEFITS OF MIDI

With the current technology, MIDI has certain advantages over other file formats such as Wave and MP3. As MIDI is a symbolic format, it is more precise to use to control game content than digital audio. Certain MIDI parameters such as Note On messages can be directly used as game control parameters, while more control parameters can be found by making some simple calculations based on MIDI events for instance on a specific channel.

The file size of MIDI is considerably smaller than that of digital audio. Because of this, a larger amount of files can be stored to e.g. a portable device and new files can be rapidly downloaded over the air. Due to the recent increase in storage space and computational power, there are no longer strict limits to the size of wavetable soundbanks and sophisticated synthesis methods such as physical modeling can also be used to generate the sounds.

MIDI has also some important benefits if compared to other symbolic audio formats such as Open Sound Control [1]. The format is widely spread so there are lots of software and hardware tools available. People can easily download new songs from the Internet, and in the case of mobile phones even ringing tones can be used to create new game levels.

5. MAPPING MIDI EVENTS TO GAME PARAMETERS

Mappings between musical control parameters and game parameters are most effective when players immediately understand the relationship between what they hear and what they see. If a player knows a certain piece of music well and understands the mapping used in the game, he can anticipate some of the actions that will occur in the game.

In order to support a large number of players having different musical tastes, the mapping should not be tailored to a single genre. There are undoubtedly interesting differences between musical styles, but in most cases the game designer should aim at a generic mapping that works nicely in the case of any genre. It is beneficial if distinct musical genres produce distinct game experiences, but this should not be done at the cost of the general playing experience.

While most MIDI parameters affect how the music sounds like, some may not be noticed by inexperienced listeners. Parameters that are understood by most players should usually be connected to major foreground events in the game, while the not as obvious ones can be used to control less important things like background graphics and so on.

For the purposes of background music reactive games, MIDI parameters and control data calculated based on them can be divided into three principal groups: “Event”, “state”, and “transition”. In the following, we describe these groups in more detail (but not exhaustively) and make suggestions on how useful they would be to control a game.

5.1 Event Parameters

“Event” parameters are musical features that occur occasionally and last for a brief time. They are appropriate for e.g. triggering new objects or causing some abrupt actions in the game. Most useful musical features belong to this parameter group.

According to our experiences, the most useful MIDI message type belonging to this category is Note On. All Note On events in the song or only certain pitches could be mapped to e.g. spawn new game objects to the screen. Another interesting possibility would be to map higher notes to spawn objects to the top of the screen, and lower notes to the bottom of the screen. Note On velocities, which typically have values above 50, can also be used to modify the game.

An average listener seems to notice quite well when a certain percussive sound (such as bass drum or crash cymbal) has an effect on the game. In the case of polyphonic music, mappings between harmonic notes (especially those on accompanying tracks) and game events seem to be more difficult to notice.

In general, MIDI channels that have a low number are more widely used than channels having a high number. Channel 10 is an exception, as almost all MIDI files include at least one drum or percussion track. If only a subset of MIDI channels is mapped to the game parameters, drums are a good alternative as an average listener can easily separate them from the mix.

Another interesting musical feature, although not directly a MIDI event, is polyphony. As high-polyphony music often sounds more massive and intensive than low-polyphony, exceeding a certain polyphony threshold could be mapped to e.g. game’s difficulty level, amount of enemies, and so on.

By mapping Program Change messages to game parameters, it is possible to modify the game according to instruments used in the song. Traditional instruments such as piano are really popular, while e.g. different sound effects sounds are quite rarely used. Almost all songs include at least one drum or percussion track, and the basic drum set (bass drum, snare drum, hihat, etc.) is usually used.

Pitch Bend MIDI event is typically used in the case of stringed instruments and lead sounds, and its effect on the music is quite clear. The event can be used e.g. to control the vertical position of objects on the screen.

System Exclusive and NRPN messages are quite laborious to create. If they occur in a song, the composer has most probably used them to modify some parameter of some specific synthesizer. If another synthesizer is used to play the song, it is very likely that the listener will not hear the effect at all. These messages are really rare, and therefore can be used to create random –like surprise elements to the game.

There are 127 different MIDI Control Change messages, most of which are rarely used in songs. Popular messages include Channel Volume (#7), Pan (#10), Modulation Wheel (#1), and Damper pedal i.e. sustain (#64). Their effect on the music is quite evident, while some other CC messages may be such that an average listener does not notice if they have been used or not. Rare CC messages should be mapped to create random –like behaviour to the game or neglected.

5.2 State Parameters

“State” parameters are musical features that stay more or less the same for a longer time. This kind of features are suitable for controlling longer-term game properties such as speed of game and average number of enemies.

According to authors’ experiences, the most important and easily noticeable mapping seems to be connecting song’s tempo to control the overall speed of the game. Other interesting alternatives include e.g. controlling a single moving game object and modifying the difficulty level according to tempo. All MIDI files include a tempo meta-event, and it usually remains constant throughout the song. The tempo of most pieces of music lies between 60 and 140 BPM (beats per minute). Values outside this range can be used to create e.g. some kind of surprises and extreme speeds.

Key and time signature meta-events are much less common than tempo. If the time signature does not exist, a default signature of 4/4 is used. Most songs are in 4/4, so other divisions can be used to create surprises to the game. Key signature information (e.g. C major or F minor) could have an effect on e.g. the mood of the game, time of day, and so on. As lyric and text meta-events are mainly used in karaoke MIDI files (*.kar), a background music reactive game should not rely on them.

Other examples of state parameters include e.g. average polyphony inside a specified time window and “note density” i.e. the number of notes inside a specified time window.

5.3 Transition Parameters

By “transition” parameters, we are referring to significant changes from one musical quality to another. Examples include large intervals in a melody line, the change from silence between two pieces of music to an aggressive beginning of the next piece, large change in note density, and change in time signature. “Transition” parameters can be used for similar purposes to control game content as “event” parameters.

6. AUDIO ASTEROIDS TEST PLATFORM

The authors specified a demonstrative game that could be used to evaluate the "background music reactive games" concept in practice. The idea was to study useful connections between MIDI parameters and game content. An open source game "Maelström" [13], which is based on the well-known arcade game Asteroids, was modified accordingly and renamed "AudioAsteroids". The game was selected because it is simple and intuitive, and has several types of objects flying around that can be controlled by music. The most important modification was the integration of a custom software synthesizer as game's MIDI player engine. The synthesizer was able to analyze the MIDI file during the playback and thus control the actual game engine. The MIDI engine also supports simultaneous real-time input from a MIDI controller.

In the game, the player controls with the alphanumeric keyboard a spaceship that must avoid colliding with asteroids and other objects flying around in space. The player must attempt to shoot dangerous objects such as asteroids, enemy ships, and black holes with the ship's laser weapon. There are also some bonus objects the player must collect in order to get more points, more lives, etc.

6.1 Defining Connections

The players were able to define the connections between MIDI control parameters and game events by themselves. Two types of musical control parameters, namely "event" and "state", were supported. The properties (e.g. speed and amount) of game objects could be controlled by musical events like the pitch of a note and the number of simultaneous notes. Overall speed of the game could be made dependent on the musical tempo, and so on.

Figure 2 shows a window where a mapping for the game is specified. A single connection could be described with the following formula: "A MIDI event from a certain MIDI channel is connected to a game event that affects a game object with a certain factor". Connections can be scaled ("fine-tuned") with a factor between –100% and 100%, which are represented in the game as integers 0 and 200, respectively. High positive percentage values mean that the MIDI event has more effect on the game event. As an example, when the tempo of a MIDI file increases the speed of the game could also increase. Negative percentage values have the opposite effect (e.g. when the tempo increases the speed decreases). Value 0% has no impact, so if all the factors are set to it the game works just like the original Maelström.

Every musical parameter can control multiple game events, and every game event can be controlled by multiple musical parameters. In the latter case, the final game event value is a sum of affecting musical parameters.

AudioAsteroids has also two special connections that are used in a different way from the basic connections. After turning either special connection on, the user has to select a drum sound from the standard General MIDI 1 [13] drum kit, some game object, and specify how many of these objects can exist simultaneously. When the selected drum is played in the song, it will spawn the chosen game object to the screen.

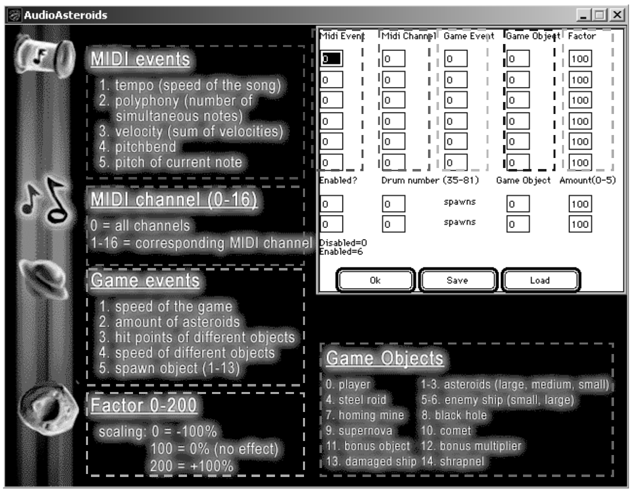


Figure 2. Defining connections in AudioAsteroids.

6.2 Example of Generic Mapping

Although the player can make his or her own mapping before starting the game, this can be an iterative and very time-consuming process. Therefore, a representative set of connections (which would be as illustrative as possible for several types of background music) was defined. The set was stored as a generic connection file that is delivered with the game package. The selected mappings are shown in Figure 3, where:

- Tempo of music controls game's overall speed;
- Combined polyphony of all MIDI channels affects the de-acceleration of UFOs;
- Pitch bend amount on channel 1 controls the number of damaged spaceships on the screen;
- Pitch of current note on channel 5 affects the amount of bonus multipliers;
- Pitch of current note on channel 9 controls the number of small asteroids;
- Pitch of current note on channel 16 spawns steel asteroids to the screen;
- Hit of crash cymbal spawns the smallest amount of UFOs; and
- Snare drum spawns the smallest amount of comets.

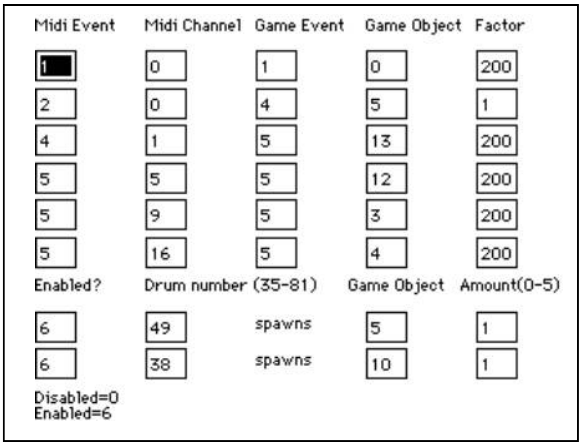


Figure 3. Example of a mapping used in AudioAsteroids.

6.3 Findings from AudioAsteroids

Many MIDI files were tested with AudioAsteroids. In addition to showing the value of a generic mapping, they also proved that the concept works nicely in practice. The selected generic mapping worked well on any MIDI song generating numerous variations to the game.

The authors learned that the most important and easily noticeable mapping was connecting song's tempo to control game speed. Another very perceivable connection was mapping a certain drum being played to generate a specific game object. In the case of other mappings, it was not always evident why some things happened on the screen. However, the fact that different songs made the game appear and behave differently was enough to provide a satisfying gaming experience for people who tried the game. In a sense, any MIDI file in the player's collection could render a different game level.

In general, musical people seemed to enjoy the game more than those who had never played an instrument or listened to a lot of music. They also understood the used mappings better.

7. REAL-TIME MIDI ANALYSIS SOFTWARE FOR SYMBIAN

Mobile phones are becoming increasingly popular devices for playing games and listening to music. So far the main use of their MIDI synthesizers has been the playback of ringing tones. Therefore, mobile phones' MIDI players do not usually have a dedicated callback API (Application Programming Interface). A MIDI file can be played and stopped, but an application does not have any way to get detailed information about the contents of the file.

However, in the case of background music reactive games this information is required. It is also desirable that the game receives the information in real time and in synchrony with the music playback. Because of this, a MIDI analysis software module called "Virtual Sequencer" or "VS" was developed. At the time of programming the software, target platforms were Nokia mobile terminals with Symbian 6.1 (including N-Gage game deck, [12]) and 7.0 operating systems. VS was implemented with C++ programming language.

In the case of a traditional MIDI player engine, a file parser software component reads the MIDI file to be played and sends the parsed data to a sequencer component. Sequencer is responsible for sending scheduled MIDI events to a synthesizer component at appropriate moments. Synthesizer generates the actual audio waveform and sends it forward to be played through loudspeakers or headphones. The implemented Virtual Sequencer software consists of only file parser and sequencer components, both of which are considerably simpler to implement and run than the synthesizer.

Figure 4 illustrates using VS in a game for a mobile phone. As VS has been separated from the actual game engine, it is possible to utilize the same code module again in other games. In the figure, the game engine commands VS and MIDI player blocks to load the same MIDI file. (Here term MIDI player refers to the software or hardware synthesizer of the used mobile phone model.) Both blocks then parse the MIDI data. After receiving a play command, MIDI player's sequencer component starts sending scheduled events to its synthesizer part, while VS's sequencer starts sending control events to the game engine. The game engine then maps these events to selected game parameters, and the game reacts to the music in real time.

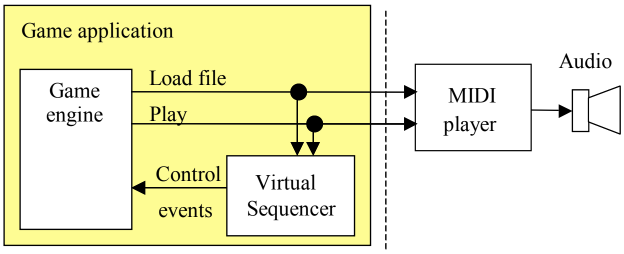


Figure 4. Illustration of using Virtual Sequencer in a game.

Virtual Sequencer is a light-weighted component that does not consume an excessive amount of processing power even when using complex MIDI files. It is a dynamically linked library (DLL) that is used through its API. VS does not generate any sound or interfere with Symbian's native MIDI player. The operation of VS is two-way: It can be instructed as a normal MIDI player (load, play, stop, jump back to song start) and it can make callbacks to inform its host application (e.g. a game) about MIDI events, polyphony levels, etc. that the host wants to be informed about. Some of VS's control events are derived directly from individual MIDI messages, while others are based on some simple calculations. A good example of the latter is "polyphony *N* was exceeded" control event.

The host can be informed via callbacks when any of the following situations occurs during playback:

- Tempo Change message is used in the song;
- A certain number of notes is played simultaneously;
- Note On or Note Off message is used on certain MIDI channels;
- Program Change, Pitch Bend, or Control Change (CC) messages are used on certain MIDI channels;
- SysEx message or meta-event is used in the song;
- Certain notes are played on certain MIDI channels;
- NRPN, Polyphonic or Channel Key Pressure events happen on certain MIDI channels;
- Certain instrument (i.e. defined instrument number) is played (by Note On) on certain MIDI channels; and
- Any MIDI message is used on certain MIDI channels.

Each callback includes all relevant MIDI data as parameters. For example, when Note On is called it is important to know also the note number and its velocity.

8. CONCLUSIONS

This paper elaborated on the concept of "background music reactive games", where game's background music is used to control game parameters and thus actions on the screen. Each song generates a new game level with varying characteristics and difficulty, and players can try to solve playlists of their favorite music or even mobile phone ringing tones.

A test platform called AudioAsteroids was implemented for experimenting with different mappings. A generic mapping was defined in order to test the game behavior with several types of music. It was learned that the most important and easily noticeable connection was controlling game's speed with the musical tempo. AudioAsteroids convinced the authors that the idea of using music to control game content works in practice, and that MIDI is a suitable format for it. By defining an appropriate mapping between musical control parameters and game parameters, it is possible to develop games that behave differently with each piece of background music. The concept introduces a new dimension into the experiences of gameplay when players realize how game content and behavior changes as a result of certain characteristics of the background music.

The benefits of using MIDI were also discussed. Three types of control information ("event", "state", and "transition") can be effectively calculated from MIDI data, and one-to-one mappings between specific musical events and game events are possible. Similar exact information is currently very difficult to extract from digital audio, as practical digital audio analysis algorithms predominantly include detecting changes in sound volume levels, beat tracking, and monophonic pitch detection. To name one possible future implementation, MIDI-control is a way to realize rhythm games that can be played with any MIDI file as background music.

In order to enable the creation of Symbian games on mobile phones, a software component called "Virtual Sequencer" was implemented.

9. FUTURE WORK

An exciting future development would be to combine the activities of playing a game with those of making music with musical controllers. Aspects of group performance could be introduced to this activity. For example, one of the performers could control the hero in the game, while the musical performance of others would create obstacles for the hero. In [15], Blaine and Fels have discussed the principles of collaborative musical interfaces. These guidelines could be adapted into the design of background music reactive multiplayer games.

One constraint in AudioAsteroids was that it is controlled with the alphanumeric keyboard, which most people do not regard as a real-time musical controller (despite it can be used to trigger samples). AudioAsteroids (and many other video games) can be played with as few as three or four distinct buttons, so there are several musical controllers that could be used for this purpose. The use of these controllers as well as new mapping strategies should be examined carefully. In addition, more detailed user testing than done so far would be required.

Theoretically, there are three ways to combine real-time MIDI input with video games like AudioAsteroids. The game could be based on an exclusive real-time performance or the playback of a prefabricated MIDI file with live parts performed on top of it. In addition, the actions of the player (e.g. maneuvering a space ship and shooting) could trigger musical sounds instead of sound effects. All these interesting possibilities can only be explored by implementing new game prototypes.

10. ACKNOWLEDGEMENTS

The authors would like to thank their colleagues Mikko Heikkinen and Antti Eronen as well as Nokia Research Center student trainees Lauri Oikari and Olli Etuaho for their contribution to the project.

11. REFERENCES

[1] Open Sound Control. http://www.opensoundcontrol.org/, 25.1.2006.

[2] Staraoke. http://www.staraoke.fi (only in Finnish), 25.1.2006.

[3] Gitaroo Man. http://ps2.ign.com/objects/015/015184.html, 25.1.2006.

[4] Hämäläinen, P., Mäki-Patola, P., Pulkki, V., and Airas, M. Musical Computer Games Played by Singing. In *Proceedings of the 7th Int. Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 5-8, 2004.

[5] Musical Invaders. http://www.dmi.usherb.ca/minvaders/en/, 25.1.2006.

[6] Musical Space Invaders and Music Scale Teacher. http://www.sharewaresoft.com/Music-Scale-Teacher-download-5736.htm, 25.1.2006 .

[7] Mad Maestro. http://ps2.ign.com/objects/017/017479.html, 25.1.2006.

[8] Vib-Ribbon. http://www.vib-ribbon.com/, 25.1.2006.

[9] Parappa The Rapper. http://www.parappa-the-rapper.com/, 25.1.2006.

[10] Blix. http://www.shockwave.com/sw/content/blix/blix.html, 25.1.2006.

[11] Rez. http://www.sonicteam.com/rez/, 25.1.2006.

[12] Nokia N-Gage. http://www.n-gage.com, 25.1.2006.

[13] Maelström, S. Lantinga/Ambrosia Software, http://www.devolution.com/~slouken/Maelstrom/, 25.1.2006.

[14] Blaine, T. The Convergence of Alternate Controllers and Musical Interfaces in Interactive Entertainment. In *Proceedings of New Interfaces for Musical Expression (NIME) Conference*, Vancouver, Canada, May 26-28, 2005.

[15] Blaine, T., and Fels, S. Design Issues for Collaborative Musical Interfaces and Experiences. In *Proceedings of New Interfaces for Musical Expression (NIME) Conference*, Montreal, Canada, May 22-24, 2003.

[16] Blaine, T., and Forlines, C. JAM-O-WORLD: Evolution of the Jam-O-Drum into the Jam-O-Whirl Gaming Interface. In *Proceedings of New Interfaces for Musical Expression (NIME) Conference*, Dublin, Ireland, May 24-26, 2002.

[17] Holm, J., Havukainen, K., and Arrasvuori, J. Novel Ways to Use Audio in Games. In *Proceedings of Game Developers Conference (GDC)*, San Francisco, USA, March 7-11, 2005.

[18] Holm, J., Havukainen, K., and Arrasvuori, J. Personalizing Game Content Using Audio-Visual Media. In *Proceedings of Advances in Computer Entertainment (ACE) Conference*, Valencia, Spain, June 15-17, 2005.

[19] Cassidy, G., MacDonald, R., and Sykes, J. *The Effects of Aggressive and Relaxing Popular Music on Driving Game Performance and Evaluation.* Abstract in http://www.gamesconference.org/digra2005/viewabstract.php?id=94, 2.4.2006.

Turntable Music in the Digital Era: Designing Alternative Tools for New Turntable Expression

Takuro Mizuta Lippit
smashTV productions / STEIM
Sint Agnietenstraat 6 1012EG Amsterdam,
The Netherlands
+31-06-1515-2782
djsniff@gmail.com

ABSTRACT

Turntable musicians have yet to explore new expressions with digital technology. New higher-level development tools open possibilities for these artists to build their own instruments that can achieve artistic goals commercial products cannot. This paper will present a rough overview on the practice and recent development of turntable music, followed by descriptions of two projects by the author.

KEYWORDS

Turntable music, DJ, turntablist, improvisation, Max/MSP, PIC Microcontroller, Physical Computing

1. INTRODUCTION

The vinyl record has lost its function as a practical audio playback media, gradually being replaced by magnetic tape, CDs, and now by compressed digital audio files. However, this century-old medium is not yet obsolete because its playback device, the turntable, proved to be an irreplaceable performance tool. The performers of this device, the DJ and turntablist, have played significant roles in the emergence of distinct musical genres and have been at the forefront of musical experimentation. They have become a cultural phenomenon replacing the guitar with the turntable as a symbol of youth musical culture [4]. The vinyl record and the turntable have convolved into a single musical instrument that is now universally recognized across multiple musical genres and disciplines.

Today, pro-audio manufacturers have a strong focus on DJ related products. New digital DJ tools both in software and hardware are being developed at a rapid pace, eventually to replace the vinyl as the primary audio source for DJ performances. However these new tools do not necessarily open doors to new musical expression. Simulation and efficiency of existing practice is the main focus of these tools, and artistic experimentation is difficult or just not possible. These products promote a future that only evokes the familiar past. Turntable music has evolved through both aesthetic and technological experimentations by the artists. Affordable computers and high-level programming environments have created a rich context for artists to build their own unique digital performance tools. It would benefit the modern day turntable musician to embrace this new technology and knowledge for both critically reflecting on their practice and creatively projecting new ideas for expression.

2. ASPECTS OF TURNTABLE MUSIC

The following section will introduce some aspects of turntable music that have directly influenced the author's projects. There are other publications that have covered the subject more thoroughly. Kjetil Falkenberg Hansen [5] has written an overview on the practice of turntablists and numerous reports on scientific analysis he has conducted on "scratching." Also, Bill Brewster and Frank Brougton's *Last Night A DJ Saved My Life* [3] is a valuable source for DJ history.

2.1 The DJ, The Turntablist, and The Turntable Materialist

The turntable musician can be roughly divided into three categories based upon their focus on skills and musical practice.

The DJ (sometimes written as Dee Jay to distinguish from the disc jockey) composes a sound-event by skillfully playing an array of recorded music. Many techniques exist for transition between different records, but the most critical technique lies in the ability to create a narrative flow through the selection of records. The DJ must continuously and spontaneously create a linkage in time that inspires a collective musical atmosphere. When the DJ is successful their presence becomes unstable, constantly emerging and withdrawing from the musical consciousness of the audience.

The turntablist in the context of Hip-Hop music, also referred to as the scratch DJ or battle DJ, strictly focus on turntable and DJ mixer manipulation techniques. The term "Turntablism" was born out of their devotion to hours of practice and their strong sense of community. Many of these turntablists will insist on only using the Technics SL-1200 series turntables. The high motor torque created by the company's Direct-Drive technology gives a wonderful tactile feeling for platter manipulation, and many Turntablism skills rely on the physical power of this turntable. In recent years they have pushed instrument manufacturers to produce tools that they desire, sometimes resulting in products with truly innovative designs [15].

So-called "experimental turntablists" have had very little connection to Hip-Hop music or DJ culture, and are part of the experimental and improvisational music scene. Similar to how experimental filmmaker Stan Brackage [2] treated film at its materialistic level and emphasized its physical quality; these musicians sonically extract the fundamental characteristics of the turntable. The recorded sound on the vinyl is merely just one sound source amongst the other rich sound textures, such as the hiss and crackle from the damage of the vinyl, the feedback noise from the pickup needle, and the hum generated by the motor. These musicians each use a different setup of tools, combining numerous effect boxes or handmade devices with the turntable. Improvisation and conscious listening is often their main concern during a performance. Every sonic event is a reflection of what the turntablist decides to listen to, whether it is played from the turntable or something that emerges through not playing. Listening to the 'act of listening' becomes an important aesthetic for both the performer and audience.

2.2 Technology Hacking

Turntable musicians have hacked and modified their tools to achieve new artistic expression. Grandmaster Flash revolutionized the art of DJing when he added a headphone monitor feature to his DJ mixer. Although the practice of modifying and hacking has declined with so many products catered toward the turntablist, some musicians still find it necessary for means of new expression. Janek Schaeffer [12] and DJ Peaky have each constructed multi-arm turntables to play multiple sections off a single vinyl record. Kitundu [7] builds beautiful hybrid turntable-string instruments.

3. TURNTABLE MUSIC IN THE DIGITAL ERA

Though many DJs still prefer to play vinyl records, it seems only a matter of time for better physical interfaces to be developed and become standardized to shift the majority of the practice to digital technology. For the non-DJ turntablist, intuitive tools are scarce and musical experimentation still rely on clunky samplers and guitar effect pedals for the most part.

3.1 Digital DJing

One of the annoyances for a DJ is the bulkiness and weight of the vinyl records that one must carry to perform. Tools that can play more compact media with the same feel as the vinyl on the turntable was eagerly anticipated. Commercial products such as Stanton's Final Scratch [13] and Rane's Serato [11] accomplish this through a combination of time-coded vinyl records and decoding software running on a host computer. Pioneer's DVJ-X1 [10] and Technics SL-DZ1200 [9] are digital hardware devices that simulate the analog turntable. On these devices conventional turntable gestures can playback and manipulate sound data stored on multiple media types such as CD, DVD and flash memory sticks. The primary goal of these products is to perfectly simulate characteristics of the analog turntable and vinyl media. This goal is nearly achieved, but it is questionable whether these products will bring anything more to the practice than relieving the DJ's chronic shoulder pain.

3.2 D'Groove and Ms. Pinky

The potential for new expression in digital turntable music lies in more experimental and open-ended projects. D'Groove [1] by Timothy Beamish is a prototype for a force feedback turntable controller. The most novel feature of this project is that it presents a new possibility to monitor and manipulate digital information. Turntablists already intuitively work with the strong force feedback from the motor. Adding characteristics of the audio source that is being played to the motor's feedback would not only be practical but would also encourage new improvisational techniques. Ms. Pinky [16] by Scott Wardle is a combination of a vinyl record with a constant signal source and a pitch tracking Max/MSP object to read the change in rotation speed. The strength of this tool is that it can easily expand the turntable gestures to be used for any application through the Max/MSP programming environment. Numerous interactive art projects have used Ms. Pinky as an input controller for physical output devices or for live video manipulation.

4. DESIGNING ALTERNATIVE TOOLS FOR TURNTABLE EXPRESSION

Like any musical instrument, the tactile feel of the turntable is essential to its performance. The challenge for designing new tools for the turntablist comes in two-fold: 1) how to translate useful characteristics and information into the digital domain without sacrificing the integrity of the instrument, and 2) how to apply that data in a meaningful way for a musical performance. The following section will discuss two projects by the author. Both projects focus on building peripheral hardware tools to a single Technics turntable and DJ mixer, which are the primary instruments, and creating software environments that encourage new performance methods and aesthetics. Physical Computing [6], C compilers for PIC microcontrollers and Max/MSP are all relatively high-level and resourceful development tools that helped achieve the technological tasks.

4.1 Lupa – A Real-Time Sampling Environment for the Turntable Musician

Lupa is a custom hardware and software toolset that enables the turntable musician to capture, layer, and manipulate the sounds that are generated during a live performance. It is a tool to condense the DJ performance into a shorter span of time, allowing the DJ to spontaneously compose with fragments of sounds rather than entire tracks. Prohibiting all physical interaction and reducing visual interaction between the musician and laptop computer during the performance became a prominent guideline for the design and development.

4.1.1. Hardware and Software

The hardware (see figure 1) was designed after two previous prototypes [8]. An 18F452 PIC microcontroller is used to translate numerous sensor inputs to MIDI messages. Sounds are sampled by a foot switch and later transformed by different modes of the joystick. Large game components are chosen to match the overall gestural movement of the turntable musician.

Figure 1. Lupa - hardware controller.

The software (see figure 2) is written in Max/MSP with four sample banks displaying the waveform, loop point, time progression and volume. The software not only records sound, but also can record every physical input applied to the controller. While the large red button is pushed down, all parameter changes that take place on each sample bank is stored. This data is played back as an automated sequence when the same red button is hit twice.

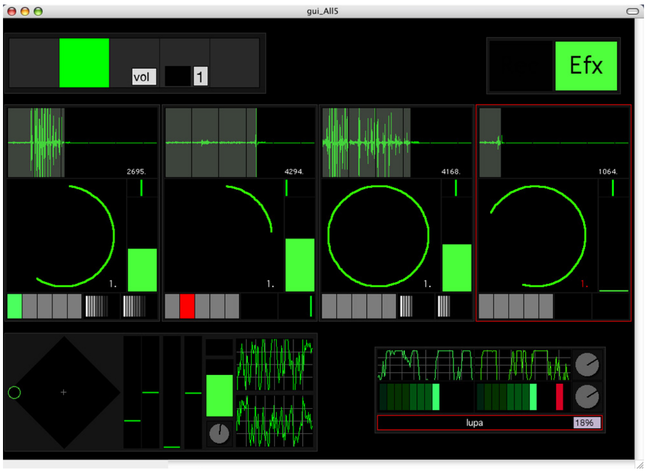


Figure 2. Lupa - software interface.

4.1.2. "At Glance" Graphical interface

The graphical interface needed to display data that the hardware interface could not represent, such as the sample waveform and time progression of the loop. It was important to design a single layer layout that was comprehensible at a glance and not engage too much of the performer's attention. Text labeling was kept minimal both to save screen space and to encourage the performer to learn the layout through practice. The LCD object in Max is used extensively to create large planes with moving parameters or for simple on/off indicators.

4.1.3. Anti-presets, anti-automation, anti-prerecorded samples

Building a system that was coherent to the audience was another strong motivation for this project. This was a reaction to the typical laptop musician and performance that was becoming prominent at the time. As a result, no parameter presets, automation algorithms, and prerecorded samples can be used in Lupa. Every sound generation and manipulation must take place during the performance on stage. Therefore all sound sources are restricted to what is generated from the turntable and any automation was merely a recorded sequence of a physical input that previously happened. The only predetermined factors are the instrument, the system, and the records that the turntable musician decides to bring. The execution completely depends on the skill of the musician and is often improvised.

4.1.4. Evaluation and future development

Lupa proved to be a successful tool, both as an intuitive sampling environment for turntable musicians and as a performance with clear sound relationships for the audience to understand. The undetermined nature made every performance unique, but at the same time placed enormous pressure on the performer. Unless constant attention is paid to multiple sound events and changes are applied, the sound structure quickly becomes repetitious and boring. Virtually the turntablist becomes a one-man marching band where one must question whether this much effort is contributing or distracting from the desired musical expression. Successful performances have been done with two turntablists, and this direction may be further considered. The system still requires the performer to look at the computer monitor for some critical parameters. Another hardware update is planned with improved visual feedback on the hardware itself. MIDI is sufficient for simple fader and switch data transmission, but USB will be chosen to power the device through the bus and reduce an adaptor that one might forget to bring to a gig.

4.2 Audile – Semi-Automated DSP Effects for Turntable Improvisations

Audile is a set of DSP effects controlled by a custom USB controller. Max/MSP is used to write the signal processing software and an 18F4550 PIC microcontroller is used to read the sensory input and transmit the data through the USB HID class protocol. The purpose was to create a dedicated tool for the turntable musician to perform with other improvisational musicians. Turntablists, especially scratch DJs, occupy both hands during sound generation with little bandwidth to control signal-processing parameters. Foot switches and expression pedals are useful interfaces, but are limited by the coordination of the foot. In order to create an instrument that is controllable without disrupting the existing performance flow, two approaches are taken.

4.1.2. Semi-automated parameter shifting

Inspired by the rotation of the turntable, constant movement is given to the parameters of the signal processing modules in the program (see figure 3). The movement can be stopped or modulated through the physical controller interface. This allows the turntablist to treat the effect module as an autonomous machine, each with a distinct but constant characteristic to work with and alter. *Turn* is a stereo delay module that changes its parameter based on a rotating dot. The position of the dot specifies the different delay time in both channels, with the maximum delay time depending on its rotating diameter. This diameter is changed through a foot pedal. The rotating movement can be turned on and off with a switch, and the speed of rotation and feedback amount are controlled by potentiometers. A 360-degree potentiometer disc controls the position of the dot. *Ghost* is a module that moves a point between four locations in a random "drunken walk" manner. Each location is assigned with reverb, delay, and distortion. The parameters of these effects are changed by the proximity of the drifting point. The foot switch triggers the drift and a mini joystick influences the tendency mask on which direction it moves to.

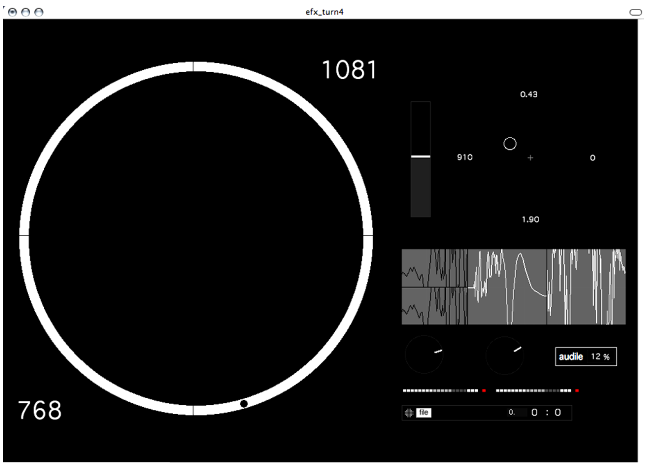


Figure 3. Audile - software interface.

4.2.2. Extracting gestural information from platter manipulation

There are several methods one can use to translate the gestural movements on the turntable to useful data. When a method that was portable and least intrusive was sought, the Tascam TT-M1 turntable controller (see figure 4) [14] became an ideal interface to incorporate. TT-M1 is a small device that attaches to the turntable and reads the rotation speed of the platter. Unlike the encoded records of Final Scratch or Ms. Pinky that are meant to play digital files on a computer, this device can read the platter movement while playing the sound from the vinyl record. Because this device is intended to be used with designated CD turntables, some reverse engineering was needed. The TT-M1 works like an optical mouse. An encoder disc attached to a wheel rotates at the same speed as the turntable sending pulses through an IR receiver. The firmware on the PIC chip interprets these pulses and transmits it to the host application. This data is used to change the amount of reverberation applied to the incoming sound. Another potentiometer is used to control the subtlety of this reverberation.

Figure 4. Tascam TT-M1 and Audile hardware interface.

4.2.3. Future development

Audile is still under development. The use of semi-automated effects and simple physical controllers has minimized the performer's attention to the computer monitor and allows more concentration towards the sound events. However, musical mapping of the turntablist's gestures needs to be further developed. Simply mapping reverberation to platter speed proved not to be that sonically interesting. Other gestures, such as fader movement and mouth movement will be explored. Additionally, Max/MSP does not have an object to receive and send HID data. Creating such an external object will benefit not only for this project but for others as well.

5. CONCLUSION

Designing new digital tools not only result in new sound textures and sound arrangements by means of computation, but also bring forth characteristics of existing tools that would otherwise not be noticed. This process has contributed to my expression as a turntable musician, forcing me to look deeper into my musical practice. New open-ended and developmental tools that are becoming readily available provide another option for the turntable musician to explore their musical possibilities and advance the practice to a new level.

6. ACKNOWLEDGEMENTS

This research was conducted during my internship at STEIM, Studio for Electro-Instrumental Music in Amsterdam and with the support of the Cultural Ministry of Japan. I am grateful for all their help and understanding.

7. REFERENCES

[1] Beamish, T., Maclean, K., Fels, S., *Manipulating Music: Multimodal Interaction for DJs, Proceedings of CHI2004* – Vienna, Austria

[2] Brackage, S., *Mothlight*, 1963

[3] Brewster, B., Broughton, F., *Last Night a DJ Save My Life: The History of the Disc Jockey*, London: Groove Press, 2000

[4] Collins, N., *ed. Leonardo Music Journal 13: Groove, Pit and Wave*, Cambridge: MIT Press, 2003.

[5] Hansen, K., http://www.speech.kth.se/~hansen, 2006

[6] Igoe, T., O'Sullivan, D., *Physical Computing: Sensing and Controlling the Physical World with Computers*, Boston: Thomson Course Technology, 2004

[7] Kitundu, http://www.kitundu.com/instruments.html, 2004

[8] Lippit, T., *Real-time Sampling System for the Turntablist – version2 16padjoystickcontroller, Proceedings of NIME, NIME04* – Hamamatsu, Japan

[9] Panasonic Corporations, Technics SL-DZ1200, http://www.panasonic.com/consumer_electronics/technics_dj/prod_intro_sldz1200.asp, 2004

[10] Pioneer Electronics, DVJ-X1, http://www.pioneerprodj.com, 2005

[11] Rane Corporations, Serato Scratch Live, http://www.rane.com/scratch.html, 2005

[12] Schaeffer, J., www.audioh.com 2005

[13] Stanton Magnetics, Final Scratch 2, http://www.stantondj.com/v2/fs/prod_fs2.asp, 2005

[14] TEAC Corporations, Tascam TT-M1 Turntable Controller, http://www.tascamdj.com/TT-M1.html, 200Vestax Corporations, QFO DJ Turntable, http://vestax.com/v/products/players/qfole.html, 2004

[15] Vestax Corporations, QFO DJ Turntable, http://vestax.com/v/products/players/qfole.html, 2004

[16] Wardle, Scott, Ms P/TT-M1.html, 2004

spinCycle: a Color-Tracking Turntable Sequencer

Spencer Kiser
Interactive Telecommunications Program
New York University
721 Broadway
New York, NY 10003
spencerk@nyu.edu

ABSTRACT

This report presents an interface for musical performance called the spinCycle. spinCycle enables performers to make visual patterns with brightly colored objects on a spinning turntable platter that get translated into musical arrangements in real-time. I will briefly describe the hardware implementation and the sound generation logic used, as well as provide a historical background for the project.

Keywords

Color-tracking, turntable, visualization, interactivity, synesthesia

1.INTRODUCTION

The original spinCycle consists of a turntable and video camera mounted to scan the radius of the platter and connected to a multimedia computer. Several variations of the interface have been implemented, each involving different audio content, but all consisting of the same hardware setup. Translucent plexiglass disks, with diameters of 2 or 3 inches, and tinted red, yellow or blue, are used as sound objects that can be arranged in visual patterns on the platter of the turntable. As the turntable spins, the video camera acts analogously to the needle and head cartridge of a traditional turntable, transducing sound from the colors it senses rather than from vibrations. A visual representation of what the camera sees is displayed, providing visual feedback to the audience, informing them of the correspondence between color and sound.

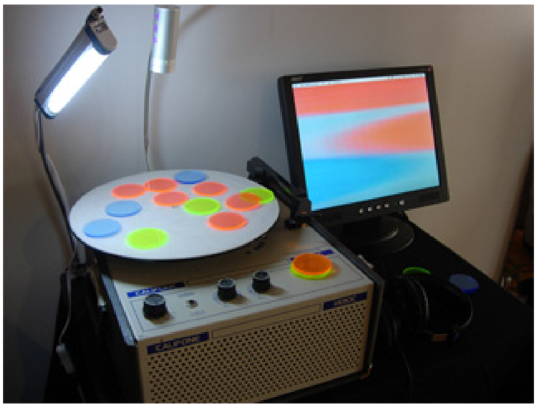


Figure 1. spinCycle, an early installation prototype

2.BACKGROUND

Recorded history of efforts to develop a relationship between color and sound dates back to the ancient Chinese and Persians. In the West, it was Sir Isaac Newton's system for mapping color to tones, laid out in his treatise "Opticks" (1704), that became the most prevalent scheme for connecting musical notes and colors. He arbitrarily divided the spectrum of visible light into seven colors (red, orange, yellow, green, blue, indigo and violet), and made a connection between their mathematical relationship to each other and the relationship between the notes of the musical scale.[1]

There are also many precedents of mapping color to sound in the design of musical instruments. Father Louis Bertrand Castel, a Jesuit monk, built an Ocular Harpsichord around 1730 that involved a six-foot frame above a regular harpsichord. A system of pulleys and rope would lift small curtains on the frame to reveal candles filtered by colored glass in time with the playing of the instrument. Each curtain corresponded to a key on the harpsichord. Development of the color organ continued, and several inventors have had measured success with the device. In the late nineteenth century Bainbridge Bishop outfitted an organ with stained glass windows that were lit based on the keys pressed, and Alexander Wallace Rimington requested that the audience wear white to enhance the effect of the lights projected from his color organ.[2]

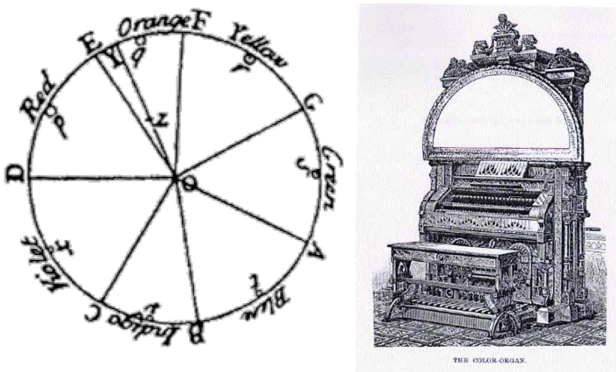


Figure 2. Sir Isaac Newton's Color Wheel (left) and Bainbridge Bishop's Color Organ (right)

There is a rich history of performers and composers using turntables in novel ways to create and perform new arrangements of prerecorded music. John Cage was one of the first to use the turntable as an instrument in Imaginary Landscape No. 1, which called for records to be played on a variable-speed record player.[3] Nikita Pashenkov also provides an excellent history of optical turntables in his paper presented in NIME 2003, where he mentions the Piano Optophonique created by Vladimir Baranoff-Rossine, which "generated sounds and projected revolving patterns onto a wall or ceiling by directing a bright light through a series of revolving painted glass disks, filters, mirrors and lenses."[4] Gideon D'Arcangelo points out that truly interactive uses of the turntable as an instrument came in the 1970's when the advent of hip-hop coincided with art experiments being done by Christian Marclay.[5]

More current contributions to the field of optical turntables includes Jacques Dudon's Photosonic Instrument,[6] which involves light shining through optical disks and filter onto a photocell, and Miyakodub's Video Turntable, which employs color detection to trigger sounds, events and other musical parameters (delay, pan, on/off).[7]

Other recently developed interfaces worth noting that involve using tangible objects on a surface to arrange and create music are the reacTable*, developed by the Music Technology Group at the Universitat Pompeu Fabra[8], and the Music Table from ATR Media Information Science Laboratories.[9]

3.IMPLEMENTATION

3.1 Hardware

The current prototype consists of a traditional turntable capable of 16 rpm, white slipmat, firewire web cam on an adjustable gooseneck mount, small fluorescent lamp with adjustable mount and multimedia capable computer. The disks are made of fluorescent tinted plexiglass (red, yellow and blue), which provide vivid colors to ease color recognition by the computer.

In development at the moment is a custom turntable with a wider platter and continuously adjustable speed. The new platter has raised edges to prevent the disks from sliding off while spinning, which only happens when many are stacked on top of each other. The web camera is permanently affixed to the base of the turntable. The new version also provides a knob to adjust speed of the turntable and a series of buttons that can be used to select and assign the colors to be sensed. There are also plans to provide hardware controls that adjust parameters related to video sensing, such as white balance, focus and saturation; these parameters are adjusted in a software interface in the current prototype.

3.2 Software

The application was developed in Max/MSP/Jitter. Basic functions of the application include video sensing calibration (white balance, focus, saturation), selection of colors to be sensed, assignment of desired sound to color, and volume control for each sound channel.

The original performance prototype was developed for Gideon D'Arcangelo's NIME class at ITP in Spring 2005. The theme for the class was "Tools for the Remix", and accordingly, the prototype was designed to use sound coming from one channel of a two-channel turntable mixer as a source. Samples were selected on the fly and then remixed with the original record. The sensing algorithm uses edge detection of the colored disks to determine when a sound should be triggered, and granular synthesis is employed to play samples associated to disks located closer to the center faster than, but at the same pitch as those closer to the edge. A representation of the image that the camera sees at any given time is projected on to a screen behind the performer, which provides the audience with a connection between the colors and the sounds being emitted, and evokes a hypnotic synesthetic experience.

The installation prototype has two forms: drum machine and sine wave generator. The drum machine version has bass drum, snare and hi-hat color-sound assignment; users arrange drum patterns in real time and can make interesting syncopated patterns by shifting the disks slightly as the platter turns.

The sine wave generator installation prototype allows users to experiment with more colors and sounds by stacking and thereby combining the colors to make green, orange and violet. Each color is hard coded to sine waves at frequencies that make up a chord to insure harmonious results. The video sensing algorithm maps the area of a given color to the amplitude of the corresponding sound, thereby altering the envelope.

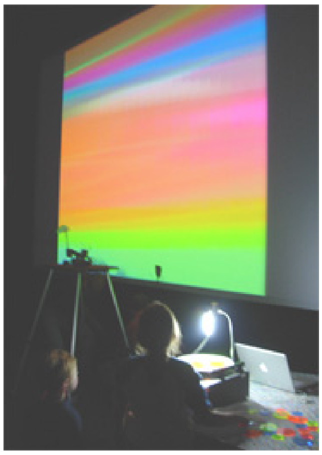


Figure 3. spinCycle sine wave generator installation

4.CONCLUSION

Using the turntable as the basis for an interface for musical expression is not a new idea. The familiarity of the turntable and its constant circular motion provide a good basis upon which to build creative devices for sequencing and looping music. What makes spinCycle distinctive is the simplicity of its interface that invites novices to arrange music and experiment effortlessly, but also provides a challenging experience for performers that can be perfected over time. The synesthetic nature also makes it well-suited for multimedia installations intended for children and adults alike, as well as provides a dazzling visual component for performance.

5.ACKNOWLEDGEMENTS

The following people were integral to the realization of the spinCycle: Gideon D'Arcangelo, Luke DuBois, Jamie Allen, MJ Hitchings, and my wife, Chih-Yi Chou.

6.REFERENCES

[1] Hutchison, Neils. Music For Measure. 2004 *http://home.vicnet.au/~opticks1.htm*

[2] *http://www.paradise2012.com/visualMusic/musima/*

[3] *http://www.medienkunstnetz.de/werke/imaginary-landscape-1/*

[4] Pashenkov, Nikita, "A New Mix of Forgotten Technology: Sound Generation, Sequencing and Performance Using an Optical Turntable" *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME-04), Hamamatsu, Japan.*

[5] D'Arcangelo, Gideon, "Recycling Music, Answering Back: Toward an Oral Tradition of Electronic Music" Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME-04), Hamamatsu, Japan.

[6] Arfib, Daniel and Dudon, Jacques, "A digital emulator of the photosonic instrument" Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02), Montreal, Canada.

[7] Miyakodub, *http://miyakodub.com/color.html*

[8] http://www.iua.upf.es/mtg/reacTable/

[9] http://www.siggraph.org/s2003/conference/etech/augmented.html

The Chopping Board: Real-time Sample Editor

Jason Lee
Interactive Telecommunications Program, New York University
721 Broadway, 4th Floor
New York, NY 10003 USA
+1 9176718164
jwl283@nyu.edu

ABSTRACT
"Chopping" is a compositional technique used by many hip-hop producers and other artists working with audio samples. This technique involves separating a sample into smaller segments and re-sequencing them into a new composition. The Chopping Board is a composition and performance tool designed specifically for this technique. It is meant to inject some of the elements of traditional musicianship into sample–based music.

Keywords
Hip-hop, sampling, musical interface, gesture, plunderphonics, mash-up

1.INTRODUCTION
1.1Background
As a hip-hop producer and DJ, my music is primarily sample based. My compositions are comprised of small audio snippets from previously recorded music. This is a very common practice among producers working in hip-hop, as well as other forms of music. Chopping a sample came about because of necessity, and as a logical progression. In the early 90's when copyright laws became strict, hip-hop artists had to devise new methods in order to avoid legal repercussions. By chopping a sample, it becomes a lot harder to recognize and thus a lot harder to prove what the source is. It also allows for more flexibility in terms of arrangement and tempo. A looped sample is restricted to the original tempo and arrangement whereas a chopped sample is not. Some feel that chopping is a more creative way to approach to sampling. By re-arranging the sample, the producer is adding or building on the original. The original sample is not being used verbatim.

The use of pre-recorded music to create compositions is a fairly popular technique that spans multiple genres. The introduction of the Fairlight CMI and other digital samplers in the early 1980's made it possible to easily replicate and manipulate digital samples. Pop groups like Art of Noise and Malcolm McLaren all used samples prominently in their music. Hip hop producers gravitated towards sampling as a logical extension of their turntable based music. Around the same time, other artists were doing similar work. Christian Marclay used records and turntables to make installations and sound collages. The legal ramifications of sampling have also made it a subject of discussion. Works like Danger Mouse's "The Grey Album" have brought sampling and copyright infringement to the forefront of pop culture. The artistic merit of sampling is often debated as well. John Oswald's "Plunderphonics" and Gideon D'Arcangelo's "Recycling Music" both offer interesting arguments in favor of sampling and general re-use of music.

1.2Reasons For Sampling
Some may be curious as to what the purpose of going through this process is when one could simply re-play the desired sample. Part of the appeal of sampling is that you can capture the production sound of an era of recording that no longer exists. Sampling enthusiasts feel that the music of the 60's and 70's "sounds" better than current recordings due to it's use of analog equipment. Many feel that the recordings of this time have more "warmth" and "grit". Another appeal of samples is its use of what I like to call "accidental sound". Lets say our source sample is chopped into four notes, "a", "b", "c", and "d". It is quite possible that at the tail end of the "c" sample there is a scream from a band member, or some other kind of ambient sound. Every time the "c" is triggered, the scream will play as well. If it is triggered frequently, the scream will take on a percussive quality. If the new composition is at a different tempo, the scream may be slightly off beat, but it may add a unique swing to the rhythm. The scream finds it's way into the composition by accident. But it will add an element to composition that would probably not exist otherwise. These situations arise frequently when working with samples. Sometimes they can add a great deal to the composition.

1.3Problems with Sampling
Sometimes, working with samples can feel more programmatic than musical. A fair amount of editing and prep work must be done before one can begin composing. This process can be time-consuming and cumbersome. It is possible that after all the editing is done, one might decide that the sample is unsuitable. This prep work may even prevent certain musical experimentation. Because the music is made using some kind of sequencing program, rigidity and precision are a part of its nature. After it is sequenced, the piece will never be off beat. To some, this is a blessing. However, I cannot help but think that hip-hop and electronic music production could benefit from some of what I am calling "human factors". Live musicians add subtle nuances to music, which can contribute greatly to a song. Musicians interact with the music, as well as the people around them. This idea of the "human element" is part of what makes music

exciting. Some, if not most of this, gets lost in electronic music.

2.THE CHOPPING BOARD

The Chopping Board is an interface designed specifically for composition and performance using samples. It injects some of the characteristics of traditional musicianship into sample-based music. The Chopping Board maps a sample across a space approximately 18” long called the “editing pad”. The editing pad is a physical representation of the sample, the left side being the beginning, the right side being the end. Wherever the editing pad is triggered, the sample will begin playback at the equivalent point in its timeline. The idea is that the user triggers various points of the sample in a musical fashion, in effect chopping the sample. There are also controls for volume, pitch, and sample selection.

One of the reasons that audiences watch and respect musicians is because they have developed skills that not all have. I hope that the Chopping Board will require some of these skills as well. For example, a guitar player must accurately and quickly position her fingers in order to play the right notes. With the Chopping Board, the player must memorize specific locations and trigger them accurately and quickly as well. I am hoping that it will parallel traditional instruments in their need for manual dexterity, timing and precision.

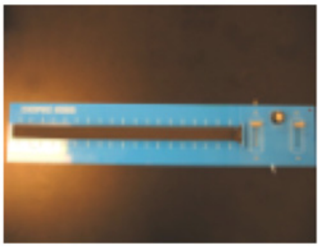


Figure 1. The Chopping Board

Like traditional instruments, and unlike most sample-based music, the Chopping Board is played live for the duration of a performance. The user must be able to play the part with precision for an extended amount of time. It is my hope that this will add some of the “human factors” that I discussed into sample-based music. Instead of playing back music from a pre-recorded format, the Chopping Board will allow some of the elements of the music to be played back live. This could be more engaging for audiences. It will allow for more musical improvisation and interaction as well. The user positions his or her self above the Chopping Board, much like you would when playing a vibraphone or a 12-string guitar. The instrument is played by touching various locations on the editing pad. The musician’s interaction with the instrument is completely visible to the audience, allowing them to see the direct correlation between the musician, the gesture, the instrument, and the sound. The Chopping Board is also designed to streamline the technical process of sample-based composition. The producer no longer has to edit and prep the sample. As soon as the sample is loaded, the producer can begin to “chop”. This will allow composers to work more quickly and intuitively. It will also allow for more experimentation.

3.TECHNICAL SPECIFICATIONS

The Chopping Board needs to measure the x-location of the users hand on the editing pad. Currently it uses a combination of an infrared sensor and a touch sensor to do this. The IR sensor is positioned on the right side of the editing pad. A touch senor spans the length of the editing pad. When the user touches the pad, the IR sensor takes a reading of the distance of users hand, which is directly in its path. This data is sent to a PIC microcontroller, where it is converted to MIDI data. The MIDI data is sent to a computer running a MAX/MSP patch. The audio is played back out of the sound card, into some kind of speaker or PA system. The MIDI protocol offers a resolution of 127, which seems fairly low, but I have not felt limitations in this respect. However, because of the combination of the IR and touch sensors, there is some noticeable latency in the audio playback.

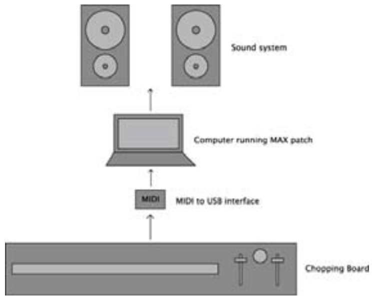


Figure 2. System layout

The Chopping Board has faders for both volume and pitch. There is a sample select knob, which sends commands to the MAX patch about which sample to load. There is a switch for toggling between “momentary” and “sustain” playback modes. In momentary mode, the sample will only play for as long as the editing pad is pressed. In sustain mode, the sample will play all the way through after being trigged. These modes are designed for various composition techniques. All of the data from these controls are sent to the PIC, and output as MIDI data. No audio signals flow through the Chopping Board.

Online documentation and a video demo for the Chopping Board can be found at: http://jasonwlee.com/chop.htm.

4.REFERENCES

[1] D’Arcangelo, Gideon. *Recycling Music, Answering Back: Toward an Oral Tradition of Electronic Music*. Proceedings of NIME, NIME 2004 – Shizuoka University of Art and Culture.

[2] Lippit, Takuro Mizuta. *Realtime Sampling System for the Turntablist Version 2: 16padjoystickcontroller*. Proceedings of NIME, NIME 2004 – Shizuoka University of Art and Culture.

[3] Oswald, John. *Plunderphonics or Audio Piracy as a Compositional Prerogative.* http://www.plunderphonics.com/xhtml/xplunder.html (1985)

[4] Schloss, Joseph G. *Making Beats: The Art of Sample Based Hip-hop*. Middletown, Connecticut: Wesleyan University Press, 2004.

A Tactile Closed-Loop Device for Musical Interaction

Staas de Jong
LIACS
Media Technology program, Leiden University
staas@liacs.nl

ABSTRACT
This paper presents a device implementing a closed tactile loop for musical interaction, based on a small freely held magnet which serves as the medium for both input and output. The component parts as well as an example of its programmable behaviour are described.

Keywords
Musical controller, programmable haptic and tactile feedback.

1. INTRODUCTION
Musical expression needs its own specialized instruments, and haptic feedback is essential to their development. Both the promise and necessity of variable force feedback for improving the expressive potential of electronic musical instruments have been noted, for instance in [6] and [1], respectively.

Another important notion concerns the fruitfulness of exploring and exploiting well-developed and readily available general input devices as musical controllers. This has been stressed in [7], illustrated by the example of a digitizing tablet. Another example is the use of force feedback joysticks and mice in [3].

At the same time, an integrated approach to musical instrument design - as for instance advocated in [5] - can motivate the conception and construction of controllers specifically intended for musical use. It is with this in mind that this paper presents a small but complete platform for investigating touch as a shared medium between performer and instrument.

2. THE DEVICE
2.1 Sensing
The device is operated by a small, handheld permanent magnet (Figure 1). The proximity of this magnet to the device surface is used as input: an LDR (light-dependent resistor) measures the gradual blocking of environment light caused by changes in its distance.

The LDR circuit's measurements are converted to digital input using a MIDI-based solution (developed at the Royal Conservatory of The Hague) giving a resolution of 7 bits at a sampling rate of approximately 100 Hz. This input is read out for further processing by a Max/MSP patch running on a suitable host computer.

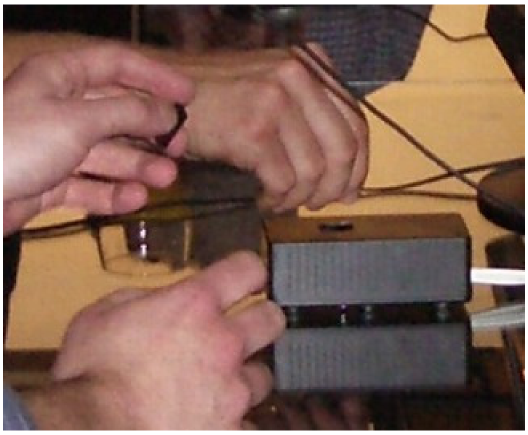


Figure 1. The device during use.

In order to transparently adapt to a wide range of lighting situations, only the range of values actually coming in is used: calibration is performed simply by doing a full movement towards the sensor after switching on. The resolution is further decreased by an additional detection threshold of 5%, to avoid slow fluctuations in environment light (for instance, passing clouds) being detected as an object closing in. A low-sensitivity backup input has been added for situations where the 50 Hz. flickering of light bulbs cannot be avoided.

After these and various other improvements, the sensor achieves an effective resolution of 105 steps in good lighting conditions (daylight or stable electrical light). This over a typical distance range of 0 - 4 cm. straight above the sensor; where the maximum distance can vary depending on the angle of incoming environment light.

2.2 Force feedback
The sensor is mounted directly onto an electromagnet, which is used to apply a variable force to the freely held permanent magnet. This force is invisible in the sense that it is transferred through air without the use of moving mechanical parts. In this, the use of electromagnetism here differs from the approach in [2].

The electromagnet was taken from an RXAB010 electro-mechanical relay and is used with a maximum current of 2 A. It is capable of generating a rejecting force sufficient to make single-handedly touching it with the permanent magnet nearly impossible; fluctuations in this force can be perceived in a range of 0 - 10 cm. above the coil. (Consequently, the playability range of the device is determined by its sensing range, which is smaller.) The electromagnet is capable of attraction as well as rejection, without the need for current reversal: when current is low, magnetization of its core by the nearby permanent magnet generates an attracting force.

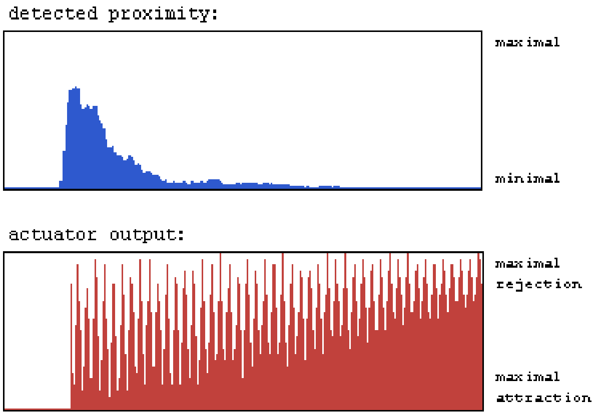


Figure 2. An example behaviour for controlling percussive sound. (Time increases towards the right, 100 samples/sec.)

The same MIDI device mentioned before controls the circuitry of the electromagnet. Here a fixed calibration has been used for digital output, resulting in 26 steps of resolution between maximal attraction and maximal rejection, at a sampling rate of 200 Hz. In order to be able to use Max/MSP's powerful facilities for generating complex signals, the output object for controlling the electromagnet takes an arbitrary audio-rate signal as input. There is no filtering or averaging stage before downsampling in order to allow for the generation of sharp block pulses.

3. EXAMPLE OF PROGRAMMABLE BEHAVIOUR

Besides handling the input and output for the sensor/actuator, the Max/MSP patch also generates sound. In one implemented example behaviour, passing a certain proximity threshold with the handheld magnet triggers synthesis of a low percussive sound. Simultaneously, a maximal rejecting force is activated, heavily modulated by a gradually decreasing triangular wave oscillating between rejection and attraction. When making a percussive movement towards the device, this creates a sense of touching a surface and causing a reverberation, which gradually dies away with the sound it produces.

Figure 2 shows the simultaneous tactile input and output of the system during such a percussive movement. The spikes in the actuator output can be seen reflected in the proximity input. (These and other example behaviours were successfully tried out by the visitors of an exhibition featuring the device.)

4. CONCLUSION AND FUTURE WORK

This paper has presented the implementation of a closed tactile loop for musical interaction, based on a freely held permanent magnet in combination with a fixed electromagnet. Work is under way to improve the technology underlying the device, especially the quality of its I/O. Also in progress is the implementation of more subtle tactile hints. Future work includes the integration of tactile behaviour with different mapping strategies [4] in order to achieve increasingly engaging control; as well as a more general goal of investigating more and more refined interaction techniques.

5. ACKNOWLEDGMENTS

Many thanks go to Lex van den Broek of the Royal Conservatory of The Hague for generously sharing his expertise in current control. Thanks also to Emiel de Jong for donating electromagnets.

6. REFERENCES

[1] Bongers, B. Physical Interfaces in the Electronic Arts. *Chapter for the IRCAM book and CDROM "Trends in Gestural Control of Music", submitted*, 2000.

[2] Cadoz, C., Luciani, A., Florens, J., and Castagné, N. Artistic Creation and Computer Interactive Multisensory Simulation Force Feedback Gesture Transducers. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, Montreal, Canada, 2003.

[3] Howard, D.M., Rimell, S., and Hunt, A.D. Force Feedback Gesture Controlled Physical Modeling Synthesis. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, Montreal, Canada, 2003.

[4] Hunt, A., and Wanderley, M.M. Mapping Performer Parameters to Synthesis Engines. *Organised Sound 7:2, pp. 97-108*, 2002.

[5] Jordà, S. FMOL: Toward User-Friendly Sophisticated New Musical Instruments. *Computer Music Journal, 26:3, pp. 23-29*, 2002.

[6] Paradiso, J.A. The Edge of NIME - from Cult to Conference. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02)*, Dublin, Ireland, 2002.

[7] Wessel, D., and Wright, M. Problems and Prospects for Intimate Musical Control of Computers. *CHI'01 Workshop New Interfaces for Musical Expression (NIME'01)*, Seattle, USA, 2001.

PETECUBE: a Multimodal Feedback Interface

Peter Bennett
University of Brighton
Faculty of Arts and Architecture
Grand Parade, Brighton, BN2 0JY
peterdavidbennett@gmail.com

ABSTRACT
The PETECUBE project consists of a series of musical interfaces designed to explore multi-modal feedback. This paper will briefly describe the definition of multimodal feedback, the aim of the project, the development of the first PETECUBE and proposed further work.

Keywords
Multi-modal Feedback. Haptics. Musical Instrument.

1. INTRODUCTION
A multimodal system can be defined as a system that "supports communication with the user through different modalities such as voice, gesture and typing." [1]. The 'mode' term of multimodality can be used to refer to both mode and modality. Mode refers to "a state that determines the way information is interpreted to extract or convey meaning" [1] whereas modality refers to "the type of communication channel used to convey or acquire information."[1]. 'Feedback' can be defined as "the return of part of the output of an electronic circuit, device, or mechanical system to its input, so modifying its characteristics" [2]. Hence, a multimodal feedback interface can be defined as an interface with multiple communication channels that returns a portion of its output to the input of the system. The output of the system in the case of an instrument is the sound produced, and the input can be seen as the user playing the instrument.

Many instruments have been developed that use various forms of feedback, however it is felt by the author that the instruments are normally biased towards one of the particular senses and that other sensory feedback is somewhat neglected. The aim of this project is to create a series of instruments in which all forms of feedback are equally considered, and more importantly are used together in a coherent whole. Of the five Aristotelian senses (sight, hearing, touch, smell and taste), it has been decided to concentrate upon the three that are most pertinent to playing a musical instrument; sight, hearing and touch. All musical instruments already incorporate *passive* feedback of all of these senses (i.e. you can see, hear and feel a piano or guitar). However, the interest of this project is in *active* feedback, so that the designer of the instrument can specify how an instrument will react within each of those modalities.

Successful research that explores this area is the PHASE project [3]. The PHASE group have implemented a multimodal installation that offers haptic, visual and audio feedback operating on a model of a turntable like device with both a 'writing' and a 'playing' head. The PETECUBE project differs from this in several key ways. Firstly, the PETECUBE aims to embody the multimodal feedback within a single object. Secondly, the PETECUBE is designed for live performance, not an installation, so the size and complexity of the setup is restricted by the need for portability. Thirdly, and most importantly, the PETECUBE is designed as part of a series in which each cube is limited in its modalities, and as such, individual cubes should not be considered complete instruments, but as investigations into particular combinations of sensory feedback.

2. DESIGN
To impose some limits on the design of the interface, it has been decided to limit the physical form to the shape of a cube (see figure 1). Although arbitrarily chosen, the cube was found to be a useful design; it is an ideal shape on which to mount sensors and actuators, a 2D representation of a cube is easily seen as a 3D cube (figure 3), it is a robust shape ideal for rough handling and it is easily grasped by the hand (figure 4). Another consideration is that a cube is not an imitation of a conventional instrument, so that users should approach it without any preconceptions on how to play it.

Figure 1. A Prototype PETECUBE.

The system diagram below (figure 2) shows how the PETECUBE is organised. The three levels depict the user interface level at the top, the hardware level in the middle, and the software level at the bottom. At the top level the user can manipulate the PETECUBE whilst also receiving three forms of sensory feedback; vibration from the cube, sound from the speakers and visualization from a monitor or projection. The middle level consists of hardware to communicate between the user interface level and the software level. The bottom level consists of three separate software programs to handle each of the feedback modalities.

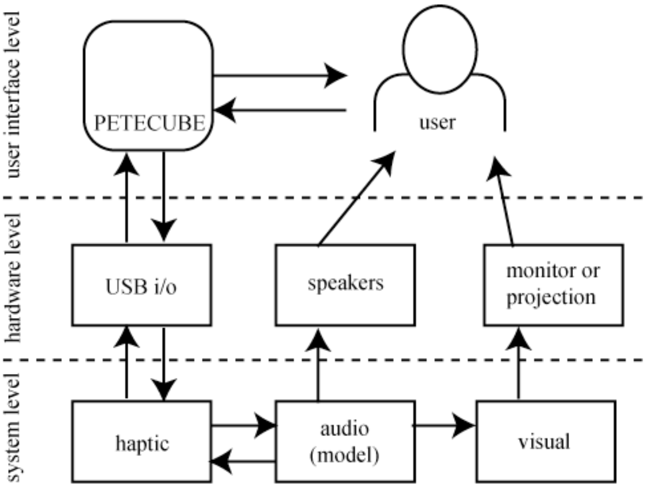


Figure 2. System Diagram.

It was decided to break up the software level in this manner to create a modular system. Communication between the programs is achieved using Open Sound Control [4] over a network. There are two main benefits to this approach. Firstly, the programs can be created in different languages depending on the requirements, and secondly the programs can be run on separate computers to avoid slowing down the response of the feedback.

2.1 Haptic Design

The role of haptic feedback is to allow the player to feel the result of their playing. An example of this in a traditional instrument would be feeling the skin of a drum vibrating after it has been struck. Benefits of incorporating haptic feedback in an interface include improving the players accuracy [5], and allowing the user to have less reliance on visual and audio feedback.

Haptic feedback can be achieved most simply with vibration, such as using a motor with an off-balance weight attached. This is the method employed in many games controllers and mobile phones. Although simplistic, the vibration motor has been deemed a good starting point before exploring more advanced methods such as force-feedback.

2.2 Audio Design

The problem with designing an electronic instrument is that any sound imaginable can be potentially used, thus giving an overwhelming choice to the designer. One way to simplify the problem is to divide the generation method of electronic sounds into sampled and synthesised. Sampled sounds allow the user to play any sound they like, thus making the instrument more versatile, whereas synthesized sound has the potential for greater expressivity but a more limited range of sounds. It has been decided to use sampled sounds to start with, so that the cubes can be used in a variety of musical situations. However, when wider varieties of PETECUBE's have been produced, investigation into synthesized sound, especially physical modeling synthesis, will be made.

The audio module is important because it acts as the central model of the system. The model holds the instruments current state, which is continuously updated from the output of the PETECUBE's sensors. In the case of using sampled sounds, the parameter being updated is the position of the tape-head within each sample. When the audio model has been updated, the state is then translated into the different output modalities to be fed-back to the user.

2.3 Visual Design

The visual design of traditional non-electronic instruments is generally directed by the mechanical constraints presented when building the instrument, without these constraints it is problematic in deciding how to visualize the virtual instrument. To narrow the design possibilities it has been decided to represent the virtual cube in a relatively realistic manner, so that it is easy to see the link between the virtual cube and the real cube. This virtual cube can then be visually augmented in a manner that would be impossible with a real cube. This augmentation currently takes the form of sound samples being projected perpendicularly from the six faces of the cube, so that the user can see the sample that is being played.

A particular importance of visual feedback is not only in informing the user, but also in displaying the instrument to an audience. Ideally, the visual depiction is clear enough so that the audience can gather what is going on, but at the same time dynamic and exciting enough so that they don't lose interest.

An addition to the visualization of the instrument is anaglyphic 3D-glasses. This allows users (and the audience) to see the virtual cube as three-dimensional. Using anaglyphic 3D glasses is just a temporary stage though, as ultimately the visualization should be located on the cube using augmented reality techniques.

3. PROTOTYPE

A fully functional prototype has been made, as outlined below.

3.1 Prototype Hardware

The prototype uses six light-dependent resistors (LDR's) to sense the users movement. To optimise their sensitivity, LDR's on opposite faces are linked together in a half-bridge, so that the signal generated is the difference between the two sensors readings, rather than the absolute value from each sensor. This has the advantage of negating the ambient light conditions, allowing the cube to be used in nearly all lighting conditions. A less obvious advantage is that by arranging the sensors in the half-bridge, the six sensor outputs are reduced to three, using less ports of the USB i/o.

The i/o hardware is the National Instruments USB-6008. This was chosen because it offers 8 analog inputs, 2 analog outputs and 12 assignable digital i/o lines, potentially allowing two PETECUBE's to be run simultaneously. Another advantage is the scalability when using its device independent C++ library, as a device with more i/o lines, or a higher sampling rate could be used at a later date, with minimal change in the code.

To provide the vibration, two motors with unbalanced loads were appropriated from a Playstation dual-shock controller. Because the loads have different weights, varying intensities of vibration can be achieved. Both motors are placed in the centre of the PETECUBE and secured firmly.

3.2 Software Design

The software is broken up into the three modules of haptics, audio and visualisation as outlined above. The three modules use Open Sound Control [4] to communicate over a network connection, allowing the flexibility of running the programs on separate computers if needed. The use of this is not only to spread the processing load, but can be used in a performance where one laptop could be positioned on stage connected to the USB i/o whilst a second laptop could be positioned at the back of the room and plugged into the mixing desk and projector. In this situation, a wireless network can be used to avoid the use of long cables.

The haptic module's main task is to connect with the USB i/o and to map incoming OSC to voltages out, and incoming voltages to OSC out. It also scales the data, so that the OSC messages are kept within a universal range. The current method of generating the haptic output is to map the amplitude of the sound to the amplitude of the vibration. Although crude, this gives a relatively coherent experience. Currently, both motors are used in the same manner, although improvements are to be made so that the two different intensities of motor are used to greater effect.

The audio module has the central task of not only generating sound, but also to send information on the audio models current state to the haptic and visualisation modules. Currently the audio module is being prototyped in Max/MSP [6], although it is planned to use the Synthesis Tool Kit [7] in C++ for further work. The sound model currently used is a simple one-second long sound file (selected by the user) that can be scrubbed back and forth by the input from the cube. As there is a sample on each side of the cube, six samples can be loaded at any one time. As the playback position within each sample is changed, a ramp is generated between the old and new position to ensure a relatively smooth transition. Because this ramp time is fixed, it becomes possible to control the speed (hence pitch) of the sample by scrubbing faster or slower.

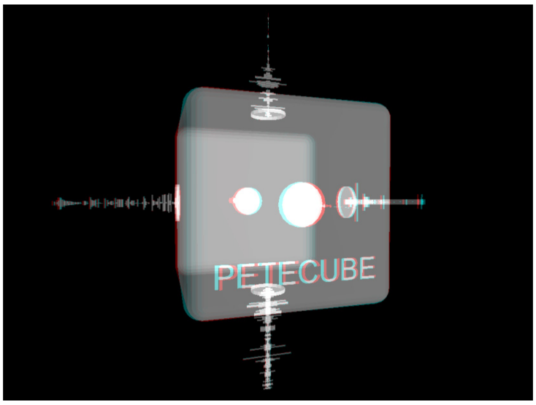


Figure 3. Functioning Prototype Visualisation.

The visualisation module is written in C++ using OpenGL [8]. An anaglyphic library [9] is used to display the model in three dimensions, suitable to be seen with red/cyan glasses (see figure 3 above). The model is designed in a 3D modeling package and then imported into the visualisation module so that an accurate representation is used. To display the waveforms that protrude from the surfaces of the cube it was decided to use a dynamically updating approach. This avoids having to send the entire waveform from the audio module to the visualisation module. To achieve this it is necessary for the audio module to send not only the current position of the 'tape-head' in the sample but also the amplitude of the wave at this point. This allows the visualisation module to build up an image of the waveform after a couple of sweeps of the 'tape-head'. This has two distinct advantages. Firstly, because only the position and amplitude are being sent there is only minimal increase in network traffic, as compared to the alternative of sending a whole waveform over the network. Secondly, because it is a real-time update, if the sample is changed in the audio module, the visualisation will update to reflect this.

3.3 Results

Informal testing has given positive results, especially in the way that people use the visual feedback to determine more accurately what they are playing. The only confusion seems to be in the slight rotation of the virtual cube that accompanies the movement of the play-heads. This misleads people to think that the rotation of the real cube controls rotation of the virtual cube. To remedy this the rotation can be removed, then reinstated when some form of rotational tracking is added to the cube.

The hand-held cube design has been beneficial, as people are not as intimidated as they may be if presented with a traditional instrument. This has proven useful in a gallery situation, where the cubes' design needs to invite people to pick them up and interact with them.

It has been found that the sample-scrubbing model works well for abstract sounds and expression, however it proves to be not particularly suitable for more controlled or measured performance, especially if a pitched sound is required. Rhythmic sounds can be convincingly used, and the gestures involved in rapidly moving towards and away from the cube are a successful method of playing the PETECUBE. Another method of playing that has been found is using a strong uni-directional light-source (such as a desk-lamp) and rotating the cube without deliberately covering the LDR's with the hand. This allows rotational gestures to be used for easily repeatable sound generation. For finer control of samples, a method of cupping the hands over opposing faces of the cube, and using the palms to block out light allows subtle movements to be captured.

A current problem lies in the haptic feedback. The mapping between the audio model and the two motors is underdeveloped compared to the audio and visual feedback, resulting in slight incoherency between the sound and the vibrations. Although the mapping is being developed further, it is felt by the author that a more sophisticated haptic system needs to be explored in future PETECUBE's to catch up with the development of the audio and visual feedback.

4. FURTHER WORK

Now that the basic system is set up and functional, it is possible to continue research into further variations of PETECUBE (for current progress see [10]). The aim is to create a series of cubes, each with different combinations of sensors, actuators and control models, which can be used as the basis for investigation into feedback in musical interfaces. Examples of potential cubes are listed below:

- Record-Cube. A cube that can record and play back data from all of its active modalities. Can this be used with a series of cubes for a form of multimodal sequencing?
- Twist-Cubes. Two cubes joined by a motor and encoder. This would allow more advanced force feedback in a rotational manner.
- Shock-Cube. Can unpleasant feedback (such as electric shocks) be used in a multi-modal interface?
- Tele-Cube. Cubes that are connected at a distance over a network (or the internet). Is directing feedback from one cube to another remote cube useful in collaborative music making? Can feedback from two remote cubes be simultaneously displayed in a single cube?
- Tracking-Cube. The use of an inertial or gyroscopic tracking system would allow the visualisation to accurately follow the movement of the cubes, while also adding an extra input modality.
- Push-Pull-Cubes. Two cubes connected by a linear damper (such as a Magneto-Rheological Fluid Damper). This would allow the user to use the cubes in an accordion-like manner, with controllable linear damping. Effects could be explored such as making it harder to 'push' through a louder sample.

Another aspect to be explored in further work is the embodiment of all the feedback modalities within the cube. The haptic feedback is already localised within the cube, however the audio and visual modalities rely upon speakers and monitors respectively. The audio speakers are likely to be the easiest to locate within the cube, whereas the visual augmentation will require advanced Augmented Reality techniques.

Other intended further work will involve standardising the OSC message system so that the software modules will become interchangeable. This will then lead to creating templates for each module in various languages (C++, Java, MaxMSP) so that it is a straightforward task for other people to develop their own modules. Due to the relatively cheap parts and very simple design, it is then hoped that people will experiment with building their own PETECUBE to accelerate the research in multimodal feedback.

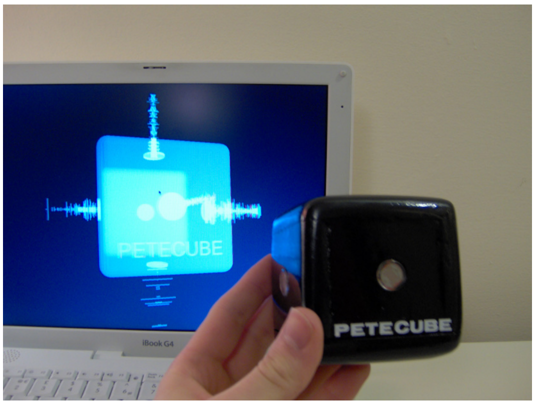


Figure 4. The PETECUBE in use.

5. CONCLUSION

This paper has discussed the definition of a multimodal feedback interface, discussed design concerns in its realisation, given an account of the current state of the PETECUBE, and outlined possible further work. As the project progresses, it is hoped that the PETECUBE will become the basis for many experiments into multimodal-feedback instruments.

6. ACKNOWLEDGMENTS

The author would like to thank the NIME reviewers for their feedback.

7. REFERENCES

[1] Schomaker, L., Nijstmans, J., Camurri, A. Lavagetto, F., et al., *A Taxonomy of Multimodal Interaction in the Human Information Processing System*, Esprit Basic Research Action 8579, Multimodal Integration for Advanced Multimedia Interfaces (MIAMI) 1995.

[2] Collins English Dictionary. http://www.collins.co.uk/wordexchange/, 2006

[3] Rodet, X. Gosselin, F. Mobuchon, P. *Study of haptic and visual interaction for sound and music control in the PHASE project.* In Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Vancouver, BC, Canada.

[4] Wright, M. Freed, A. Momeni, A. *OpenSound Control: State of the art 2003.* In Proceedings of the 2003 conference on New Interfaces for Musical Expression (NIME 03), Montreal, Canada.

[5] O'Modhrain, S. *Playing by Feel: Incorporating Haptic Feedback into Computer–Based Musical Instruments.* PhD thesis, Stanford University, Palo Alto, California, November 2000.

[6] Max/MSP, http://www.cycling74.com/, 2006

[7] Scavone, G. Cook, P. *RTMIDI, RTAudio and a Synthesis Toolkit Update* In Proceedings of the 2005 International Computer Music Conference, Barcelona, Spain.

[8] OpenGL.org. http://www.opengl.org/, 2006

[9] Bourke, P. *Creating Anaglyphs Using OpenGL.* http://astronomy.swin.edu.au/~pbourke/opengl/redblue/, 2006

[10] PETECUBE, http://www.petecube.com/, 2006

The G-Spring Controller

Denis Lebel
Computational Acoustic Modeling Laboratory
McGill University - Music Technology Area
Montreal, QC, Canada
denis.lebel@mail.mcgill.ca

Joseph Malloch
Input Devices and Music Interaction Laboratory
McGill University - Music Technology Area
Montreal, QC, Canada
joseph.malloch@mail.mcgill.ca

ABSTRACT

This paper presents a novel digital musical instrument using an interface constructed from a large spring that offers natural kinesthetic feedback through its inherent stiffness. The design of the first prototype is described and discussed in relation to the notion of effort in musical performance. Audio feedback and distortion are introduced as a possible way to extend the musical limits of the simple interface.

Keywords

Digital musical instrument, kinesthetic feedback

1. INTRODUCTION

The concept of effort in musical performance is often referred to in the context of digital musical instrument (DMI) design. Tanaka, for example, observes that sensor-based instruments often lack the sophisticated inherent feedback feature that demands physical effort in manipulation [14], and asserts that this difference holds back the enjoyment and acceptance of DMIs. Furthermore, although the design of these allows for a reduction in required performer effort, many consider that without this effort an instrument cannot be performed expressively [12, 10].

The G-Spring design is inspired by this notion of effort. Using a physical interface with inherent kinesthetic feedback, it may be possible to create a DMI that feels more like a traditional acoustic instrument, which may in return open doors to higher expressivity for the performer [2]. The instrument – despite its digital nature – will hopefully appeal more to the performer and audience as a response to the feedback manifested through the performer effort [14].

In [16], Wessel and Wright discuss the “low entry fee with no ceiling on virtuosity” design approach. They mention that traditional acoustic instruments are typically not easy to learn, but provide for a high degree of musicality. When talking about the simplicity of some DMI interfaces, they point out that one quickly surpass the interface by discovering its limits, which highly contrasts with traditional instruments.

Along the same lines, a way to reach a balance between an easy-to-use interface and continuous musical evolution is explored with the G-Spring. It is an attempt to design an instrument with a simple interface that produces sound easily, but afford musical complexity through additional and more refined controls. Through practice, these qualities can be used by a performer to create unique sounds and push the musical limits of the basic instrument.

The remainder of this paper describes the design of the G-Spring, shown in Figure 1, and explains how effort is introduced by the choice of material as well as how possibilities for continuous musical evolution are investigated through the use of performer-controlled audio feedback and built-in distortion. The controller design is briefly evaluated and discussed in regards to the notion of effort. Finally, some directions for future work are outlined.

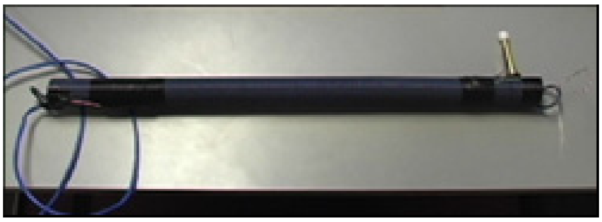


Figure 1: The G-Spring controller.

2. RELATED WORK

The G-Spring is superficially similar to the Sonic Banana [13]. Both controllers measure bending and map it to numerous sound synthesis parameters. In the Sonic Banana, four bend sensors are fixed linearly inside a flexible rubber tube to measure its bending. The interface differs in that little physical force is required by the performer to bend the Sonic Banana due to its high flexibility, whereas the G-Spring was designed with effort and kinesthetic feedback in mind.

Harmonic Driving, a controller part of the Brain Opera, is an interface that does provide kinesthetic feedback to the user. It uses a large open-coil spring as an interface for musical steering tasks, using capacitive sensing to determine the relative displacement between two adjacent coils [11].

Past DMI designs that use light-sensing include the Photosonic Disk [1], the Circular Optical Object Locator [7], and the Light Pipes [17].

3. DESIGN

3.1 Physical Interface

The G-Spring interface is meant to be bent and thus, in order to provide decent kinesthetic feedback, the material used for its construction needed to be fairly deformable while requiring a reasonable force to be bent. Furthermore, it had to return automatically to its equilibrium position when pressure is released from it. Although some rubber substances might have been suitable, it was judged that the simplest solution was to use a bendable spring.

The spring used, which is shown in Figure 1, is a large and heavy close-coil extension spring commonly found in garage door mechanisms. It measures 63.5 cm unstrained, has a diameter of 3.5 cm, and weighs 1.85kg, which makes it particularly comfortable when held in average-sized hands. Its stiffness and flexibility are moderate in that it can be bent such that both extremities touch each other without demanding excessive force.

A spring door stop was fixed at one of the extremities of the spring, but perpendicular to it (Figure 1, bottom). The door stop is meant to bend towards the center of the large spring.

3.2 Sensing Methods

Although bend sensors might initially seem like a more logical choice to measure the bending of the large spring, this approach is more problematic than useful. The spring coils are very compact and will quickly crush anything inserted in between them, which makes it particularly hard to fix sensors on the spring without destroying them. The radius of curvature of the spring (when bent) is quite large due to its dimensions and close-coil characteristic. A last observation to be made is that in order to allow free bending of the spring, attached sensors must be flexible in all directions. These requirements make standard bend sensors less than ideal. The density of the coils also prevents the use of capacitive sensing as in the Harmonic Driving controller[1].

In practice, bending the spring causes slight gaps to form between adjacent coils, allowing ambient light to enter the interior of the spring. Thus, a light-dependent resistor (LDR) was found more suitable and durable to measure the bending of the large spring, since the sensor and wiring do not actually need to be in contact with it. Fixed inside the spring (Figure 2, top), and positioned a little off-center to align with its most comfortable flexion point, the LDR provides an inexpensive way to determine the bending without restraining it.

In order to provide additional control on the sound synthesis, a force-sensitive resistor (FSR) was placed near one end of the spring. Electrical tape was used as an insulating layer to protect the external sensor from being damaged by the spring coils when it bends (Figure 2, middle).

A bend sensor was fixed against the spring door stop at the opposite end of the spring. It was fixed this way (Figure 2, bottom) to get a more accurate measurement of its bending by reducing the radius of curvature. Introducing this angle in the bend sensor causes a wider range of resistances from the sensor, thus differentiating small variations in the bending more easily [6].

[1]In the Harmonic Driving, the spring coils were distant from each other.

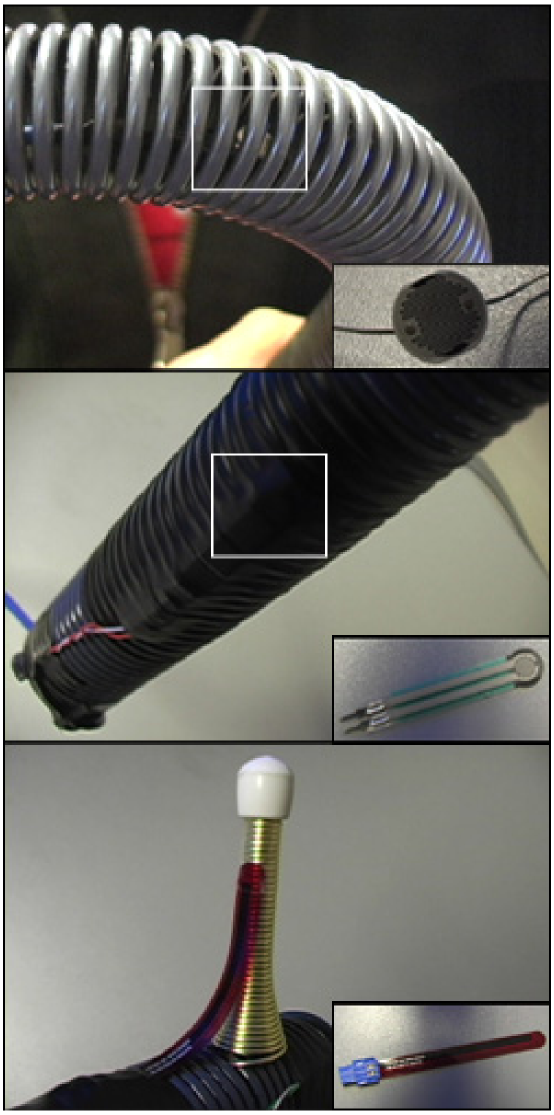


Figure 2: Sensors used in the G-Spring controller. Top: LDR inside the spring. Middle: FSR taped near one end of the spring. Bottom: Bend sensor fixed against a door stop at the other end of the spring.

3.3 Interface-Computer Communication

The communication with the computer is performed through USB, using a homemade AVR-HID device offering six analog input channels [9]. The AVR-HID is inexpensive and transforms the controller into a plug-and-play system that works on several systems including Windows, Mac OS X, and Linux.

3.4 Synthesis

The sound synthesis was implemented in Max/MSP and uses a source-filter model. The source is a bandlimited pulse wave rendered at low frequency (60Hz), which is fed into a bandpass filter with controllable center frequency, input amplitude and quality factor[2]. The resulting signal is processed by a comb filter, whose delay length, feedback coefficients, and feedforward coefficients are modulated.

[2]Quality factor is defined as the center frequency of the filter divided by its bandwidth.

3.5 Mapping

The output of the LDR is mapped to the center frequency of the filter. The center frequency is increased as more light hits the resistor. The control signal is mapped exponentially, which allows for a finer control over the lower frequencies and a coarser control of the higher frequencies[3]. The LDR signal is also mapped linearly to the quality factor of the filter.

A one-to-many mapping is used for the FSR [8]. The sensor output is mapped exponentially to the bandpass filter input gain and fed into a 2-second ramp, which allows for very smooth and continuous control of the sound. The global output gain of the synthesizer is also controlled by comparing the FSR value against a threshold level to cut off the sound as soon as the pressure is released.

The bend sensor is used to control the parameters of the comb filter. First, it controls the modulation frequency of the varying delay line length. It also controls the feedback and feedforward coefficients, which are modulated as well. In addition, a cutoff value is used to determine whether or not the door stop is being bent.

4. MANIPULATION

With the sensors in their current positions, the instrument can be manipulated with each hand holding one extremity of the spring. Bending the spring changes the range of frequencies that are let through by the bandpass filter. In general, the more the spring is bent, the higher the frequency range becomes, since more light reaches the LDR. The sound level is controlled by applying pressure on the FSR with the thumb of one hand. The higher the pressure is, the louder it is. The comb filter parameters are modified with the index of the other hand by bending the door stop towards the center of the spring.

It is perhaps more comfortable to stabilize one end on a stationary surface, since it makes it easier to smoothly bend the spring and helps counter its stiffness. Also, by fixing one end of the spring to a solid, stationary surface or stand, it could be bent more easily in any direction using both hands. This would also free the hands to directly modulate the amount of light detected by the LDR, by waving or placing them over the flexion point.

5. RESULTS AND DISCUSSION

5.1 Design Evaluation

The substantial weight of the spring enhances the overall feel of the controller: the performer has the impression of holding a tangible and robust instrument. The spring stiffness, however, makes it uncomfortable when manipulated over an extended period of time, since it requires that force be constantly applied on it in order to maintain a bent position. This also prevents any further addition of finger controls due to human biomechanics limitations. Thus, an instrument requiring a constant physical effort by the performer may not necessarily be desirable. In the G-Spring's case, a different mapping strategy, in which the bending of the spring controls some modulation, could be potentially more suitable.

The LDR works well at measuring the spring bending, although using only one LDR does not provide accurate measurements since the amount of light that reaches the sensor is not necessarily proportional to the bending. For example, if a part of the spring distant from the sensor is bent, or if the sensor is pointed away from the coil gaps, less light will illuminate the sensor.

The bend sensor provides a very fine control that is ideal for feedback parameters. The use of a cutoff value greatly enhances the capabilities of the instrument, allowing the performer to keep a certain set of feedback parameters during a performance by simply releasing the pressure on the door stop, which effectively stops modulation.

The ramp applied to the FSR control signal was judged to be the most successful mapping strategy of the instrument as it provides a very smooth and continuous control over the sound level and makes the instrument easier to play. Moreover, the cutoff applied to the output signal allows for the sound to shut off as soon as the pressure is completely released from the FSR, which is a very important feature to have in a musical system [16].

5.2 Continuous Musical Evolution

Even though the G-Spring interface is simple and offers few degrees of freedom, this does not necessarily mean the instrument can be mastered in a few minutes, or that it is sonically limited. The performer-controlled audio feedback allows the creation of a fairly wide spectrum of sounds, generated through precise manipulations of the instrument. By simultaneously modifying the center frequency of the bandpass filter (spring bending) and the feedback parameters (door stop bending), numerous sounds are possible. Furthermore, while applying a constant force on the spring, adequate control of the feedback parameters is challenging and requires practice before expert results can be created.

The calculation of the quality factor of the filter introduces distortion at certain frequencies. When sweeping the frequency range of the filter, the sound level varies from a moderate level at low frequencies, to a high level near 500Hz and then decreases to a very low level at high frequencies. This occurs due to the varying bandwidth of the filter. The distortion created in the mid-range frequencies acts as an additional parameter that can be used by the performer through control of the center frequency and amplitude of the bandpass filter. This can be further refined by simultaneously playing with the feedback parameters.

This suggests that the sonic possibilities of DMIs can be extended through clever synthesis and mapping strategies even though the controller interface is rather simple. In the G-Spring, the feedback and distortion definitely provide some complexity to the sound output. Unfortunately, not enough time was spent experimenting with these two parameters to clearly determine whether they really provide a way for continuous musical evolution, but it remains nonetheless an idea worth investigating.

5.3 Performance

Some subjects, mostly non-musicians, were asked to watch a short video[4] of a performance involving the G-Spring controller, without having encountered the instrument before. An interesting comment was made by the subjects regarding the lack of movement during the performance, saying that they felt less engaged by it. Consequently, what these subjects expressed was that the amplitude of the motion induced by the instrumental gestures, as defined in [3], was minimal and sometimes left the lis-

[3]Note that the instrument was originally intended to focus on low frequencies.

[4]Available on the project website: http://www.music.mcgill.ca/~lebel/gspring/

tener/viewer without clear visual cues as to whether or not the performer was really involved in the sound modification process. Due to the inherent kinesthetic feedback of the spring and the mapping strategies implemented, mostly continuous, very precise and low-amplitude movements are used to modify the sound parameters. In other words, when bending the spring, a force needs to be constantly maintained, which limits the apparent motion of the performer. This suggests that although effort may help convey visual cues, the apparent motion of the performer's instrumental gestures also plays an important role in the understanding and appreciation of the instrument and the performance by the audience [5].

People who tried the instrument enjoyed the fact that it produces sound instantly, which supports the "Instant music, subtlety later" principle stated by Cook in [4]. The sound modification controls are also easily grasped due to the visualization of the filter parameters through a graphical filter object in Max/MSP. This suggests that the design of DMI interfaces do not necessarily need to be extremely simple, but rather can be used with visualization to ease the learning curve of the performer. Once a good intuition of the controller is gained, the instrument could then be used without visualization.

6. FUTURE WORK

Since increasing the ease-of-use of DMI interfaces is not necessarily desired by performers [10], and visual feedback seems to make the instrument control more intuitive, it would be interesting to investigate the use of control parameter visualization with complex interfaces. Also, a more in-depth and quantified study of kinesthetic feedback in regards to the ergonomics of interfaces would be useful to better integrate the notion of effort in the design stage of DMIs.

7. CONCLUSION

A new digital musical instrument was presented featuring inherent kinesthetic feedback, which requires physical effort by the performer to be played. The interface is intuitive, thanks to the visualization of the control parameters, while allowing for more complex and subtle modifications of the sound through user-controlled audio feedback and built-in distortion. Observations were made in regards to the importance of visual feedback for both the audience and the performer.

Acknowledgments

The authors would like to thank Mark Marshall for his great help with the AVR-HID USB interface.[5] Many thanks go to everyone in the Music Technology Area of McGill University for countless suggestions and very interesting discussions. A special thank you is directed to Marcelo Wanderley, whose seminar on Gestural Control of Sound Synthesis initiated the G-Spring idea.

8. REFERENCES

[1] D. Arfib and J. Dudon. Photosonic disk: Interactions between graphic research and gestural controls. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 9–17. IRCAM – Centre Pompidou, Paris, 2000.

[2] W. Buxton and R. Baeker. The haptic channel. In R. Baeker and W. Buxton, editors, *Readings in Human-Computer Interaction*, pages 357–365. Morgan Kaufmann, 1987.

[3] C. Cadoz and M. Wanderley. Gesture-music. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 71–93. IRCAM – Centre Pompidou, Paris, 2000.

[4] P. Cook. Principles for designing computer music controllers. In *Proceedings of the New Interfaces for Musical Expression Workshop*, 2001.

[5] J. Davidson. Visual perception of performance manner in the movements of solo musicians. *Psychology of Music*, 21:103–113, 1993.

[6] Flexpoint Sensor Systems. Bend sensor technology mechanical application design guide. `http://www.flexpoint.com/mechanicaldesignguide.pdf`, 1997.

[7] T. Hankins, D. Merrill, and J. Robert. Circular optical object locator. In *Proceedings of International Conference on New Interfaces for Musical Expression*, pages 163–164, 2002.

[8] A. Hunt, M. Wanderley, and R. Kirk. Towards a model for instrumental mapping in expert musical interaction. In *Proceedings of the International Computer Music Conference*, pages 209–212, 2000.

[9] M. Marshall. Building a usb sensor interface using the atmel atmega16 and the human interface device standard. `http://www.music.mcgill.ca/~marshall/projects/avr-hid/`, 2006.

[10] A. Mulder. Toward a choice of gestural constraints for instrumental performers. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 315–335. IRCAM – Centre Pompidou, Paris, 2000.

[11] J. Paradiso. The brain opera technology: New instruments and gestural sensors for musical interaction and performance. *Journal of New Music Research*, 28(2):130–149, 1999.

[12] J. Ryan. Some remarks on musical instrument design at steim. *Contemporary Music Review*, 6(1):3–17, 1991.

[13] E. Singer. Sonic banana: A novel bend-sensor-based midi controller. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 220–221, 2003.

[14] A. Tanaka. Musical performance practice on sensor-based instruments. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 389–405. IRCAM – Centre Pompidou, Paris, 2000.

[15] M. Wanderley, J. Malloch, D. Birnbaum, E. Sinyor, and J. Boissinot. Sensorwiki.org: A collaborative resource on transducers for researchers and interface designers. In *Proceedings of International Conference on New Interfaces for Musical Expression*, 2006.

[16] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

[17] S. Won, H. Chan, and J. Liu. Light pipes: A light controlled midi instrument. In *Proceedings of International Conference on New Interfaces for Musical Expression*, pages 209–210, 2004.

[5] Additional information about his work on the AVR-HID USB interface can be found on his project website: http://www.music.mcgill.ca/~marshall/projects/avr-hid/

Orbophone: a new interface for radiating sound and image

Damien Lock
Faculty of Creative Arts
University of Wollongong
Wollongong NSW Australia
61 2 4221 3584
damien@coscilia.com

Greg Schiemer
Faculty of Creative Arts
University of Wollongong
Wollongong NSW Australia
61 2 4221 3584
schiemer@uow.edu.au

ABSTRACT
The Orbophone is a new interface that radiates rather than projects sound and image. It provides a cohesive platform for audio and visual presentation in situations where both media are transmitted from the same location and localization in both media is perceptually correlated. This paper discusses the advantages of radiation over conventional sound and image projection for certain kinds of interactive public multimedia exhibits and describes the artistic motivation for its development against a historical backdrop of sound systems used in public spaces. One exhibit using the Orbophone is described in detail together with description and critique of the prototype, discussing aspects of its design and construction. The paper concludes with an outline of the Orbophone version 2.

Keywords
Immersive Sound; Multi-channel Sound; Loud-speaker Array; Multimedia; Streaming Media; Real-Time Media Performance; Sound Installation.

1. INTRODUCTION
The concept was devised and realised by Damien Lock for a collaborative exhibit titled Shades Of Light presented at the "Don't let sleeping androids lie" exhibition, Sculpture Square, 155 Middle Road, Singapore, from June 9th to August 28th 2005, as part of the Singapore Arts Festival 2005. The work presents a selection of readings from Alvin Pang's poetry collection entitled "City of Rain" [16]. These are read by different people to document interpretations that represent a collective response to the text. Themes in the text are developed through accompanying animated images, video, and soundtrack. These elements were sequenced algorithmically in realtime.

1.1 Orbophone - radiation versus projection
Sound radiation refers to a loudspeaker array in which sound emanates from a speaker source or speaker sources positioned in a central location in a room. It differs from standard forms of sound spatialisation where loudspeakers project from the periphery of a room to create an illusion of space. With sound radiation each speaker source within the array emanates from a discrete physical location, in much the same way an acoustic sound radiates from a discrete source. The position of the array within the room determines the overall mix of direct, reflected and reverberant sound experienced by a listener.

Sound radiation offers advantages over the standard cinematic approach to sound spatialisation where surround speaker arrays create an artificial soundfield that surrounds the audience [9]. An artificial soundfield requires careful calibration and control of the acoustics in the room it occupies, and its optimum reception is limited to a region in the center of the speaker array [8]. With sound radiation the speaker array positions sound sources in a centralised location allowing the sound to take on the acoustic characteristics of the room.

Surround speaker arrays operate independently of the screen image, posing limitations on sound spatialisation gestures. Radiating both sound and image from a three dimensional object provides a more unified relationship between the two mediums that naturally interacts with the architectural space the system occupies.

1.2 Sound radiation – historical perspective
Instrument designers have long been producing instruments which either acoustically or electronically radiate amplified sound via speakers which form part of the instrument's body. Luigi Russolo's Intonorumori (noise intoners) constructed between 1913 and 1921 [5] were each fitted with a large metal speaker for this purpose. Performances in the 1920's by Leon Theremin's early electronic musical instrument ensemble [11] is perhaps one of the earliest examples of loudspeakers used for performing music. However, experimental four channel performances by Pierre Schaeffer in the Théâtre de l'Empire in Paris in 1951 signaled the arrival of the use of speaker arrays for spatial sound projection. Whereas Theremin's instruments were each limited to a single speaker, Schaeffer panned signals between the 4 speaker channel. In 1958, Edgar Varese and Iannis Xenakis performed pieces composed for a 425-speaker array installed in the Philips Pavilion at the Brussels Exposition. The first large scale spherical arrangement of speakers was premiered in 1970 at Osaka World's Fair - an arrangement comprising of 50 speakers designed for the performance of Stockhausen works. The Japanese Steel Pavilion in the same year presented Xenakis' Hibiki-hana-ma used "800 speakers situated around the audience, overhead and under the seats" [21]. The first permanent installation for acousmatic works was the Gmebaphone' of the Groupe de Musique Expérimentale de Bourges in 1973 [4].

In musical performances, when a performer's instrument is amplified and projected via a soundfield, the sound becomes physically dislocated from the performer. This has posed problems in both performance and the reception of the sound as "the loudspeaker disembodies the actual source" [1]. Performers experience problems maintaining balance in the mix of acoustic and electronic instruments. Audiences also experience difficulty observing the relationship between the physical gestures made by the performers on stage and the resulting dislocated mix of sounds.

Outward bound spherical speaker arrays have a role to play in addressing these problems. Because sound radiation achieves a degree of localisation and naturally dispersed amplification for live signals, they allow electronic instruments to be used in mixed ensembles. Localized in this sense refers to the proximity of:

- the player to the instrument
- the instrument/performer to the other ensemble members
- the location of the performer/instrument in the architectural space as perceived by the audience.

Spherical speakers are now commercially available [14]. Acoustic instruments have been shown to have unique frequency dependent radiation patterns [2]. By emulating this characteristic of musical instruments in loudspeaker design it seems possible to integrate sound played on electronic instruments in concert performances with acoustic ensembles. Such an approach has already been adopted by Trueman, Bahn and Cook [19].

2. Orbophone – mach 1

2.1 Structure

Panning rapidly across the surfaces of a three dimensional object seems an appropriate gesture for audio spatialisation. Such a gesture may describe radiation patterns that are omni-directional or highly directional. The icosahedron form - one of five polyhedrons defined by their congruent regular polygon surfaces and polyhedral angles - was chosen as an enclosure as it can produce a good omnidirectional radiation pattern of sound energy [15]. The dodecahedron with twelve sides is the particular variant of the icosahedron used. Though it roughly approximates a spherical surface, it is economical and capable of delivering a variety of soundfields via individually driven speakers symmetrically mounted on ten of its faces. The top and bottom faces are reserved for image projection and display.

Figure 1 shows the dodecahedral structure of the Orbophone. A loudspeaker is mounted in ten of the twelve pentagonal faces.

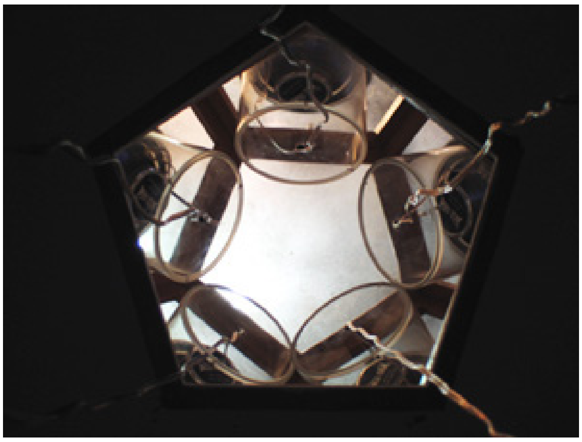


Figure 2 shows five of the ten loudspeakers each positioned in the centre of one of the pentagons on the side of the structure. An image projector is mounted in the plinth and projects through the pentagon at the base of the structure onto the pentagon at the top.

2.2 Sound

Two modes of sound radiation are possible with the Orbophone: isotropic and anisotropic. Isotropic is when the magnitude of the sound is uniform in all directions (omnidirectional). Anistropic sound radiation occurs when the magnitude of the sound varies in direction. The radiation modes may be described as:

- a monophonic sound through all speakers (isotropic)
- sound through discrete speakers (anisotropic)
- sound through groups of 2 – 10 speakers (anisotropic)
- sound moving through groups of 2 – 10 speakers (anisotropic)

This configuration of independent directional speakers can produce a variety of spatialisation gestures including circular panning around the asimuth, as well as elevation panning.

2.3 Image projection

Images are radiated via a high resolution projector connected to a computer's video card. The projector is positioned underneath the Orbophone and points upwards through its base. Images reflect off and through translucent surfaces on all faces of the dodecahedron except the bottom. The top face has no speaker so has a clear display area where images can be viewed distinctly. The translucent surface also allows light to disperse and influence the visual ambience of the space around the Orbophone. Image is also projected onto the ten pentagonal faces of the dodecahedron. Here, image is limited to the surface area surrounding the loudspeaker on each face.

Each pentagonal face is made of transparent acrylic material. This was treated to make it translucent in order to catch images projected from the rear projection system. This was achieved by thinly and evenly coating each surface using 3M spray mount artist's adhesive. Surfaces were then coated with bleached mulberry fibre to further enhance light absorption.

Fig. 3. Construction: close up of bleached mulberry fibres which enhance light absorption properties of the surface

The Orbophone projects light in a variety of ways:

- at close proximity (up to 3 metres) distinct images can be viewed on any surface
- further away (from 3-25 metres) patterns and colours appear to radiate from the Orbophone
- light is dispersed through translucent surfaces onto surrounding walls and ceiling

2.4 Media design

Composing for an instrument that radiates sound offers unique opportunities for musical expression like assigning directivities to musical streams to create spatial counterpoints. [13]. Media designed for the Orbophone was algorithmically organized and comprised of both pre-recorded and synthesized elements. Parameter mapping between sound and image systems heightened the relationship between these mediums. Abstract images worked well as they were able to be prepared without the need to accommodate geometrical aspects of the screen surface. Supercollider [12] was used to sequence the sound events and to communicate event information via Open Sound Control [20] with the image projection engine running on Max/MSP/Jitter [3]. Improvements for Orbophone media design include:

- Integration of non-abstract visual material that is oriented in a way so it may be projected as a coherent image on the 3-dimensional projection surface
- dynamic organization of the material so that it is partial to viewer interaction and the influence of control parameters originating from user interference.
- Automatic calibration of the audio/light levels to suit a variety of installation scenarios.

3. Orbophone – mach 2

The Orbophone was deployed as an installation in an art gallery situated on the Singapore tourism roadmap, and open to the general public. Audience reaction suggested several improvements. The image projection area needed to be increased as structural beams and opaque speakers tended to obstruct the visual content. The content on display could be more effective if the audience could interact with it rather instead of watching something controlled by an algorithm.

3.1 Transparent sound interface

The Orbophone offers new possibilities for collaborative interaction. The quality of human interaction in any display system is increasingly recognized as a key element in relating to the content on display [7]. Its spherical shape and moderate size allows groups of people to assemble around it, offering a comfortable and practical setting for discussion about the projected material. The addition of an interactive interface suitable for group interaction would further maximize the flexibility of the system to present information, be it artistic or informative.

The planned user interface will detect proximity of a body to each surface of the dodecahedron via an array of non-contact sensors. This allows hand gestures made above the surface of the Orbophone to influence or interfere with the projected media content. These would suit a wide range of intuitive control gestures for the audio visual content on display. Various composers and performers who work with sound spatialisation report difficulties using 3-dimensional gestures to control sound trajectories [17]. An interface using an array of proximity sensors will address that problem. A hand hovering over a particular surface could direct sound to that speaker surface. This provides intuitive direct control over the spatial position of the sound. The gesture would be further reinforced if the image display system tracked the relocated sound.

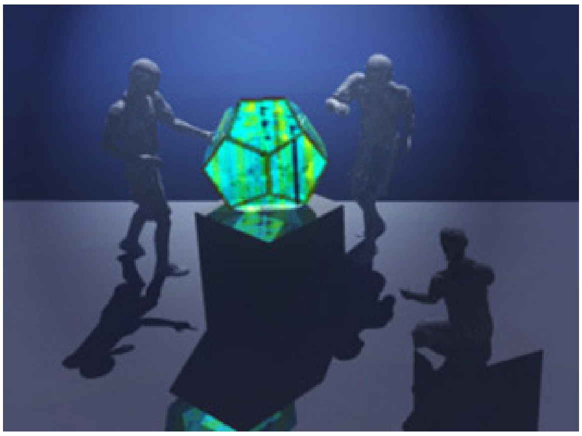


Fig4: Mock up of group interaction with Orbophone. Thin film speaker technology enables spherical sound projection together with uninterrupted image projection

3.2 Uninterrupted sound and image radiation

One obstacle for the full realization of this display system is to project uninterrupted images from all faces except the bottom face where the projector is positioned. The author has identified a thin film speaker technology built from a durable polarised polyvinylidene fluoride, material which has piezoelectric properties [10]. It is comparable to traditional speaker technologies in both performance and durability [18]. It can be translucent in color, and can be made in many sizes and shapes.

3.3 Interface development

The interface is an array of E-Field sensors, made using the Freescale Semiconductor MC33794 E-Field IC. External electrodes allow this IC to detect objects in an electromagnetic field and register their movement in the field as a change in voltage [6]. Electrodes on each surface create a constant interference field around the Orbophone. Communication with the MC33794 will use a microcontroller SPI port. This allows for expanded configurations of multiple sensors where more than 9 electrodes are required. Design challenges include calibration of range and sensitivity, minimizing interference, and eliminating crosstalk between the sensors.

4. Conclusion

The Orbophone instrument described above marries sound and image radiation into an adaptable instrument suitable for composing works that investigate the relationship between the two mediums. Future realizations of the instrument will overcome some of it's limitations by enabling the uninterrupted radiation of audio/visual content controllable via an intuitive interactive interface.

5. ACKNOWLEDGMENTS

This project was supported by the National Arts Council of Singapore and the Sonic Arts Research Network, University of Wollongong. The authors would also like to acknowledge technical support of MJ Labs (81A-3 Geylang Road, 02-03, 389199) Singapore.

6. REFERENCES

[1] Boulez, P. L'acoustique et la musique contemporaine. *11e ICA*, Vol.8, Paris, 1983, 213-216.

[2] Caussé, R., Bresciani, J. and Warusfel, O. Radiation of musical instruments and control of reproduction with loudspeakers. In *Proceedings of the International Symposium on Musical Acoustics*, Tokyo, Japan, 1992. http://mediatheque.ircam.fr/articles/textes/Causse92a/

[3] Cycling '74, *Max/MSP & Jitter*. Available from http://www.cycling74.com

[4] Emmerson, S. and Smalley, D. Electro-acoustic music. *Grove Music Online* ed. L. Macy (Accessed 10th August, 2005), http://www.grovemusic.com

[5] Flora, D. Russolo, Luigi. *Grove Music Online* ed. L. Macy (Accessed 10th January, 2006), http://www.grovemusic.com

[6] Freescale Semiconductor *Touch Panel Applications using the MC33794 E-Field IC Rev. 1.0*, Freescale Semiconductor, 2005 http://www.freescale.com/files/analog/doc/app_note/AN1985.pdf

[7] Hunt, A. and Herman, T., The Importance of Interaction in Sonification. In *Proceedings of The International Conference on Auditory Display* (Eds, Barrass, S. and Vickers, P.), Sydney, Australia, 2004, ICAD04-1.

[8] International Telecommunication Union, ITU-R Recommendation BS.1116: *Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems*. Geneva, Switzerland, 1994.

[9] International Telecommunication Union, ITU-R Recommendation *BS.775-1: Multichannel Stereophonic Sound System with and without Accompanying Picture*. Geneva, Switzerland, 1992-1994.

[10] Lee, C. S., Kim, J. Y., Lee, D. E., Joo, J., Han, S., Beag, Y. W. and Koh, S. K. An approach to durable poly(vinylidene fluoride) thin film loudspeaker. In *Journal of Materials Research*, Materials Research Society, Vol. 18, Issue 12, Warrendale, PA, Dec. 2003, 2905-2911.

[11] Martin, S. *Theremin, An Electronic Odyssey*, MGM Home Entertainment Inc., Santa Monica, CA, 1993.

[12] McCartney, J., *Supercollider*. Available from http://www.audiosynth.com

[13] Misdariis, N., Nicolas, F., Warusfel, O. , Caussé, R. Radiation control on a multi-loudspeakers device : La Timée, ICMC, La Havane, 2001.

[14] Noselli, G. *Diffusore sferico a direttività variabile*. Italian Patent, Certificate No. IT252050, 3rd May 2004.

[15] Noselli, G., Simeoni, F., Vezzoli, F., Beretta, G., Pedersini, G., Piana, E. *Study, construction and analysis of a variable directivity omnnidirectional sound source*. Thesis, University of Brescia, Department of Mechanical Engineering, & Outline Professional Audio, Brescia, Italy, 2000. http://www.outline.it/Globe-Source.htm

[16] Pang, A. *City of Rain*. Ethos, Singapore, 2003.

[17] Sheridan, J., Sood, G., Jacob, T., Gardner, H. and Barrass, S. Soundstudio4D - A VR interface for gestural composition of spatial soundscapes. In *Proceedings of The International Conference on Auditory Display 2004* (Ed. Barass, S. and Vickers, P.), Sydney, Australia, ICAD04 G. Sood et al: 1-7.

[18] Sung-jin, K. Paper-Thin Speakers to Revolutionize Industry. In *Korean Times*, Seoul, South Korea, May 2nd, 2005, 12.

[19] Trueman, D., Bahn, C., and Cook, P. Alternative Voices for Electronic Sound. In *Proceedings of The International Computer Music Conference 2000*, Berlin, Germany. http://music.columbia.edu/~dan/alt_voices/alt_voices.comp.pdf

[20] Wright, M., *Open Sound Control*. Available from http://www.cnmat.berkeley.edu/OpenSoundControl/

[21] Zvonar, R. A *History of Spatial Music Historical Antecedents: From Renaissance antiphony to strings in the wings* (Accessed 15th January, 2006), http://cec.concordia.ca/econtact/Multichannel/spatial_music.html

the gluion
advantages of an FPGA-based sensor interface

Sukandar Kartadinata
glui
Hagenauerstr. 6
10435 Berlin, Germany
sk@glui.de

ABSTRACT
The gluion is a sensor interface that was designed to overcome some of the limitations of more traditional designs based on microcontrollers, which only provide a small, fixed number of digital modules such as counters and serial interfaces. These are often required to handle sensors where the physical parameter cannot easily be converted into a voltage. Other sensors are packed into modules that include converters and communicate via SPI or I2C. Finallly, many designs require output capabilities beyond simple on/off.

The gluion approaches these challenges thru its FPGA-based design which allows for a large number of digital I/O modules. It also provides superior flexibility regarding their configuration, resolution, and functionality. In addition, the FPGA enables a software implementation of the host link - in the case of the gluion the OSC protocol as well as the underlying Ethernet layers.

KEYWORDS
Digital Sensors & Actuators, Sensor Interfaces, FPGA, OSC.

1. INTRODUCTION
The renewed interest in sensor technology in recent years has resulted in an enormous pool of resources, both in terms of online information and available tools. Part of this trend is an ever-growing number of sensor interfaces each with its own special feature set, concept, and price range. What they all have in common, however, is the fact that they are built around a microcontroller. This seems like a perfectly reasonable choice as these integrated circuits combine everything that is required for the design of a sensor interface: a CPU to handle program flow and perform arithmetics, memory to store programs and data, and I/O modules. The latter are of particular importance as they allow the microcontroller to connect directly to both sensors and host computer without the use of additional I/O chips. For the host link this is usually a serial interface like MIDI or RS232, or, in more recent designs, USB or Ethernet.

On the sensor side the most important I/O modules are the analog inputs that accept the voltages generated by the sensors. These are then measured by an integrated Analog-to-digital converter for subsequent processing by the CPU and transmission towards the host. In addition, microcontrollers provide a number of digital I/O pins that can be controlled, or queried, under program control. They can also be associated with additional internal modules to perform dedicated functions. A typical example is a PWM (Pulse-Width Modulation) generator that is often used to control the speed of a motor. This is usually implemented by using a timer module whose value is constantly compared to a register that contains the desired pulse-width. The (binary) result of this comparison then leaves the chip directly thru a dedicated pin which is connected to the motor circuitry.

Now, while the above approach is fine for many smaller projects, it is not uncommon that an instrument or installation design requires more than just a few PWM-, or other digital I/O-, modules. What's more is that the timers are also required for other tasks, e.g. generating the MIDI clock. Unfortunately, simply switching to a bigger microcontroller only helps to a certain extent as the focus is usually on more memory and general purpose pins rather than I/O modules. And adding more microcontrollers complicates the overall system design as they have to communicate with each other.

FPGAs, in contrast, do not have these limitations. As the name, Field Programmable Gate Array, suggests, the building blocks of these devices are gates, the logic foundation of any digital device. However, rather than wiring up countless TTL chips like in the early days, the designer can enter the schematic with a suitable editor[1] which is then compiled into a configuration file for the FPGA. This means that before programming the microcontroller one would first design it according to the requirements. While this may seem like reinventing the wheel, modern design tools hide a lot of the complicated details while the added flexibility outweighs the extra effort. More to the point though, we will see that many components of the microcontroller can be left out of the design.

2. BACKGROUND
The gluion is not the first device that employs FPGAs in a music technology context. Two examples for their use in custom controller designs are Dan Overholt's MATRIX [1] and the "continuous keyboard" by Freed & Avizienis [2]. What they have in common is a large number of identical sensors that would otherwise require a large array of microcontrollers. But their technology is very specific to the instruments and not available as a separate interface. The latter, however, is related

1. The preferred method though is the use of a hardware description language like Verilog or VHDL

to CNMAT's Connectivity Processor [3], an interface that integrates multi-channel audio as well as GMICS[2] and a "gesture port".

The only other pure sensor interface with an FPGA is surprisingly one of the first at all - STEIM's SensorLab [4]. For its time it offered remarkable features such as its own programming language SPIDER that was particularly well suited to event handling. It also supported special sensors like e.g. ultrasound distance measurement and was able to drive character displays over a synchronous serial link. While the first two revisions implemented this additional functionality with discrete TTL logic, rev. C combined a 80C535 microcontroller with a Xilinx LCA[3] to accomplish this task and expand on it.

However, this rather complex design came at its price, so in the following years simpler MIDI interfaces were released that focused on analog inputs and simple digital I/O. Another trend was that MIDI was used less and less to control hardware synthesizers and samplers, but was fed directly into computers as those became powerful enough for realtime software synthesis. This also meant that the controller no longer had to perform event processing as this was a comparably easy task for the computer, where it could also be implemented in a much more flexible way. The downside of this approach is that sensor data has to be streamed continuously which can be quite taxing for a slow interface like MIDI. Consequently the move to USB, Ethernet [5], and even digital audio [6].

3. TOWARDS FPGAS

3.1 Concurrency and Precision

If one dispenses with event processing like outlined above this also means that there is actually little left to do for the microcontroller's CPU in terms of performing arithmetical and logical operations. Tasks like signal filtering, threshold detection, or even simple scaling can all be done on the host now, so the microcontroller effectively becomes a data pump where its main job is now to configure, control, and query the individual I/O subsystems. The most common technique here is the use of interrupts that signal e.g. the completion of an analog-to-digital conversion or the successful transmission of a data byte/packet.

This approach can be managed comparably easy as long as there are not too many subsystems involved and interrupt service routines can be kept short. However, in the scenario proposed in the introduction, where more and more I/O modules are added, the overall timing can become critical as different interrupts collide and one has to deal with priorities and more complex scheduling strategies.

The situation gets worse when the desired functionality has to be implemented under program control. E.g. if an array of PWM outputs is required but only a single counter module is left, rather than writing the pulse-widths into their separate registers (and then "forget" them) the main process constantly has to read this one counter and perform comparisons to the PWM values stored in data RAM (using the ALU[4]), before it can set or clear the corresponding output pins accordingly. This sequential scheme alone results in a reduced accuracy of the PWM signal; it gets far worse when it is interrupted by other activities..

Now, how does an FPGA, or more specifically the gluion, handle this challenge. To begin with, there simply is no program flow to be interrupted. While it would not be a problem to implement a CPU inside an FPGA (including program counter, program memory access, ALU, instruction decoder, execution unit, etc.), it simply is not necessary. Instead all modules operate concurrently and communicate thru registers or dual-ported RAM. In other words, there are no shared resources that have to be arbitrated. Every PWM of the above array can have its own counter and pulse-width register, which are compared continuously and not just when the CPU gets to the matching program location. As a result the signal is accurate down to a single cycle of the master clock, 50ns in the case of the gluion. This may seem like overkill, but it allows the implementation of high-frequency PWM while maintaining high resolution.

3.2 Fine-tuning and unique functionality

Not only can an FPGA host a large number of precise I/O modules, it also allows the user to fine-tune their parameters. While counters and registers inside microcontrollers are usually fixed at 8 or 16 bit resolution, with an FPGA it is just as easy to have a 3-bit counter to drive an 8-way multiplexer, or a 50-bit counter to cover days. This allows the designer to balance the resources of the specific chip being used.

Furthermore, originally simple modules can be enhanced with functionality not available in most microcontrollers. In the case of the PWM example introduced above, this could be dynamic control of the frequency or a 'burst' register to specify a defined number of pulses to be sent rather than the usual continuous stream. Effectively this becomes a basic stepper motor control. More examples are discussed in chapter 5.

Finally, the FPGA allows for unique modules not found in any microcontroller, enhanced or not.

3.3 Portability

When porting an existing design to a new microcontroller-based interface it is rather likely that it will have a different chip than the previous one. This will possibly be from the same chip family or at least the same vendor, although sometimes the latter has to be abandoned when more drastic changes are required. In any case, the assumption that C-code should be easily portable is not entirely applicable here. Considering the above scenario where the microcontroller is freed from event processing and is mainly busy managing I/O registers, the programmer cannot simply transfer the existing C-code but has to map the old I/O register set to the new one, with all its pecularities. Similarly, functionality that has been implemented under program control has to be thorougly reviewed to match new timing constraints.

In the FPGA-case, this is less of a problem as we have defined our registers ourselves. In fact, the main feature of new FPGA releases is their bigger size. Exceptions are e.g. RAM blocks or PLLs[5], but these are ususally generic enough, so the design can be easily adapted. Admittedly at the current state this is somewhat theoretical though as it is not backed by extensive experience.

Finally, it should be noted that by defining the very details of a

2. A bi-directional interface for audio and sensors proposed by Gibson
3. LCA = Logic Cell Array, conceptually similar to FPGAs
4. ALU = Arithmetic Logic Unit
5. PLL = Phase Locked Loop

chip's architecture it is possible to apply the concepts of open-source software to hardware designs, with all their socio-economic implications.

4. DESIGN OF THE gluion

In designing the gluion's hardware one of the goals was to use as few external components as possible. This was particularly true for the Ethernet-based host link. Following the above portability goal, no external Ethernet-PHY (physical layer) chip was used with all the lengthy and device-specific setup procedures. Instead this functionality was implemented entirely as a soft core, so for this part the only external components are the RJ-45 connector and the isolation transformer. On the downside, only 10base-T could be achieved, but this is usually enough for sensor data.

On top of this lie rudimentary data and transport layer modules. Rather than going for a complete TCP/IP stack only those protocols are implemented that are necessary for OSC, i.e. UDP, ARP, and IP. The choice for OSC [7] was for its high speed, powerful protocol, and driver/OS-independency.

Figure 1: the gluion 'barefoot'

Apart from the I/O modules described above and in the following chapter, the gluion provides functionality to upload a new configuration via Ethernet. This means additional flexibility as the interface can be adapted when moving from project to project. Users can enter their requirements into a web interface which generates a script that drives the build process of the FPGA design tools. At the current state this is not yet fully automated though.

5. EXAMPLES FOR MODULES

5.1 Scanned Switch-Matrix

While connecting a switch to a sensor interface is a simple task with just about any sensor interface (usually a pull-up resistor and the switch/button towards ground), setting up a matrix of switches requires a multiplexed approach with both in- and output pins.

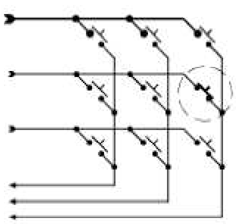


Figure 2: 6 I/O lines required for 9 buttons

The latter have to send synchronized pulses into the rows of the matrix, while the columns are connected to the inputs that listen for returned pulses which can then be matched to specific switches. The advantage of this concept is that less I/O pins are required compared to the pin-per-switch approach (approx. $2\sqrt{N}$ vs. N). As an example consider an interactive installation with a grid of 30x30 floor switches - the ordinary approach would ask for 900 pins, which is clearly beyond any available interface. In contrast, the gluion with its 68 digital pins can easily accommodate the 60 connections required for such a matrix.

5.2 Rotary Encoder

This component can be frequently found in commercial music gear as well as other industries where it is used to adjust values that are under software control. In contrast to a potentiometer it has no stops but can be turned "endlessly". In doing so it outputs a dual stream of pulses that can be counted while their phase relationship signals the direction of the rotation.

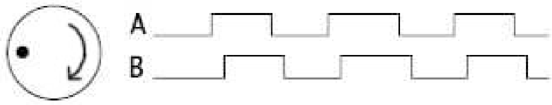


Figure 3: dual pulse trains from an encoder

A microcontroller could do this task under program control, but none of the reviewed sensor interfaces do this. It also becomes difficult to capture all pulses when using high-resolution encoders that create hundreds of increments per 360º turn. In an FPGA a simple state machine connected to a counter ensures that no pulse nor phase is lost.

Figure 4: optical encoder from an old mechanical mouse mounted on the fingertip of a data glove

5.3 Ultrasound Distance Sensor

One way to measure distances is to send out an ultrasound pulse and wait how long it takes to reach the receiver. Three timers/ counters are required here: one to generate the 40kHz signal that common transducers operate at, one to count to ~10 for a small burst, and finally the actual time lag counter whose value is proportional to the distance covered.

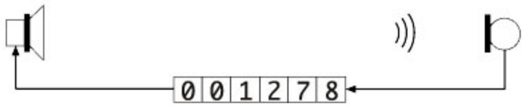


Figure 5: measuring distance by timing ultrasound lag

As the time lag counters are driven by the 20MHz master clock the internal resolution is 50ns equivalent to 0.017mm. The effective resolution, however, is limited by the thresholding circuit and is about a magnitude less. But compare this to the 1ms resolution achievable (at best) when doing the timing on the host machine, which translates into roughly 30cm.

Some ultrasound designs built for more common sensor interfaces actually have their own microcontroller to determine a distance value, which they then have to convert into a voltage to be measured by the interface's ADC. These additional steps obviously limit the achievable resolution and are hardly elegant.

5.4 Frequency Measurements

A frequency counter is ususally just a counter that is clocked from an external signal for a certain period called the gate time (which requires another counter). It can be used for sensors where the sensing element is part of an oscillator whose frequency changes depending on the physical parameter. Of course, frequencies can be converted to a voltage, however this adds to the complexity. Moreover, the frequency can be sent over long lines with less signal degradation than a voltage. In the example below a low-cost inductance sensor has been fitted on a tuba to measure its key positions.

Figure 6: left: coil free-standing, right: coil covered by tuba key

In addition to the single-channel frequency module the gluion also has a multiplexed version that is e.g. used to measure the 61 key heights of the SKRUB keyboard [8].

5.5 Pulses

For signals with low frequencies it is often better to measure their period. The process is complementary to the above frequency measurement, as the external signals provides the gate for a counter driven by the internal clock.

Figure 7: optical sensor and motor of a belt-driven record player

In the upper part of the above picture an optical sensor measures the duration of black bars passing by as the turntable rotates, effectively determining its speed. Pulses also flow in the other direction as a PWM signal controls the motor located in the lower part of the image.

5.6 Serial Data

While the above modules allow the direct control of many different sensors and actuators there are often existing devices that need to be tied into the overall system. Wether it is a Polhemus 6DOF tracker, a character display, or a DMX lighting setup, they usually communicate thru a serial interface. For this it is possible to configure modules for various serial protocols. Currently available are RS232, MIDI, DMX, and SPI. I2C and PS/2 are planned.

6. FUTURE PLANS

Currently wireless transmission of sensor data is possible through a simple WiFi bridge. However, a more integrated solution is being considered to keep the overall design compact.

Several enhancements to the existing I/O modules are underway as are entirely new modules like a touchscreen controller.

Other plans call for more hardware modules, like ultrasound amplification or drivers for motor control.

Finally a move to a bigger FPGA might be necessary to host more complex designs.

7. ACKNOWLEDGMENTS

I would like to thank Podewil (Elke Moltrecht) as well as its successor Tesla (Carsten Seyfarth) for their continued support in hosting the glui lab. Also thanks to Nic Collins and Reinhold Friedl.

8. REFERENCES

[1] D. Overholt, The MATRIX: A Novel Controller for Musical Expression, In *Proceedings, CHI '01 Workshop on New Interfaces for Musical Expression (NIME'01)*, *Seattle, WA,* 2001.

[2] A. Freed and R. Avizienis: A New Music Keyboard featuring Continuous Key-position Sensing and High-speed Communication Options, In *Proceedings, International Computer Music Conference, Berlin, Germany,* 2000.

[3] A. Freed, R. Avizienis, T. Suzuki, and D. Wessel: Scalable Connectivity Processor for Computer Music Performance Systems, In *Proceedings, International Computer Music Conference, Berlin, Germany,* 2000.

[4] http://www.steim.org/steim/sensor.html

[5] T. Coduys, C. Henry, and A. Cont: TOASTER and KROONDE: High-Resolution and High-Speed Real-time Sensor Interfaces, In *Proceedings, New Interfaces for Musical Expression (NIME-04) Hamamatsu*, 2004.

[6] J. Allison and T. Place: Teabox: A Sensor Data Interface System, In *Proceedings, New Interfaces for Musical Expression (NIME-05) Vancouver*, 2005.

[7] M. Wright and A. Freed: OpenSound Control: A New Protocol for Communicating with Sound Synthesizers, In *Proceedings, International Computer Music Conference, Thessaloniki, Hellas,* 1997.

[8] http://glui.de/prod/gluiph/SKRUB.html

Beyond 0-5V: Expanding Sensor Integration Architectures

Adrian Freed, Rimas Avizienis, Matt Wright
CNMAT
1750 Arch Street
Berkeley, CA 94709
1 510 643 9990 x308
{adrian, rimas,matt }@cnmat.berkeley.edu

ABSTRACT

A new sensor integration system and its first incarnation is described. As well as supporting existing analog sensor arrays a new architecture allows for easy integration of the new generation of low-cost digital sensors used in computer music performance instruments and installation art.

Keywords

Gesture, sensor, MEMS, FPGA, network, OSC, configurability

1. INTRODUCTION

After years of experiments and live musical productions with several revisions of analog sensor input systems for an audio connectivity processor [2] we have learned that the basic multichannel 0-5v analog input standard is not sufficient for access to interesting new sensing technologies.

The trend for voltage output sensors is towards lower output levels, e.g. 3V. MEMS sensors and hall effect sensors use clocked sampling techniques internally and they currently convert their measurands into analog signals that are then resampled. More accuracy and lower costs are achieved as the switch to digital outputs is made.

A new, simple way to support a wide variety of these new sensors is needed especially the emerging diverse, low-cost MEMS sensors. Unfortunately, no single serial digital communication standard has curried favor with the sensor vendors themselves necessitating "bridging" hardware to hide these complexities from users.

Another major requirement in many musical instrument and art installation projects is for considerably more input channels than is usually provided in USB data acquisition systems or current gesture input boxes. Many sensing problems are most easily addressed by using linear and 2-dimensional arrays of networked sensors. We routinely find need for more than 32 channels.

In this paper we review several sensor integration projects, outlining their specific requirements and then we describe how these requirements are addressed by a new architecture for sensor integration based on FPGA's and sensor identification. Finally, we describe the first implementation of this as the latest refinement of our ongoing connectivity processor project.

Table 1. Sensor Technology Sample

Device	Measurand	Analog	Digital
ADXL213	Acceleration		PWM
ADXL320	Acceleration	0-5V	
ADIS16006	Acceleration		SPI
ADIS16100	Yaw Rate		SPI
ADXRS150	Rotational Velocity	0-5V	
QT401,501	Proximity		SPI
AS5040	Absolute position (rotary)		SSI
GP2D12	Proximity (IR)	0-3V	
GP2D02	Proximity (IR)		bit-serial
VP300	X,Y+pressure	Resistive array	

2. Applications

2.1 Proximity Detector Array

This array, developed by Michael Zbyszynski, satisfies the need for a sensing device that can be mounted into scenery and that can detect a large range of performance gestures from dancers. The prototype shown in Figure 1 is a 4' x 4' wooden box containing sixteen infrared range sensors. The sensors sense distance in a range of approximately 8" to 60", and are mounted beneath the top of the box, under 2" acrylic windows. This creates a floor that is strong enough for two people to dance on, and has no rough edges. It can also be tipped on edge for a different sensing perspective. In performance, a similar array was built into a curved wall eight feet high. By comparing the inputs from multiple sensors, motion and position can be deduced, as well as judgments about the shape and posture of the body.

Furthermore, the dancers are free to move in a relatively wide area. Unlike video tracking, the array is dependable –– insensitive to the changing conditions of theatrical lighting.

Future realizations will need a denser and larger array of sensors requiring 64-128 channels of data acquisition.

Figure 1. Dance sensor Array

2.2 Tiled X/Y + Pressure Pads

This hand controller, conceived by David Wessel, consists of a central array of 24 FSR-based trackpads surrounded by long position- and pressure-sensing strips and IR proximity detectors. Five measurands are available from each 2-d track pad and two are available from the position sensing. The trackpads are 4-pin devices with internal FSR components. They require custom circuitry to tease out the x,y position and width and pressure parameters by steering controlled currents around the array and measuring the voltage induced in the resistors. This may be contrasted with one dimensional position sensors that are usually wired simply as voltage dividers for interface to voltage input sensing systems. When more complex circuitry is used the width of the object being sensed may be estimated by the bulk reduction of resistance across the sensor due to the varying width of the conductive "wiper" contact.

The fully tiled array of 24 3-dimensional sensors (x,y and pressure) provides 72 variables of output. To achieve a degree of control intimacy similar in character to an acoustic hand drum, the array must be sampled at rates near 8,000 Hertz. This variable count and sample rate far exceeds the capacity of gesture capture systems currently available such as the AtomMic Pro, EoBody, and Infusion Systems' Digitizer [4]. Consequently an expanded view of sensor integration is required.

2.3 Touch display

This display has wide application in musical controllers. We are exploring the integration of commercially available 3" and 4" diagonal touch screens into an augmented cello [9] and the aforementioned hand controller. They are addressed by serial protocols such as RS-232 or SPI.

2.4 Harp Controller

This controller developed by Adrian Freed consists of 36 identical length nylon harp strings in a rectangular frame that is sensed with individual piezoelectric pickups. The audio signal from each string is used for trigger and timbral information so an 8kHz sample rate is used rather than standard audio sample rate.

Figure 2. Harp Controller

2.5 Compact Position sensor

David Wessel's SmartSticks project uses MEMS accelerometer- and gyro-equipped drum sticks sampled at a 4k Hz rate. The goal is to accurately sense the trajectories of each stick and its position relative to an array of percussion instruments. Sensors are mounted on both ends of each stick allowing the stick orientation to be sensed. Wireless transmission of the stick sensor data is beneficial, providing the percussionist with freedom of movement.

Another important option in this project is eventual integration with a high-speed video capture system.

This array is also used extensively on the flute by Roberto Morales [7], for example in performances of his piece "Cenzontle" which one first prize at the 2005 Bourges competition. The array uses two rotational MEMS gyros and a two-axis MEMS gyro. A low frequency filter on the accelerometer is used for tilt estimation. A high frequency is used to capture key click and other transients. Two rotation rates are estimated corresponding to pitch and yaw of the flute which are free dimensions of control for the performer.

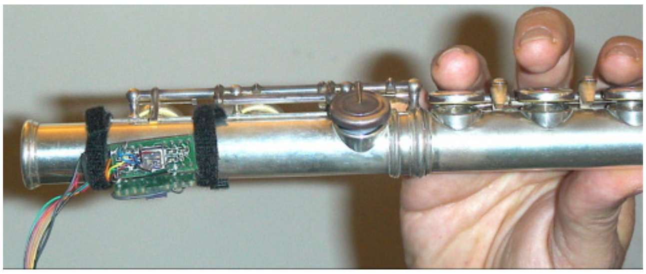


Figure 3. MEMS position sensor

To increase the mobility of the performer and provide flexibility for installation in other instruments (and bows) we have developed an RF interface to the sensors.

3. Solutions

One approach to these projects is to build entirely independent custom electronics for each. They have widely varying numbers of analog and digital sensors and standards and bandwidth requirements. We have found it more efficient to use hybrid architecture with a common motherboard fulfilling the many common requirements and customized "daughter" boards for custom needs.

Our connectivity processor has a motherboard with 8-channels of balanced audio D/A conversion, 2-channel headphone output, AES-3 I/O, ADAT optical I/O, multi-channel sync. I/O, MIDI I/O, a high speed (GIG) and 100BaseT Ethernet [1].

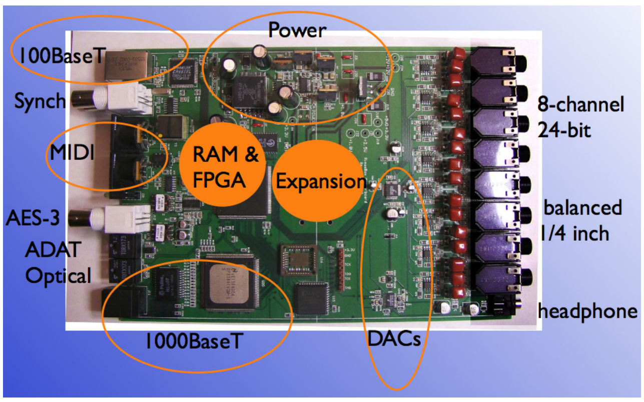


Figure 4. Connectivity Processor motherboard

Figure 5. Gesture inputs and Digital I/O

This motherboard supports two daughter cards: one is used usually for audio input, e.g., 8-channels of balanced audio; the other is a gestural input card.

The basic gestural input card supports two DB25 connectors carrying power and 16 channels of 0-5V inputs on each with sample rates up to 8kHz. Although 32-channels 0-5V channels has proven extremely valuable for many projects our current projects require more inputs and support for more diverse digital standards for sensor integration.

4. Beyond 0-5V

To address the need for broader range of sensors and sensor output formats we have developed a lightweight standard that covers the connector, power and configuration aspects of a sensor network. By hiding the complexity of the communications issues in affordable and flexible FPGA (Field Programmable Gate Arrays) we avoid the difficulties and cost of previous sensor network standards which have failed to take hold such as IEEE 1451 [5] or BISS [3].

4.1 Dynamic configuration

We use a configuration pin to dynamically assign functions to 6 digital I/O pins on each connector. This allows for a wide variety of clock rates, channel assignment and signaling protocols. Controllers, implemented with FPGA's can readily support the serial protocols chosen by the sensor manufactures including: I2C, SPI, 1-wire, RS232, RS422, MIDI, TTL, USB etc. In most cases modules in VHDL or Verilog are already available to support the protocol and usually for free [6]. Two configuration methods are supported: a single resistor to GND identifies one of 32 commonly used devices, alternatively a Maxim/Dallas 1-wire EEPROM device may be connected using the same pin. This allows for thousands of different device types to be supported. Since 1-wire devices have programmable storage this allows configuration and calibration data to be stored with the sensor array itself allowing the sensors to be moved transparently from one controller to the next.

4.2 Connector

A DB-9 was selected for the connector type because it is a widely available, reliable, lockable connector, has sufficient power handling for most sensor/actuator applications and is large enough that most existing and future sensors can be adapted to the FPGA signaling options with small circuits built-into the connector itself. Many digital sensors can be interfaced without a circuit board at all. Analog sensors requiring a few channels, e.g. MEMS accelerometers and gyros can be connected to a 4-channel A/D built into the shroud. Sensors that require more electronics can be integrated using larger shrouds around the DB9 pins including readily available DB9->DB9, and DB9->DB25 shrouds and small "project" boxes available with built-in DB9 connectors.

4.3 Power

5V @500mA is provided allowing the use of cheap power protection devices developed for. Most sensors require a small fraction of this power but sufficient power is provided for other devices such as actuators, relays and the occasionally inherently power-hungry sensing technology, e.g., IR and ultrasound rangers.

5. New Controller

Figure 6 shows the first implementation of our new approach to sensor integration, a daughter board for the connectivity processor. It implements 4 flexible ports, 4 legacy analog input ports, a wireless system for remote sensing and a large Spartan III FPGA which has sufficient memory, gates and high speed multipliers to support the many protocols required and perform DSP and sensor calibration functions.

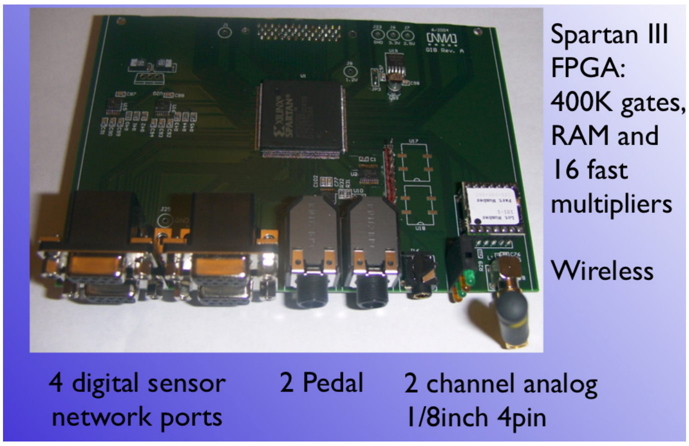


Figure 6: Flexible Sensor Integration Daughterboard

6. FPGA vs Microcontroller vs DSP

Most computer audio interface boxes and gesture interface boxes use a mix of FPGAs, a general purpose microcontroller and more recently some include a DSP chip. At CNMAT we have deliberately avoided this approach preferring to maximize development efficiency and flexibility by using only one development environment - for FPGA's. We also use simple, well-documented standards wherever possible, i.e. Ethernet. Firewire and USB are notoriously complex to develop reliable drivers for.

The problems in our applications with microcontrollers stem from the fact that they are optimized to trade performance for cost-efficiency and this usually means too little parallelism. For example, they may be fast enough to support software implementations ("bit-banging") for a few SPI serial ports and some even include built-in hardware buffering for one or two SPI ports. This doesn't scale well when dozens of sensors are employed. Current FPGA's on the other hand have plenty of available pins and all the parallel bit-manipulation capabilities to support 40 or more SPI ports.

We have found our systems to be cheaper to develop, cheaper in overall parts cost and more reliable than the conventional mixed device approach. The reliability comes from lower overall complexity and how little outside intellectual property has to be integrated.

We also note that since the Xilinx FPGA's we use are large enough to contain processors and DSP we can integrate conventional programming approaches and custom signal processing if necessary.

7. Software Support: /dev/OSC

Since the sensor output from the new controller varies according to the sensors plugged in, a flexible and extensible protocol is needed to communicate the data to application programs. The OS/X device driver for the connectivity processor extracts audio and gestural data from incoming Ethernet packets and routes it through to Core Audio and Core MIDI. Gestural data is up-sampled into audio streams for reliable real-time delivery for Core Audio. This technique does not scale well to dynamic situations possible with our new daughterboard. In this case we build OSC [8] formatted data which is queued by a special driver accessible as a UNIX file /dev/OSC. This technique avoids priority inversion problems we have observed in both the OS/X and linux operating systems when we route gestural data through the TCP/IP stack. Performance and reliability of this system were confirmed in a concert in November 2005 featuring an augmented cello played by Frances-Marie Uitti [9].

8. Conclusions and Future Work

Our new daughterboard gestural integration card controller is a solid starting point for our ongoing work to integrate a diverse range of sensors. We are extending our early favorable results in two directions: a more compact version for wearable wireless applications and a larger system combining our controller with 48-channels of analog data acquisition for big hybrid sensor applications.

9. Acknowledgements

We gratefully acknowledge David Wessel and the Chambers Fund for supporting development of the new system. Xilinx sponsors our use of their FPGA development tools.

10. REFERENCE

1. Avizienis, R. and Freed, A., OSC and Gesture features of CNMAT's Connectivity Processor. in Open Sound Control, (Berkeley, California, 2004).
2. Avizienis, R., Freed, A., Suzuki, T. and Wessel, D., Scalable Connectivity Processor for Computer Music Performance Systems. in International Computer Music Conference, (Berlin, Germany, 2000), ICMA, 523-526, http://cnmat.CNMAT.Berkeley.EDU/ICMC2000/pdf/connectivity-processor.pdf.
3. BISS:. BISS, 2006, http://www.biss-interface.com/.
4. Fléty, E., AtoMIC Pro: a Multiple Sensor Acquisition Device. in International Conference on New Interfaces for Musical Expression (NIME-02), (Dublin, Ireland, 2002), 96-101.
5. IEEE. Draft Standard for a Smart Transducer Interface for Sensors and Actuators, 2006, http://ieee1451.nist.gov.
6. Lampret, D. Open Cores, 2006, http://www.opencores.org.
7. Morales, R.a.M., Eduardo, Combining Audio and Gestures for a Real-time Improviser. in International Computer Music Conference, (Barcelona, 2005), ICMA.
8. Wright, M. Open Sound Control as an Enabling Technology for Networked Music. Organised Sound, 10 (3). 193-200.
9. Freed, A, Uitti, F, Wessel, D., Zbyszynski, M, Augmenting the Cello, NIME 2006, Paris, France,

Timbre interfaces using adjectives and adverbs

Colin G. Johnson
Computing Laboratory
University of Kent
Canterbury, England
C.G.Johnson@kent.ac.uk

Alex Gounaropoulos
Computing Laboratory
University of Kent
Canterbury, England
ag84@kent.ac.uk

ABSTRACT

How can we provide interfaces to synthesis algorithms that will allow us to manipulate timbre directly, using the same timbre-words that are used by human musicians to communicate about timbre? This paper describes ongoing work that uses machine learning methods (principally genetic algorithms and neural networks) to learn (1) to recognise timbral characteristics of sound and (2) to adjust timbral characteristics of existing synthesized sounds.

Keywords

timbre; natural language; neural networks

1. INTRODUCTION

When human musicians "interface" with other human musicians, we do so in words. One way in which we use words is to communicate about timbre: we say "can you make the sound shine more", "I'd like to get a really gritty sound", "let's try to play that more warmly".

This ability to use natural language descriptions of timbre (whether "straight" or metaphorical) is typically absent from interfaces with music technology devices. As a result, users have to either have a very strong understanding of the underlying mechanisms that produce the sound, or a large amount of "trial-and-error" experience with generating timbral changes within a system [5, 11]. A small number of attempts [2, 7] have attempted to create systems that offer an intuitive interface to timbre; however, there appears to be little recent work in this area [9].

By *timbre* we will mean the micro-level spectral characteristics of sound as discussed by Wishart [10], as opposed to the gross timbral distinctions [6] used e.g. in the MPEG-7 standard.

In this paper we discuss ongoing work which applies machine learning methods to associate changes in synthesis parameters with changes in timbre described by adverbs and adjectives. Overall the system is structured as illustrated in figure 1. Before the system is used, a training process is carried out whereby the system learns to associate features of sounds with certain timbre-description words that have been allocated by a human listener. Then,

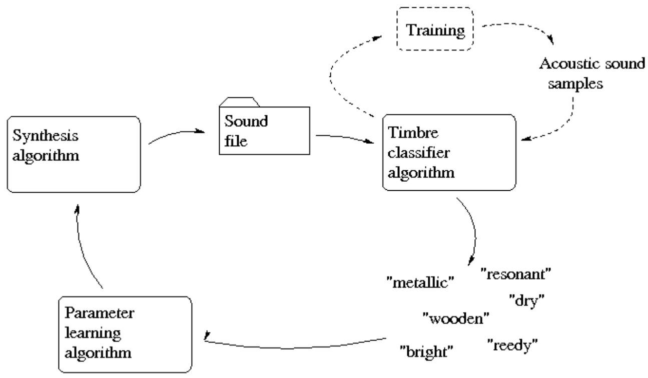


Figure 1: Indirect modification of timbres.

when the user wants to adjust the sound with regard to a particular word, an algorithm that adjusts the synthesis parameters (e.g. with a genetic algorithm), uses the synthesis algorithm to generate a sound with regard to that parameter-set, then uses the trained classifier to measure the timbral effects. The levels of these timbral effects are then measured and fed as a *fitness measure* into the parameter adjustment algorithm.

The remainder of the paper describes the two main parts of the system: the timbre classification program, and the parameter learning algorithm. Full details of this can be found in our paper [3].

2. PART 1: TIMBRE RECOGNITION ALGORITHM

Timbre recognition can be seen as a *classification* problem of the type well studied in machine learning [8]. In such a problem we have a number of data items each of which is described by the values of a number of *attributes*; each data item also fits into one or more *classes.* In this case each data item is a sound. The attributes of that sound are a number of measures that are made on that sound: the relative peak amplitude of the first 15 partials of the sound, the detuning of the partials, and measures of the amplitude envelope acquired by fitting an ADSR envelope to the sound. The classes are a number of timbre-words (drawn mostly from the list in [2]); a human listener has, for each sound, rated how much they believe that that word describes the timbral characteristics of the sound. In our current program, the words used are *bright, warm, harsh, hit, plucked, constant, thick, metallic,* and *woody.*

A common approach to such classification problems is

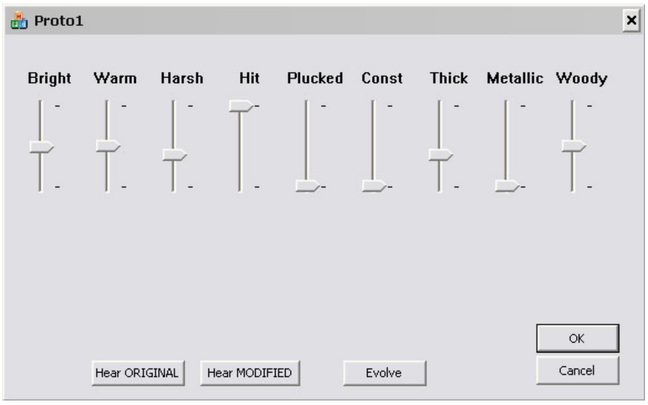


Figure 2: Adjusting timbres with sliders.

to use neural networks to learn which patterns in the attributes map on to which classes. This is the approach that we have used here. The inputs to the network are the attribute values described above, the outputs the timbre words. We use the well-known *back-propogation* algorithm (e.g. [1]) to learn the network weights using a large number of training examples. To test the classification, we fed a number of (attributes describing) new sounds, not used in training, through the network, and observed the value given for each of the timbre words by the network. Overall, a good classification of timbres was obtained when compared to classification by a human listener (details in [3]).

3. PART 2: PARAMETER ADJUSTMENT ALGORITHM

The second main part of the system is an algorithm that adjusts the parameters of the synthesis algorithm to respond to a desired timbre change. The current interface to the system is illustrated in figure 2. The user inputs a sound, the neural network assesses the timbral characteristics of the sound according to the ten timbre-words used, and the system sets the sliders accordingly.

The user then makes changes to the sliders to reflect the desired change. We have experimented with various algorithms to generate the desired change in synthesis parameters. In the first, we use a genetic algorithm to find a new setting for the synthesis parameters that (1) is close to the original settings (so that the overall sound does not change) and (2) reflects the desired timbral change. A second approach involves feeding the new desired values for the timbral characteristics through the neural network, to find regions of parameter-space that are associated with these settings, then finding the smallest parameter-space move required to go from the original sound to those regions that produce the desired timbre. Both approaches produce decent results; at present the second approach seems more promising.

4. RESULTS

Some example sounds can be heard at `http://www.cs.kent.ac.uk/people/staff/cgj/research/nime2006/nime2006.html`. We will demo the system live at the conference.

5. ONGOING WORK

We are currently pursuing a number of future directions for this work:

- Improved timbre recognition, for example by using ear-like pre-processing steps [4].
- Focusing more effort on having the system *not* adjust those characteristics of the sound that are not relevant to the current
- Interfacing this to e.g. MIDI control wheels to allow on-the-fly live manipulation of timbre.
- Creating systems that learn in advance the directions in parameter space which affect a particular timbre change, rather than running the parameter-adjustment algorithms on the fly. This may be a prerequisite for the use of the system in a live environment.

6. REFERENCES

[1] R. Beale and T. Jackson. *Neural Computing.* Institute of Physics, 1990.

[2] R. Etherington and B. Punch. SeaWave: A system for musical timbre description. *Computer Music Journal*, 18(1):30–39, 1994.

[3] A. Gounaropolous and C. Johnson. Synthesising timbres and timbre-changes from adjectives/adverbs. In F. Rothlauf et al., editor, *Applications of Evolutionary Computing.* Springer, 2006.

[4] D. M. Howard and A. M. Tyrrell. Psychoacoustically informed spectrography and timbre. *Organised Sound*, 2(2):65–76, 1997.

[5] C. G. Johnson. Exploring the sound-space of synthesis algorithms using interactive genetic algorithms. In G. A. Wiggins, editor, *Proceedings of the AISB Workshop on Artificial Intelligence and Musical Creativity, Edinburgh*, 1999.

[6] S. McAdams, S. Winsberg, S. Donnadieu, G. de Soete, and J. Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, 58:177–192, 1995.

[7] E. R. Miranda. An artificial intelligence approach to sound design. *Computer Music Journal*, 19(2):59–75, 1995.

[8] T. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[9] A. Seago, S. Holland, and P. Mulholland. A critical analysis of synthesizer user interfaces for timbre. In *Proceedings of the XVIIIrd British HCI Group Annual Conference.* Springer, 2004.

[10] T. Wishart. *Audible Design.* Orpheus the Pantomime, 1994.

[11] T. Wishart. *On Sonic Art.* Harwood Academic Publishers, 1996. second edition, revised by Simon Emmerson; first edition 1985.

SonicJumper composer

D. Andrew Stewart
Schulich School of Music
McGill University
+1 514-845-0766
dandrew.stewart@mcgill.ca

ABSTRACT

This document describes the implementation of the *SonicJumper* gestural controller bodysuit in a compositional context. It is a tool for generating musical materials, which are then used to compose a piece of music. The emphasis is on integration of gestural controllers at the earliest stage of the compositional process, rather than at the end. That is to say, the following discussion centers on controllers as a tool for creating musical material, and not as instruments for a performance. An effective compositional tool provides the composer with a manner of producing materials that have an inherent musical quality lending themselves to the formation of musical messages, which are then organized into a meaningful compositional whole. The author regularly incorporates the *SonicJumper* into his compositional process, generating materials for mixed works—compositions for ensemble and electronics.

Keywords

composition, process, materials, gesture, controller, cross-modal interaction

1. INTRODUCTION

There has been a strong push to extend musicians' potential for musical expression, via gestural controller technology. Amazingly, this propulsion has lead to a wealth of new research linking musical expression to such things as: new instrument design and control; redefining performance practice; gesture analysis and classification; gestural acquisition; music cognition and other branches of psychology. At the same time, the ultimate question still remains unanswered: Can gestural controllers be successfully woven into a musical fabric, such that the technological aspect is far less significant than the overall musical experience? Moreover, might the inclusion of gestural controller technology into a musical domain lead us to a new Art form? These questions highlight the author's principal focus. He postulates that the answers lie in our ability to create a musical whole. That is to say, electroacoustic elements and human expression are integrated—creating a musical whole—if they are perceived as inextricable. A successful composition, including work in other Art forms, is one in which the artist unifies materials. Materials are all aspects of a work that are cognitively perceptible. In particular, the manner in which materials are created, must be directly linked to their final presentation. This is obvious to a great extent in the visual Arts. Most music is also generated with clear evidence concerning its impetus—often drawn from codified stencils dictating formal design and pitch logic. The inclusion of technology into a musical work, however, creates a number of dilemmas surrounding the initial generation of materials. The immediacy at which technology so readily gives us an 'output' does not often encourage an exploration into where the output comes from or how it is generated. This potentially drives a wedge between the materials of the work and the composition's final form: its presentation in the context of a public performance. If a 'whole' musical experience is to be created, the technological aspect must be a part of work's initial development. It is the objective of this document to identify how the *SonicJumper* generates material in the earliest stage of composing, unifying the piece right from the get-go.

2. MALLEABLE GESTURAL CONTROLLER

2.1 Components

Four accelerometers (± 2 G.), five potentiometers (measuring bend from 0° to 130°), one infrared proximity sensor (80 cm.) and an orientation sensor (360°) sense body movement. Voltage values from these sensors are converted to MIDI. Max/MSP interprets and maps sensor data, controlling digital signal processing. The sensors are held in place using various types of sport braces—stretchable bands of fabric that comfortably fit around the body and do not limit movement. The voltage to MIDI convertor rests in a belt pouch along with its portable power supply. One long MIDI cable connects the convertor to a computer. Sensor placement is somewhat different for each project. It is for this reason that the jumper is considered a malleable controller—it shapes itself according to the movement requirements of each project. [1]

2.2 Synthesis Engine

Data from sensors are sent to Max/MSP, for digital signal processing. The sensors are not transmitting "one-off" triggers; rather, they are sending variable signals in real-time. The Max patch is based on *granularized*, by les & zoax, in which a signal "scrubs" through a buffer~ object at a user-defined rate. [2] Common effects associated with granular synthesis are achieved (i.e. time and pitch scaling). In addition, the Max patch is expanded to include various filtering objects, which are used to balance signal output—as opposed to creating effects such as chorus and delay. The engine is not necessarily meant to produce a specific style of composition. It is not meant to generate, say, 12-tone music or formulaic commercial music. The aim is to tap into the composer's expressiveness in a manner that is impossible with more traditional compositional tools (i.e. piano), and to offer users a sound palette that is representative of the wide-open sound world of electronics—both mimetic and abstract.

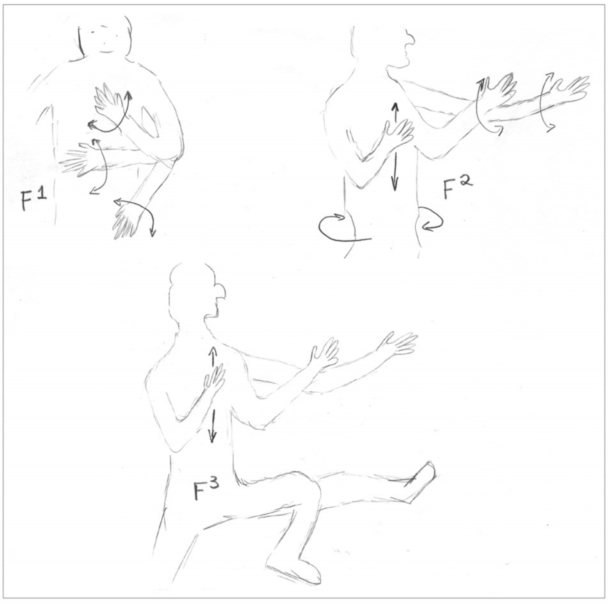


Figure 1. The first three feints.

2.3 Mapping the Movement Repertoire

Sparring rationale is used to develop the movement repertoire and sensor to signal processing mapping. In this way, the *SonicJumper* is a symbolic immersive controller. [3] There are five movement categories representing the tools of combat: feinting; drawing; leading; infighting; parrying. Figure 1 shows the first three feints. Each category contains a collection of precise coordinated head, arm, hand, torso, etc., movements —organized according to the expanse of the gesture (see Table 1). In total, there are 18 distinct movements. Mapping is derived to suit the movement repertoire. That is to say, instead of tailoring the repertoire to a fixed, rigid association to sensor data, mappings vary for each movement. For instance, the subtle action of the first feint (F1) only entails mapping the scrub start and end position (within buffer~) to left hand rotation and left elbow bend, respectively. On the other hand, the second feint (F2) entails a more complex mapping: start position to right elbow bend; end position to left elbow bend; lower limit of pitch-scaling to left hand rotation; upper limit of pitch-scaling to proximity of hand(s) to chest. For the most part, mapping is one to one, with a few examples of divergent mapping. For instance, the first parry (P1) involves a subtle rotation of the head. In this case, data from the orientation sensor is mapped to almost all granularization parameters.

3. SONIC-JUMPING

3.1 Cyclic Relationship

The core concept behind sonic-jumping is cross-modal interaction. The composer is spurred on—in particular, by aural and proprioceptive stimuli—to digest and produce sound in a cyclic manner. One can conceptualize the 'path of sound' as: out of an electronic system - into the human physiological system - returning to the electronic system. For example, an electronic sound is produced by a computer. Then, the ears receive the sound. A meaningful message is perceived (through cross-modal sensory data). Next, a movement impulse is manifested. The computer interprets user movement (via a gestural controller bodysuit). Movement data generates an electronic sound. For the most part, a composer's manipulation, or directing, of sound in this fashion, is unconscious. They do not naturally analyze the gestures they make in relation to aural stimuli.

3.2 Sound Movement Combinations

A work of Art conveys a message that is more or less clear, based on the way the message's meaning is distilled, generally speaking. Messages are anything that have either abstract or literal meaning for the onlooker. The aspiration of any artist is to provide clarity so that an audience can extract and refine messages. Providing clarity in a piece of music can be particularly difficult, because musical sounds are ephemeral. That is to say, music is a time-based Art form, with sounds only occupying enough time for them to be heard. A sound does not rest in one place, as a painting hangs on the wall for the duration of its exposition. The *SonicJumper* approach to creating clarity involves attaching a gestural component to each sound, or a sound to each gesture (refer to Cyclic Relationship, above). For the composer, the invention of a movement element gives new meaning to the sound / movement combination. The composer gradually establishes a somatic relationship between the two. This is followed by the formation of musical messages as the composer organizes somatic meaning. The suggestion here is that by infusing musical

Table 1. Five movement categories showing minute movements to expanded movement from top to bottom.

Expanse	Feint	Draw	Lead	Infighting	Parry
minute	F1 - subtle	D1 - subtle			P1 - subconscious, self-preserving
↑	F2 - false start, stunted	D2 - deceptive, with purpose			
	F3 - deceptive, with purpose	D3 - luring, enticing		I1 - reactive	
	F4 - reactive	D4 - expressive, engaging	L1 - expressive, engaging	I2 - expressive, engaging	P2 - evasive, escaping, disengaging
↓		D5 - impressive, refined	L2 - assertive, direct		P3 - impressive, refined
expanded		D6 - demonstrative, exaggerated	L3 - demonstrative, exaggerated		

materials with a somatic significance, at the earliest stage of composition, the composer creates repercussions for the presentation of the work, in front of an audience. They also create structural threads that are used to make the piece of music a unified whole. The notion of music combined with movement, or visa versa, is currently under investigation; and results are far from conclusive. In explaining the origins of somatic meaning, while sonic-jumping, this author favors one of the oldest investigations in experimental psychology concerning the nature of cross-modal sensory interactions—the degree to which information from one sensory channel influences our interpretation of information arising through other sensory channels. [4] Cross-domain mappings, enable us to perceive intensity, spatial location, tempo and rhythmic structure in an amodal manner. These abilities, moreover, appear to be innate or develop early and rapidly in human development (Lewkowicz, 2000). [5] There are other plausible explanations explaining our propensity to combine sound and movement. One could begin with a Darwinian perspective, which suggests that our internal sense of self-motion may have evolved in early hominids to deal with sounds in the environment. [6] We could also consider artistic emotion and expression. Davies (1994), suggested emotions are presented directly in the musical work through dynamic parallels to human movement, behavior, physiognomy, the human voice, gait and the like. [7] It is likely that all of the above viewpoints play a role in establishing meaning in sound / movement combinations. It is not the objective of this paper to establish which opinion is accurate; rather, the author wants to acknowledge the inextricable relationship between sound and movement, and state that this association is an intrinsic element of the *SonicJumper*.

3.3 Generating and Saving Musical Materials

Composers use different terminology to explain the earliest activity leading to an original work. Some may refer to a formulaic approach. Others might describe a rigorous pre-compositional process. Yet, other composers talk about the fruits of noodling on the piano, or improvising. What is happening, at this early stage, is the initial concretization of creative thought in the form of musical materials. The result is often a manuscript of some sort. Rigorous planning—much formulaic designing—goes into establishing the "voice" and "action" of the *SonicJumper*, before beginning to generate materials. Voice is akin to the controller's synthesis engine, while action is a result of mapping. Once voice and action are established, the user produces and digests sound in the manner described above, making decisions on-the-fly based on musical intelligence and intuition. The Max/MSP patch records both voice and action data, using a standard audio file format. In this way, the sounds of the voice are audio files, while action data more resembles a wavetable—one for each sensor output. One consequence of recording action data, is that the user is able to take "snapshots" of a particular movement. Then, the data can be used to duplicate signal processing (i.e. granular synthesis) on several different audio files, without the user having to set up the *SonicJumper* controller. This is comparable to transforming themes and harmonies via a 12-tone row table. The voice data—actual audio files—is transcribed into traditional musical notation either using the composer's ear or via computer-assisted compositional software (i.e. AudioSculpt by Niels Bogaards and others; OpenMusic by Gérard Assayag and Carlos Agon). It would be interesting to draw a comparison to other modes of composing. In some respects, *SonicJumper* composing is not far removed from traditional approaches, such as working out material while sitting at the piano.

4. CASE STUDY

In the fall of 2005, D. Andrew Stewart was commissioned to create a work for the Dutch ROSA Ensemble—tenor saxophone; electric guitar; bass guitar; piano; percussion; processed audio and live-audio streaming. Musical material for both pitch organization and processed audio was generated with the *SonicJumper*—based on sampled audio from early 1970s funk music. The composer set himself the task of 'entering' the SOUND WORLD of funk music, without necessarily evoking the funk idiom. It is important to point out that entering the funk sound took place at the earliest stage of composition (generating materials with the *SonicJumper*). The performances of the work remained in a similar sound world. The result, therefore, was a piece of music with a sense of whole, from it's inception to its realization.

5. CONCLUSION

There is no greater joy during the compositional process than to realize you have successfully captured what is in your head, in your ear or that which your creative spirit compels you to say. Indeed, composers make great effort over an entire lifetime—often unsuccessfully—to manifest their true musical thoughts in an aural form. The challenge is immense; many fail because there is no exact music to capture an artist's thought or feeling. On the other hand, we find ourselves in a unique position of being able to seize certain modes of communication, for the first time. Technology that can catch, examine and reproduce gesture brings us a few steps closer to tapping into learned and unconscious behaviour. If a composer is willing to use technology, there is a strong argument for the use of gestural acquisition for communicating creative thought; or at least, one is able to examine the relationship between gesture and creative impulse. In the early days of analog studio composition, Hugh LeCaine describes an interaction where the studio composer has intimate control of the musical outcome—the composer is closer to sound. [8] The *SonicJumper* reexamines the notion of proximity. Not only does the jumper bring its user nearer to a desired sonic result, it also allows for immediate realization of the creative impulse. If one could derive a maxim concerning gesture and creativity, the statement would go far in forwarding the idea that gestural controllers can be successfully woven into a musical fabric, such that the technological aspect is far less significant than the overall musical experience.

6. REFERENCES

[1] Stewart, D. Andrew, composer. "Stewart Sonic Jumper," home page <www3.sympatico.ca/andrew.stewart/pages/sj/jumper.html> March, 2005.

[2] Stuck, Les and Richard Dudas, engineers, Cycling74. "granularized," Max/MSP patch designed for granular synthesis, 2003.

[3] Mulder, Axel G. E.. " Towards a choice of gestural constraints for instrumental performers," Trends in Gestural Control of Music CD-ROM, eds. Marcelo M. Wanderley and Marc Battier. Paris: Ircam—Centre Pompidou, 2000.

[4] Vines, Bradley W. and others. "Cross-Modal Interactions in the Perception of Musical Performance," This article was submitted in partial fulfillment of the requirements for the Ph.D. in Experimental Psychology from McGill University by the first author (McGill University, April, 2005).

[5] Seitz, Jay A. "Dalcroze, the body, movement and musicality," Psychology of Music, 33(4), 2005.

[6] Ibid

[7] Ibid

[8] LeCaine, Hugh. "Synthetic Means", The Modern Composer and His World, eds. John Beckwith and Udo Kasemets. Toronto: University of Toronto, 1961.

Towards a catalog and software library of mapping methods

Hans-Christoph Steiner
at.or.at
New York, NY, USA
hans@at.or.at

ABSTRACT

Mapping has been discussed for decades, yet there is not standard catalog of mapping methods. The Mapping Library for Pd is a fledgling library of mapping primitives with the aim of cataloging existing mapping methods. Also included are techniques for conditioning sensor data to make it usable in the context of instrument design.

1. INTRODUCTION

> Everything should be made as simple as possible - but no simpler. - Albert Einstein

Mapping for instruments has been discussed for decades. There have been a huge range of ideas touted, and many instruments built and tried. Shared elements and ideas are repeated, and re-implemented again and again. The foundations of audio have been long since codified with the standard unit generators, with few audio software packages disregarding them. They have become a basic language to express ideas in that realm, whether in software or hardware. There is no catalog of fundamental mapping algorithms, and little work has been done to build the foundations into software.

With no standard framework, instrument designers are constantly reinventing the wheel, re-implementing mapping algorithms whenever creating a new instrument. Many mapping operations are very common, so it makes sense to have a software library that includes these operations. Even for uncommon and more complex operations, having a library of mapping methods allows the instrument designer to rapidly test a wide variety of mapping ideas without having to implement them, and even derive some inspiration from scanning the contents of a mapping library and rapidly interchanging elements.

In a similar vein, there are a myriad of methods of conditioning sensor data. When using sensors, there is often noise and errant results. Many useful sensors do not produce linear data, or even easily modeled curves. There are many techniques for conditioning this data to make it straightforward to use. These techniques are well established in the realm of electronics [8] [12]. Many of these ideas are very useful in the context of instrument design and can be applied in the software realm. Yet there seems to be no catalog of software algorithms for conditioning, especially when talking about applying them to instrument design.

This paper aims to start the discussion of what these mapping primitives are and describes work towards building a library of these ideas: the Mapping Library for Pd. Algorithms for processing sensor data are included because many of them are also used in mapping data, such as applying transfer functions to shape the data. This library is also intended to become a catalog of mapping techniques which can freely be implemented in other ways, with the hope of developing some standardized terminology.

2. PREVIOUS WORK

I started my work on the topic of mapping with the [hid] toolkit [9], which mainly focused on streamlining the process of getting data from game controllers for creating instruments. The [hid][1] object already provides access to a wide range of devices. In addition, there is work underway on supporting sensorboxes in a way that follows standards laid out by the [hid] toolkit objects. The other key part of that work is the objects for mapping data. Cyrille Henry, who is a major contributor to the software library, had developed a collection of Pd objects for sensor processing techniques. It is these sets of objects that the Mapping Library for Pd is based on.

There have been a couple of attempts at building frameworks for creating musical instrument mappings. Two notable packages come from IRCAM. "MnM is a set of Max/MSP externals... providing a unified framework for various techniques of classification, recognition and mapping for motion capture data, sound and music." [2] An earlier attempt from IRCAM is the ESCHER toolkit for jMax [11] which is a set of objects to address various problems of mapping. Goudeseune presented his mapping technique based on high dimensional interpolation he calls "simplicial interpolation" [4] . While this is an interesting technique that shows promise, it only addresses a specific approach to mapping and is not broken up into more generally useful modules. Both of these are complex systems which show promising results, but they seem to address the opposite end of the spectrum that this paper is addressing. They are built from complicated objects and take a mathematical approach to mapping. While

[1] a word in square brackets denotes a Pd object

mathematics are an integral part of mapping, the user need not experience it in that way.

3. DESIGN IDEAS

When talking about mapping, we are almost always talking about computer software, anything from custom C code to Pd patches to application preferences. Therefore it makes sense to implement a catalog of mapping methods as a software library. Mapping is generally represented firmly within the realm of mathematics. By abstracting the math into software objects, mapping can be approached as a system of logic. Software derives its vast power from the encapsulation of ideas and the reuse of code. Many complex mapping algorithms can be encapsulated into software objects, opening up new opportunities for exploration. One need not understand much about an algorithm within an object in order to insert it into a program and play with it. This way of interacting is much more like how many musicians learn to play an instrument: they play with it and see what they can figure out. Having encapsulated software objects, mapping can be more in this spirit of play, rather than purely a separate, studied effort. Also, catalog of mapping primitives aids in teaching and standardized terminology allows more fluid discussion and exchange of mapping ideas.

This paper covers explicitly defined mapping methods, where the instrument designer directly controls each aspect of the mapping. Other papers cover methods involving generative techniques or neural networks[3]. While such systems might be based on the same mapping primitives as other methods of mapping, it is difficult to derive the mapping since it is only represented internally to the map-generating process.

There is still a lack of a set of commonly agreed upon primitives for the building of mappings. There are many great ideas about mapping, but lack reusable implementations. The goal of the Mapping Library for Pd is to first provide a set of mapping primitives, then to build more complex objects using the primitives. This modular approach has a number of advantages: the code is easy to read since its based on encapsulated ideas, code can be easily tweaked or repurposed since its all written within Pd, and as more objects get written, it becomes easier to write higher level objects. These objects be general, making as few assumptions as possible for how they should be used. This idea is key to the design of Pd itself. TCP/IP was famously designed this way, and it has proven itself to be useful in ways far beyond the original creators' intentions.

In some ways, this library is a return to basics. Many interesting yet complex methods of mapping have been proposed and discussed. It seems that its too soon to be moving onto such complex methods when the basics are not clearly established. Software has become an integral part of designing new instruments, and it is rare for a new instrument these days to have absolutely no software component. Mapping should then be codified in software beyond being written about. Not only is software functional, but it is also a highly effective method of communicating the ideas related to mapping, arguably more effective that written language. There are numerous clearly defined ideas in audio synthesis which are implemented in most computer music software these days. Much how the standard audio unit generators encapsulate the mathematics used in synthesizing audio, a mapping ideas should also be laid out and encapsulated.

As with the [hid] toolkit, the Mapping Library uses the data range of 0 to 1 wherever possible for the reasons outlined in the paper on that software. Almost all of the mapping objects expect input and output data in the same range. This range is applied everywhere wherever possible, including to somethings that might disturb mathematicians, like [polar], which converts cartesian coordinates to polar coordinates. Even the angle is scaled to the range of 0 to 1. This allows other mapping objects to be used after the conversion without having to rescale the data. Though the objects are designed to work within this range, many of them also work beyond that range. For example, [spiral] has an infinite range, with 1 representing one full rotation.

Another realm of mapping which has not yet mentioned are the issues of processing sensor data to make it usable. Sensors are subject to noise, errant glitches, and unique properties which make them difficult to use. Like the world of instrument mapping, there is not a standard catalog of sensor primitives in the realm of software. Another promising area of research is data stream processing. The Stanford STREAM group has created a standard query language for data streams[10]. This language is oriented towards typical database applications, but many of their techniques could be useful in processing sensor data stream. A key part of the Mapping Library effort is to create software that encapsulates these techniques and make them usable to instrument designers.

4. EXAMPLES OF MAPPING OBJECTS

There are already quite a few objects implemented, here are some selected examples: control rate filters ([iir], [fir], [mpfilter], [lop]); basic transfer functions ([curve], [curve_power], [gaussian]); interpolation ([wave], [interpolate]); sensor data conditioning ([hysteresis], [local_min], [local_max]); testing ([test_n], [box], [stream_presense]); and, ranging and sizing ([resample], [upsample], [downsample], [clip], [distance]). At the most basic level, the Mapping Library includes objects for generating various curves over a range. [curve] is an object that provides a continuum of curves starting from 0 for linear. For positive numbers, the curve is a power curve, for negative numbers, its a root curve. For people who want to use standard functions, [gaussian], [sinusoid], [curve_power], [curve_root], [curve_exp], and [curve_log] are provided. Scaling is another common operation, so there is [autoscale] which dynamically scales input data to an output range, [notescale] which scales 0 to 1 to the specified range of integer note values. Other ranging objects include [local_min], [local_max], [min_n], and [max_n]. Objects with the "_n" suffix mean that they take an numeric argument which control how many previous values that object should consider (i.e. apply the function to n elements). Averaging

The mapping objects are built from the ground up from quite primitive operations into higher level objects. For example, the [spiral] object is built using the [polar] object, which is in turn built using the [vector] object, which uses [radians-¿mapping]. The objects names have been carefully chosen to accurately represent the idea with a minimum of confusion. Commonly used words for certain methods were adopted wherever possible, for example with [diverge] for one-to-many and [converge] for many-to-one since these are words commonly used to describe such mappings [5]. Some unusual words are used, like [disjoin], because it makes sense with its opposite operation: [join]. This is just a fledgling

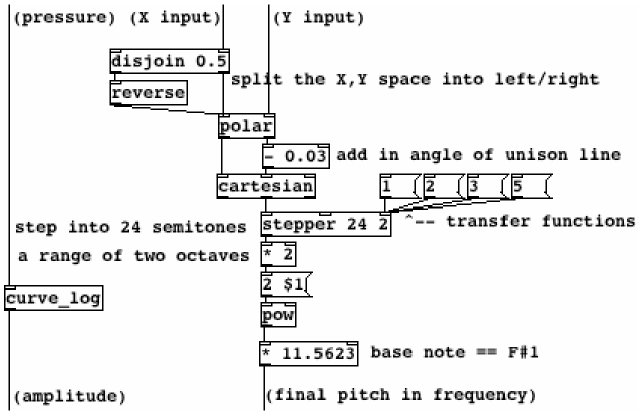


Figure 1: A Pd patch showing the mapping of The Ski's pitched mode.

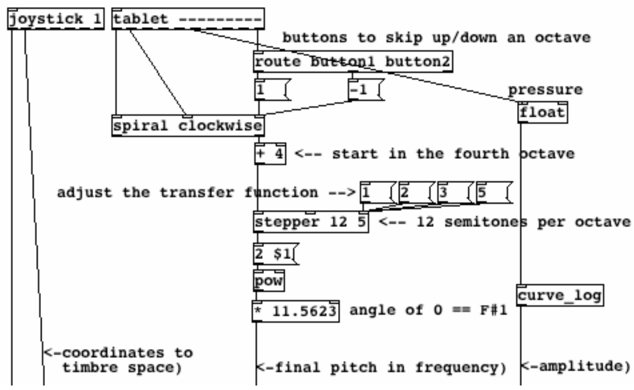


Figure 2: A Pd patch showing the mapping of the Voicer.

effort, so the names are bound to change as things develop.

5. BREAKDOWN OF EXISTING MAPPINGS

To explore these ideas, the mappings of two specific instruments are analyzed and reproduced using the Mapping Library for Pd. There is a huge variety of new instruments and wide range of ideas for mapping. These instruments where chosen because they have been played extensively, performed in concert, and each instrumentalist has achieved a high level of fluency with his instrument. Instruments that are regularly played will have a more honed mapping, and the spotlight of performance is excellent at drawing exposing problems.

5.1 The Ski Angular Mode

With Huott's Ski [6], he outlines four different modes for mapping the tactex pads to controlling samples: linear, polar, angular, and linear velocity. All of these can be easily implemented using exisiting mapping objects. For example, here is how to implement angular mode. First, [autoscale] automatically scales in the incoming data to a floating point range of 0 to 1. The scaled cartesian coordinates from the tactex pad are then fed to the [polar] object, which converts the data to a magnitude and angle. The angle is output in the range of 0 to 1 instead of the more usual -π to π. The allows other mapping objects to be easily chained after the [polar] object.

5.2 The Ski Pitched Mode

The front pads are used in a pitched mode, this layout is diagramed in Figure 4 of Huott 2002. First, the pads are split into left and right sides using [disjoin]. The left side is reversed using [reverse]. The cartesian coordinates are converted to polar, then the angle of the unison line is created by subtracting 0.03 from the angle (represented from 0 to 1 not -π to π). The data is then converted back to a rotated cartesian plane, and the Y value is taken to control frequency. The range is split into 24 semitones and a x^2 transfer function is applied to allow glissando between notes. The range is multiplied by 2 to span 2 octaves, then converted to frequency with a base note of $F^{\#}1$. The pressure data is curved logarithmically to control amplitude.

5.3 The Voicer

Kessous developed his Voicer [7] using a tablet and a joystick as controllers. The tablet controls the pitch and amplitude while the joystick navigates a timbre space of vowel sounds. Pitch is derived from the polar angle, in a spiral. The [spiral] object does this using [polar] to convert to polar coordinates, then it keeps track of rotations. Since the [spiral] object has a "clockwise" argument in the patch, the data increases in the clockwise direction, rather than counter-clockwise as is usual with polar coordinates. [stepper] converts the linear angle data from [spiral clockwise] into a stepped line. The first argument of 12 creates 12 steps within the range of 0 to 1. Each step locally curved according to the transfer function specified by the last argument/inlet. This curved by taking the input and raising it to the power specified by the transfer function argument/inlet. The output of [stepper] is then converted into frequency values, with an angle of 0 equal to the beginning of the $F^{\#}4$ segment. In Arfib, et at, 2002, Figure 5 displays a graph of the output of the [stepper] object. It is built using other Mapping Library objects: [segment], [curve_power], and [desegment]. [segment] is in turn built using [disjoin], and [desegment] is built using [join]. For the amplitude control, the pressure in taken from the pen using the [tablet] object, which outputs all axes in the range of 0 to 1. The pressure data is then curved logarithmically using [curve_log], to match the human perception of amplitude.

6. CONCLUSION

The process of analyzing these two instruments has affirmed many of the existing ideas in the Mapping Library for Pd, and has exposed a number of weaknesses and omissions. Many interesting approaches to mapping, such as a multi-layered approach discussed by Arfib[1], Kessous, Wanderley, do not seem inherently compatible with these current objects. This is just the beginning, but these two examples clearly demonstrate that there is promise to this approach to mapping. With a flexible toolbox of mapping methods, instrument designers can experiment more fluidly with mapping ideas. With a catalog of mapping methods, we can more easily discuss new mapping ideas and demonstrate them using code.

7. FUTURE WORK

Now that the basic foundation has been laid, a more objects will be created to work towards completing a catalog of sensor processing and mapping methods. Also, following up on the [hid] toolkit, we aim to create a framework for working with raw sensors and sensor boxes, and making them interoperate with game controllers and the Mapping Library for Pd. This will lead away from a focus on music and hopefully towards a more generally useful library, contributing towards visual instruments, mapping for interactive installations, robotics, or whatever needs data mapped to controls.

8. ACKNOWLEDGMENTS

Cyrille Henry contributed in a number of key ways. He was involved in the formations of the original ideas, and helped me focus my ideas through discussions about mapping, and wrote a wealth of objects for the Mapping Library.

9. REFERENCES

[1] D. Arfib, M. Couturier, L. Kessous, and V. Verfaille. Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces. *Organised Sound*, 7(2):127–144, 2002.

[2] F. Bevilacqua, R. Muller, and N. Schnell. MnM: a Max/MSP mapping toolbox. In *Proc. of the Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, Canada, 2005.

[3] A. Cont, T. Coduys, and C. Henry. Real-time gesture mapping in pd environment using neural networks. In *Proc. of the Conference on New Interfaces for Musical Expression (NIME04)*, Hamamatsu, Japan, 2004.

[4] C. Goudeseune. Interpolated mappings for musical instruments. *Organised Sound*, 7(2):85–96, 2002.

[5] A. Hunt and M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(2):97–108, 2002.

[6] R. Huott. An interface for precise musical control. In *Proc. of the Conference on New Interfaces for Musical Expression (NIME02)*, Dublin, Ireland, 2002.

[7] L. Kessous. Bi-manual mapping experimentation, with angular fundamental frequency control and sound color navigation. In *Proc. of the Conference on New Interfaces for Musical Expression (NIME02)*, Dublin, Ireland, 2002.

[8] R. Pallàs-Areny and J. G. Webster. *Sensors and Signal Conditioning*. John Wiley and Sons, 2nd edition, 2001.

[9] H.-C. Steiner. [hid] toolkit: a unified framework for instrument design. In *Proc. of the Conference on New Interfaces for Musical Expression (NIME05)*, 2005.

[10] The STREAM Group. Stream: The stanford stream data manager. *IEEE Data Engineering Bulletin*, 26(1), March 2003.

[11] M. Wanderley, N. Schnell, and J. B. Rovan. Escher-modeling and performing composed instruments in real-time. *IEEE Systems, Man, and Cybernetics*, 1998. `http://intl.ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=727836`.

[12] J. S. Wilson, editor. *Sensor Technology Handbook*. Elsevier, 2005.

LINE: Interactive Sound and Light Installation

Daisuke Kobori†1, Kojiro Kagawa†1, Makoto Iida†2, Chuichi Arakawa†1

†1Graduate School of Interdisciplinary Information Studies, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo
113-8656 Japan
+81-3-5841-6438

†2School of Engineering, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo
113-8656 Japan
+83-3-5841-6438

{daisken, kobujiro, iida, arakawa}@cfdl.t.u-tokyo.ac.jp

ABSTRACT
LINE is an interactive installation that enables a synchronous expression of sound and light. Audiences can simultaneously control sound and a three-dimensional light object which appears in a cylindrical tank of water. By moving a hand-held control device in the air, audiences can experience a harmonious expression of sound and light.

Keywords
Sound and light relationships, three-dimensional light object, arm gestures, interactive installation.

1. INTRODUCTION
In the field of classical music, musical harmony has been researched on a systematic and theoretical basis. Here, certain aesthetic standards have been derived from the ratio in frequencies of sounds [1]. The first goal of this project is to link such harmony of sounds with visual harmony expressed by lights, colors, etc.

We aimed to create two types of expressions. The first is an interactive exhibit using integrated light and sound that have corresponding parameters. The other is an expression where the audiences can visually experience harmony of sounds.

On another note, in the field of Interactive Installation and Media Art, there have been various works with simultaneous audiovisual expression that use projectors. Previous works of interactive audiovisual installation that use projectors include the series of works by Golan Levin [4, 5], and "Piano – as image media" by Toshio Iwai [2].

However, since projectors themselves are originally made for the purpose of projecting images on a flat screen, such visual expressions naturally remain two-dimensional ones. Another goal of this project is to create a three-dimensional expression using a projector, which is generally versatile and easy to control.

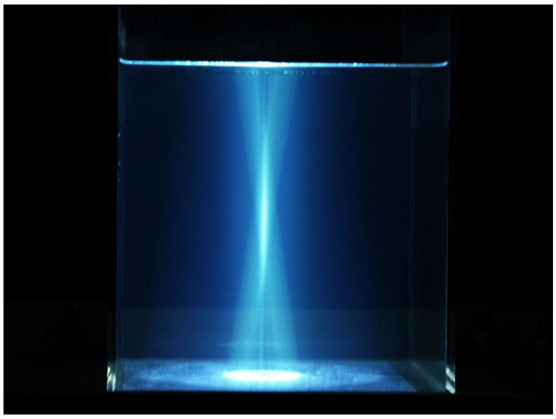


Figure 1. Three-dimensional Light Object

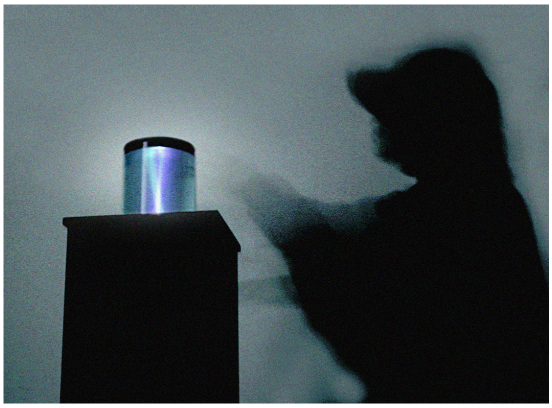


Figure 2. The sideview of LINE

Therefore, we examined and developed a set of original software and hardware: the LINE system. LINE's hardware enables the generation of a three-dimensional light object using two-dimensional images emitted by projectors (Figure 1). LINE's software processes the input from a hand-held control device, and then carries out sound synthesis and generation of two-dimensional images corresponding to the input. As result, this system enables an interaction with both sound and the three-dimensional light object (Figure 2 and Figure 3).

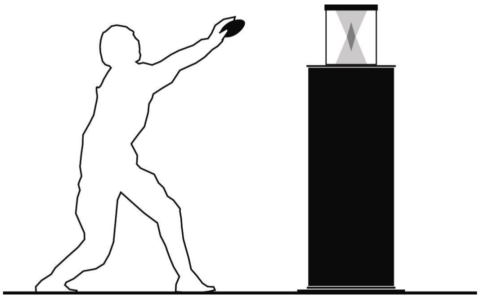


Figure 3. Audience operating LINE

2. HARDWARE IMPLEMENTATION

The LINE system outputs multiple light beams from an LCD projector that creates a three-dimensional light object. As shown in Figure 4, the space where the light beams intersect becomes relatively brighter, thus creating a light object. The light object appears to illuminate in space where the beams intersect, with light beams flaring vertically outward from its center.

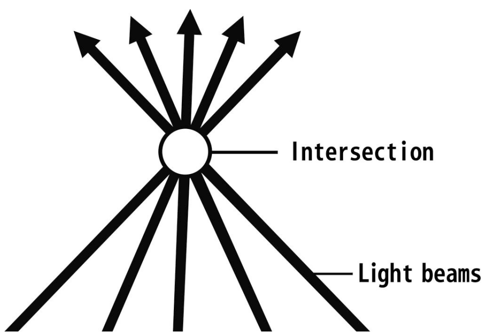


Figure 4. The basic concept of this system

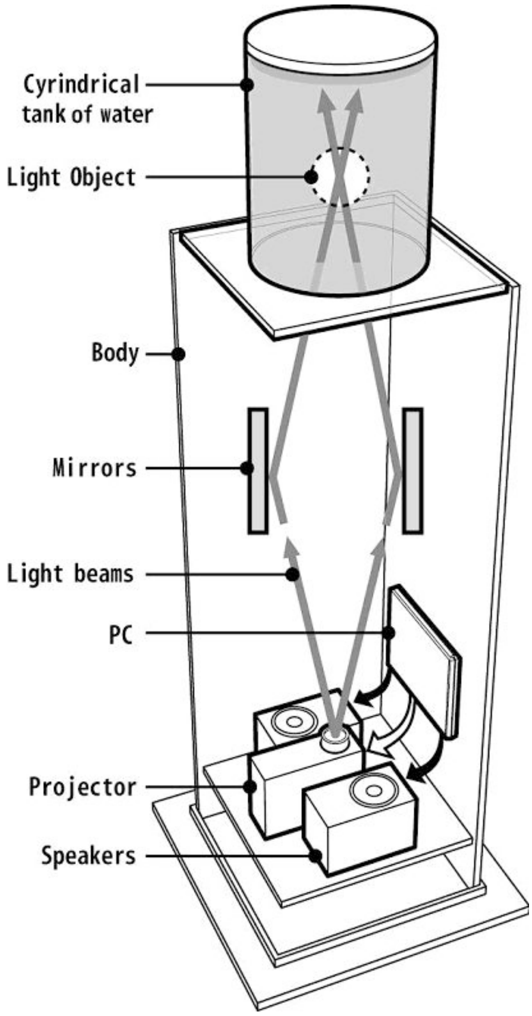


Figure 5. Hardware diagram

Figure 5 shows the hardware diagram of this system. The hardware consists of a body (450mm wide x 1200mm high), an LCD projector, mirrors, a cylindrical tank filled with water, a PC, and speakers. A SHARP PG-B10S projector is used.

The PC (a Windows machine) processes the input from the control device and outputs image data to the projector and sound data to the speakers. The projector is set at the bottom of the body with its lens pointing upward, projecting two-dimensional images that creates the light beams.

The courses of the beams are altered by the mirrors as they are projected upward. The beams intersect at a given space inside the water tank set on top of the body, creating a light object. Water soluble dye was mixed in the water to increase the visibility of the light object. Also, the two-dimensional images emitted from the projector can be manipulated to adjust each of the light beams, thus enabling the system to control the size, color, and position of the light object, as shown in Figure 6.

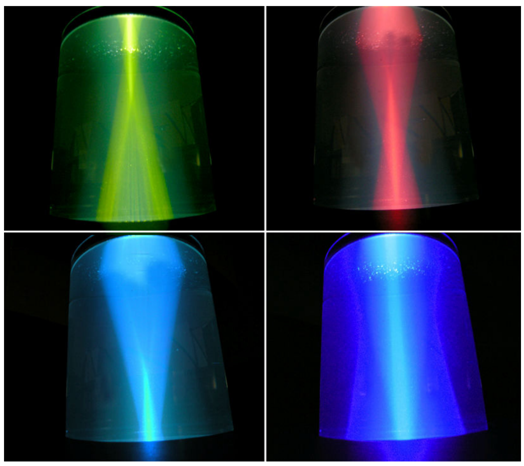


Figure 6. Variations of Light Object

3. SOFTWARE IMPLEMENTATION

Three main functions constitute the LINE's software:

The first function is that it processes the input from the control device. Audiences holding this control device can control the sound and the light object just by swinging their arm.

The second function is that it carries out sound synthesis. Every beam corresponds to 1 synthesizer. This synthesizer is a simple 2 operators FM synthesizer implemented using the real-time audio I/O C++ classes of Gary Scavone's RtAudio [6, 7], and its parameters can be modulated independently.

The third function is that it generates two-dimensional images. As shown in Figure 7, this system generates quite simple images, which are emitted using the projector as beams. Each source of the beams corresponds to 1 synthesizer, and their colors can be controlled individually.

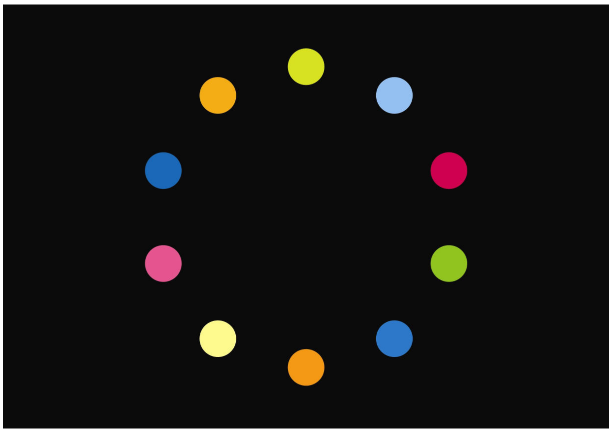


Figure 7. An image generated by LINE's software

3.1 Interaction using Arm Gestures

The control device uses a gyro-sensor and converts the input from the arm gestures to mouse pointer values, X and Y. The software measures the change in the values of X and Y in a given time, and carries out sound synthesis and image generation according to the mappings shown later.

When the audience holding the control device swings up his arm in a large and quick motion, the light object and sound emerges or vanishes, as with a lamp.

If the audience slowly moves his arm up and down when the light object is visible, the light object moves vertically along with the arm's motion. Likewise, a slow, horizontal motion of the control device changes the light object's size (See also Table 1 and Figure 8).

Table 1. Light Object and arm gestures correspondence

Light Object Properties	Arm Gestures Properties
position	vertical motion (Y axis)
size	horizontal motion (X axis)

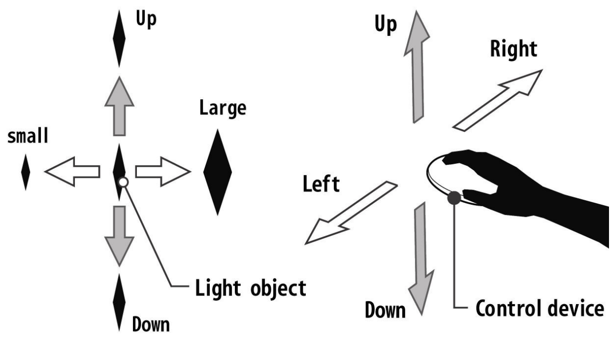


Figure 8. Light object is controlled by arm gestures

3.2 Mappings between Synthesizer and Beam

Mappings between synthesizer's parameters and beam's parameters are as given in Table 2.

Table 2. Mappings between Synthesizer and Beam

Synthesizer Properties	Beam Properties
frequency	color
harmonics	size
amplitude	brightness

The synthesizer's frequency corresponds to the beam's color. Using hue and saturation in terms of HSB color space [8], precise correspondence between frequency and color is represented in the following equation:

$$freq = 20 \times 2^{\lfloor 10s \rfloor + h}$$

where *freq* is frequency (in Hz),

s is saturation, $0 \leq s \leq 1.0$,

h is hue, $0 \leq h \leq 1.0$

The brackets surrounding $10s$ represents a floor function. Thus, if we increase the synthesizer's frequency from 100Hz to 1000Hz continuously, the beam's saturation increases while its hue returns to the same value as at 100Hz at the time of 200Hz, 400Hz and 800Hz.

Frequency is made to correspond to hue and saturation so that the harmony of sound and that of light can be attained simultaneously, using the similarities in the characteristics of visual and acoustic senses. When the frequency doubles, the resulting sound is recognized as being in the same "pitch class" as the original sound. At the same time, colors can be presented in a "hue circle", where colors moving across a spectrum circulate, eventually returning to its original color [8]. Also, by using saturation to express the general bandwidth of the frequency, we are able to avoid the colors at, for example, 100Hz and 200Hz from having exactly the same colors.

The synthesizer's harmonics correspond to the beam's size. If we increase the beam's size, we hear resonant sound derived from the increase in FM modulation index. Thus horizontal motion of the control device changes harmonics of sounds as with the size of light object made up by the beams.

The synthesizer's amplitude corresponds to the beam's brightness. If we increase the beam's brightness, the synthesizer's sound becomes louder.

3.3 Rule for Harmonious Sound

As time progresses, each synthesizer's frequency changes autonomously according to the following rule. Each synthesizer compares its frequency to that of a neighboring synthesizer. Unless the ratio of the two is in harmonious relationship, where the ratio of the frequencies is in simple integer such as 2 or 3, the synthesizer's frequency changes.

For instance, if this rule is applied to an initial state where each synthesizer's frequency is set at random, some of them that are in harmonious relationships are gradually collected together, until eventually every ratio of the synthesizer's frequencies is in harmonious relationship (See also Figure 9). Also, as the frequencies change, the beam's color changes according to the mapping mentioned above.

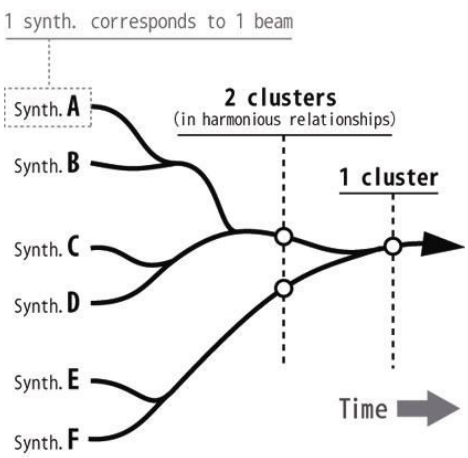


Figure 9. Clustering process

4. EXHIBITION

This installation was exhibited for 3 days during iii Exhibition 4 [3]. It was exhibited using a space approximately 2.5 meters x 2.5 meters surrounded by a curtain.

5. CONCLUSION

We present here an expressive installation that enables an audience to interact with integrated sound and light. We developed a hardware that generates a three-dimensional light object and a software that processes the control device's input to generate sound and light.

As for the relationship between sound and light, when the sounds were split into two or three clusters in frequencies, we were able to express the light in two or three clusters as well. As a result, we were able to express a mutually complementary relationship of auditory and visual experiences.

Although the light object only move vertically at the present time, it is possible that the light object will be able to move horizontally as well as three-dimensionally in theoretical.

In the future, we will further enhance the expression of this installation by adding horizontal and three-dimensional motion to the light object.

6. ACKNOWLEGEMENTS

The authors wish to acknowledge the assistance of Taro Suzuki, Seiichiro Matsumura and the Arakawa Lab members of Tokyo University, and also Daisuke Kirihara of Keio University.

7. REFERENCES

[1] Department of Acoustic Design – Kyushu Institute of Design. *Introduction to Acoustic Design – Sound, Music, Technology*. University of Kyushu Press. 2001.

[2] Iwai, T. Piano – as image media. http://ns05.iamas.ac.jp/~iwai/artworks/piano.html

[3] Kagawa, K. and Kobori, D. SLIT – Sound, Light, InTegrated. https://www.iii.u-tokyo.ac.jp/i3e4/iii4.html

[4] Levin, G. and Lieberman, Z. Sounds from Shapes: Audiovisual Performance with Hand Silhouette Contours in *The Manual Input Sessions*. *5th International Conference on New Interfaces for Musical Expression (NIME05)*. pp. 115-120. 2005.

[5] Levin, G. An Audiovisual Environment Suite. http://acg.media.mit.edu/people/golan/aves/

[6] Scavone, P., G. RtAudio: A Cross-Platform C++ Class for Realtime Audio Input / Output. *International Computer Music Conference, Göteborg, Sweden*. 2002.

[7] Scavone, P., G. RtAudio: a realtime audio i/o C++ classes. http://www.music.mcgill.ca/~gary/rtaudio/

[8] The Color Science Association of Japan. *Color Science Handbook, revised edition*. University of Tokyo Press. 1998.

[9] The Illuminating Engineering Institute of Japan. *Lighting Handbook, The second edition version*. Ohmsha. 2003.

[10] Tsujiuchi, J., Oki, H., Kojima, T., Kuroda, K. and Kawada, S. *Optical Engineering Handbook*. Asakura Shoten. 2002.

Decay in Collaborative Music Making

N. Bryan-Kinns
Interaction, Media, and Communication Group
Department of Computer Science
Queen Mary, University of London
Mile End, London. E1 4NS. UK
nickbk@dcs.qmul.ac.uk

P. G. T. Healey
Interaction, Media, and Communication Group
Department of Computer Science
Queen Mary, University of London
Mile End, London. E1 4NS. UK
ph@dcs.qmul.ac.uk

ABSTRACT

This paper reports on ongoing studies of the design and use of support for remote group music making. In this paper we outline the initial findings of a recent study focusing on the function of *decay* of contributions in collaborative music making. Findings indicate that persistent contributions lend themselves to individual musical composition and learning novel interfaces, whilst contributions that quickly decay engender a more focused musical interaction in experienced participants.

Keywords

Music improvisation, creativity, group interaction, design.

1. INTRODUCTION

Music making is inherently a social activity, yet a vast majority of our musical instruments are designed to be used by individuals. Such instruments rely on physical proximity to foster group music making i.e. being in the same space as others. Whilst we have embraced new forms of group verbal communication such as text messaging on mobile phones, for many of us informal and pervasive group music making has lost its place as an everyday form of social interaction [6]. Reviews of support for music collaboration [1] indicate that remote group music making is a field ripe for exploration yet there is little work investigating support for remote group making, let alone exploring what it might mean to engage in such activities.

We believe that we can start to design richer and more satisfying musical experiences by understanding what encourages and supports *mutual engagement* between people as they interact with each other. We characterize mutual engagement as points at which participants start to play with, and explore their interaction with others over and above the mediating devices involved. At these points participants start to rely on their shared beliefs and understandings of what is going on, what might happen, and who might do what in the interaction. Such points are crucial in group music making which relies heavily on both shared expectation and experimentation. Indeed, group music making is a pertinent example of a basic form of group creativity which has many parallels to normal verbal conversation including being multimodal and co-present.

The rest of the paper is organized as follows: First the design of a novel group music making tool is outlined, then a study of the effect of persistence of contribution has on mutual engagement is detailed followed by discussion of the emergent patterns of use. Finally the paper is concluded with some discussion about implications for the design of group music making tools.

2. DESIGN

Daisyphone is an on going design project [2, 3] whose aim is to support remote group music making. By remote *music making* we refer to the form of musical interaction which is somewhere between improvisation and composition; participants can jam together and yet the resultant music is persistent and editable. In previous work we identified four design features which we believe contribute to the support of mutually engaging collaborations and which we employed in the design of Daisyphone: **Localization** within the artifact being co-produced; **Mutual awareness** of actions; **Shared and consistent** representations; **Mutual modifiability** of contributions.

The Daisyphone user interface is illustrated in figure 1. Notes are lower in pitch towards the edge of the circle. As the grey arm rotates clockwise, the notes underneath are played, so each of the spokes represents notes played at the same time. Hues of notes indicate who contributed them (this provides *mutual awareness* of actions), and intensity of color represents the volume of the note. Different shapes represent different instruments including piano (circle), and percussion (diamond). Volume and instrument are modally controlled from the four central spokes.

In Daisyphone's current form up to 10 remote participants can create and edit a short shared loop of music semi-synchronously – typically updates take under one second to be shared. This provides support for a form of remote group music making whilst requiring little network bandwidth. As with other remote group making tools such as WebDrum [4], Daisyphone works by clients sharing indications of musical contributions *via* a central server through the internet so providing a *shared and consistent* representation of musical loops being constructed. There is no ownership in Daisyphone – people can edit each others' notes and play the same instruments supporting *mutual modifiability*. As well as sharing musical contributions, Daisyphone also shares graphical annotations on and around the music composition space; drawing occurs whenever the mouse button is pressed which results in a 'messy' form of interaction. This annotation is intended to support both *localization* within the composition, and social and discursive exchanges

Previous studies [2, 3] have identified several design issues with Daisyphone and its support for group creativity. In this paper we explore the nature of persistence in contributions. In pervious versions of Daisyphone all contributions were persistent. It quickly became clear through studies that participants did not clear up after themselves and the musical space quickly became saturated with notes which created a cacophonous noise. The ability to move to new, clean, sessions was one of the first design developments and resulted in the circular session selector illustrated in the top left of figure 1. However, further studies showed that this still was not sufficient to promote flowing and engaging musical interaction – participants simply got bogged down in a sea of contributed notes. In order to investigate the effect of persistence of musical contribution, a new version of Daisyphone in which notes disappear was developed (referred to as the **decay** version). Only the notes are transient, therefore the graphical annotation created when the notes are contributed remain providing some visual cues to the contributions (a form of history of contribition). The rate of decay of the notes is critical to the design – too quick and coherent sharing of music will not occur given the semi-synchronous nature of the infrastructure; too slow and the musical space will continue to became overcrowded. For the studies here, decay is created by halving the volume of notes every time the arm passes over them. This typically gives 3 plays of a loud note before it disappears which appears from initial studies to be sufficient for co-ordination.

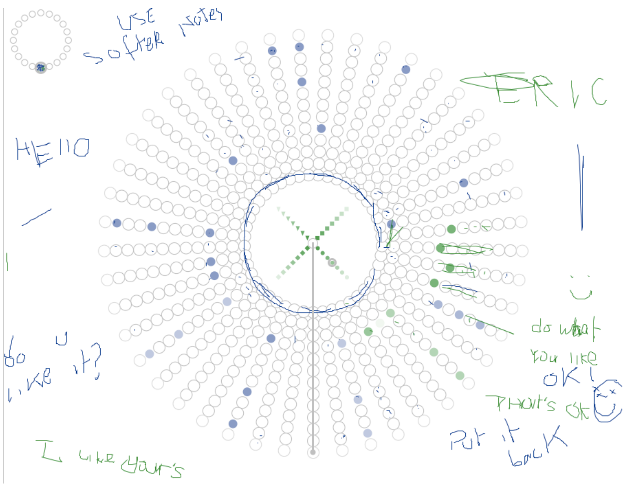


Figure 1: Daisyphone interface

3. STUDY

The aims of this study are twofold: 1) To investigate the effect decay of contributions has on remote group creativity in music; 2) To further explore the nature of remote group music making in general.

3.1 Format

Ten post graduate students studying Advanced Methods in Computer Science at the authors' institution were set a piece of coursework in which they were asked to:

- Use both the persistent and decay versions of Daisyphone to remotely create music together over three weeks.
- Perform their piece of music for the rest of the group.
- Analyze and report on the interaction that took place in Daisyphone in both versions.

The students grouped themselves into 3 groups. They had a wide range of musical ability from novice to proficient musicians playing in bands. None had ever used a tool like Daisyphone before.

Participants were asked to report on whether, and how, they experienced flow as a group [7]. We also asked the participants to identify points of *attunement* between each other on a three point scale: **Acknowledgement** –they were aware of the contributions of another; **Mirroring** – they mirrored, or reflected, others' contributions; **Transformation** – they transformed others' contributions (indicating a high level of mutual engagement). These reports and ensuing discussions are used in the rest of this paper to help make sense of the observed behavior. Flow was categorized in terms of: Chance of completing the activity; Ability to concentrate on what they're doing as a group due to clear goals and adequate feedback; Deep, effortless involvement with a reduction in concern for external factors; Sense of control over actions; Transformation of time.

Additionally, logs of all actions in Daisyphone were stored for later re-play and analysis.

4. PATTERNS OF USE

This section outlines the patterns of use and behavior that took place in the study with the persistent and decay versions of Daisyphone. Initial analysis of logs are presented here – detailed analysis is currently being undertaken. An average of 8 sessions with the persistent version and 3 sessions with the decay version were recorded for each group. Each session lasted on average 16 minutes for the persistent version and 12 minutes for the decay version.

Participants reported being fairly relaxed about deleting other participants notes and making modifications to their contributions. This is in contrast to previous studies and ongoing public use where reluctance to edit others' contributions is evident. We suggest that this is due to the nature of the exercise set ('you must create a piece of music together for performance later') and the social situation (they all knew each other quite well and had possibly worked together before).

4.1 Patterns of Use with Persistent Version

As with ongoing analysis of the use of Daisyphone, in both versions the participants tended to spend the first parts of their sessions exploring Daisyphone on their own. Typically in the shared environment this meant working in a particular quadrant of the loop of music. Once participants were able to understand Daisyphone's interface they then moved on to working in other areas to develop longer tunes or contribute to other participants' work.

Interestingly, an informal role assignment developed when using the persistent version with participants tending to stick to one instrument. Moreover, a 'leader' tended to emerge during the sessions. This person typically constructed the main melody which was then supplemented by others in the group. Daisyphone has no explicit mechanisms or guidance for how to divide the collaborative effort, so we believe that we are starting to see here some emerging behavior which could give us insight into how to develop more engaging collaborations in the future. We suggest that role assignment emerges naturally and does not need to be explicitly built in to the interface *i.e.* in this case there was no need for ownership control of instruments as participants negotiated it themselves.

In pervious studies we noted that participants tend to write their name on Daisyphone. Given the ongoing nature of Daisyphone public trials, and the informal nature of other trials, we suggested that this name writing was a form of stating ownership - saying ‘This is mine’. From post study discussion it became clear that participants were using their names as presence and authorship indicators – saying ‘This is me’. Furthermore, as the study progressed, participants started to use shorter and shorter tags – starting from more explicit versions such as ‘Hi, its me, Nick’, to abbreviated versions such as ‘Nick’. Daisyphone was designed to provide mutual awareness of actions through shared and consistent representations of: the current state of the shared loop, different hues for each participant, and a flicker on the session selector when activity occurs in that session. We suggest that the emergent and conventionalized behavior of writing one’s name on entry to a session indicates that the messy nature of the interface additionally supports the informal evolution of expressions of identity. To this end we do not believe that the introduction of explicit identity into the interface is necessary or worthwhile. Interface features such as pictures, textual names, *etc*. add an unnecessary layer of interaction (setup, login, and so on) which we seek to avoid in the development of informal, *ad-hoc*, serendipitous musical interfaces.

4.2 Patterns of Use with the Decay Version

The use of Daisyphone with decaying contributions was not as engaging we had as anticipated. Participants complained that they could not keep up with the required contributions and that sessions tended to become unstructured and uncoordinated. Experience with Daisyphone as a musical instrument was a key factor in engagement with the decay version – the more experience participants had, the easier they found the decay version to handle.

When looking back over the logs of the interactions it is clear that in the version with decay participants tended to make musical ‘gestures’ rather than placing individual notes as they had on the persistent version. This is illustrated by the amount of annotation in figure 3 which reflects the creation of music through gesture rather than placing of notes as in figure 2. These gestures tended to be quickly drawn lines which could easily be replicated to keep the tune going. Perhaps providing an even more fluid form of interaction where gestures are interpreted around the Daisyphone would provide easier ways to create musical motifs in real time. It was also clear that the decay version required more focus on the music, and much less discussion of pieces, with participants having to keep musical motifs in their head in order to keep a tune going. In some ways this makes the decay version more akin to conventional group musical improvisation where typically the music and gestures provide for communication between participants as opposed to speech (or text in Daisyphone).

Anecdotally, there were more reports of experimentation with compositions with the decay version as the space did not require cleaning up. However, the persistence of annotation which provides some history of contributions did not prove as useful as anticipated as the proliferation of contributions meant that there were a lot of indications of old notes as illustrated by the mess of graphical lines in figure 3. Perhaps the sequence of contributions also needs to be indicated in some way.

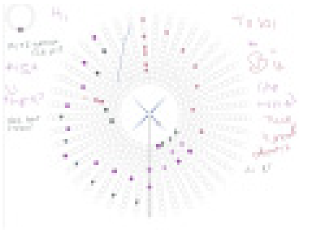


Figure 2: Persistence

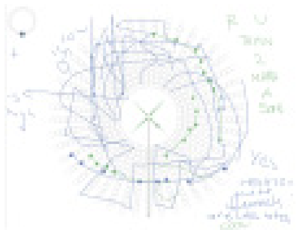


Figure 3: Decay

Also, interestingly there was anecdotally more convergence of tunes between participants with the decay version *i.e.* they started to *attune* to each other and make similar tunes within a group more quickly than they did with the persistent version. This indicates that decay may encourage quicker convergence of musical patterns after a period of experimentation. However, participants felt that they experienced flow as a group far less when the notes decayed as they felt anxious about making enough contributions, and felt that they had lost some control of the situation. So, it seems that whilst they mirrored each other more quickly, they did not transform each others’ contributions as they disappeared too quickly.

In terms of organization, participants found that with the decay of notes the division of labor was more egalitarian. That is, there was no longer the typical emergent leader of the piece, instead participants contributed what they could, with the tendency to converge quickly on a musical theme (if one could quickly be established).

Unexpectedly, participants tended to make non-overlapping contributions as with the persistent version. We had expected that when notes decay participants would start to make more contributions at the same time (as with conventional music playing). This may be a feature of the way music is made with Daisyphone rather than an indication of mutual engagement.

Finally, from analyzing the logs it is clear that participants contributed notes more frequently with the decay version (*e.g.* one group made approximately twice as many contributions per minute with decay *versus* persistence). This is clearly because of the amount of contributions that are needed to keep a tune going when the notes disappear. Figures 4 and 5 illustrate overviews of Daisyphone generated by the log tool. In these diagrams time is represented horizontally from left to right, and points in the timeline indicate a contribution of some sort with each column representing one second of interaction. Thus the amount of activity is indicated by how tall the columns are. As with Daisyphone itself, colors represent users – in figure 4 there are two users represented by green and blue, whereas in figure 5 the users are represented by purple and pink. Note that there are multiple saturations of the same color as saturation represents the volume of the contribution. Yellow points indicate the removal of notes in Daisyphone. In figure 5 we see the timeline for the example session shown in figure 3 lasting 13 minutes where contributions decay over time. There are several peaks throughout the interaction, and the total number of contributions 4230. In contrast, figure 4 illustrates the interaction in the persistent version over 11 minutes with fewer peaks (in this case these peaks are actually points at which writing takes place in the interaction whereas in the decay version they are musical contributions) and approximately half the number of contributions.

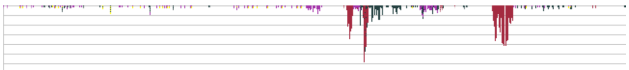


Figure 4: Example Persistence timeline

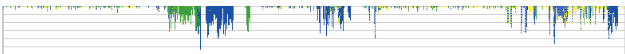


Figure 5: Example Decay timeline

5. Discussion

The key implication with respect to the decay of contributions is that contributions should only start disappearing once people have learnt how to usefully make them. We had expected the converse to be true – that when contributions decay it would be easier to learn the effects of the interaction through experimentation. So, we suggest that in order for creative musical experiences to become more engaging people's contributions should become more transient as they become more experienced, whilst support for the logistics of collaboration remain constant *e.g.* mutual awareness of actions should not change. We can usefully relate this to Csikszentmihalyi's analysis of flow and its relation to skills and challenges [5] as illustrated in figure 6. From this point of view, in order to experience flow one should have an appropriate match between the skills that people have and the challenges they are encountering – high levels of skill with low challenge leads to boredom, whereas high challenge with low skill leads to anxiety. As people become more skillful in relation to the activity they need to encounter greater challenges to remain in a flow experience.

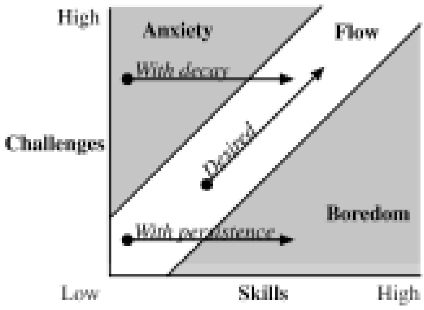


Figure 6: Flow, skills, and challenges, adapted from [5]

In the case of Daisyphone we believe that with persistence people became bored of the interaction as the challenge was no longer sufficient for their skills, whereas with decay participants were initially anxious, but some did increase their skills enough to experience flow as illustrated in figure 6. We suggest that as people become more skilled with the interface the rate of decay should gradually increase so that the challenge is sufficient for a flow experience, as illustrated by the desired interaction in figure 6. This could either be done automatically as time passed, or under user control as they felt boredom approaching. Doing so would provide people with an experience of music in which their initial low skills are supported by persistence of contribution, so not being too anxiety provoking, whilst boredom is abated by increasing the challenge (decay). Moreover, we suggest that by keeping the collaboration support constant the participants will become more engaged with each other as well as the product at hand. We would expect to see more convergence of music, and hopefully more reliance on others' contributions in the joint production. Furthermore, we believe that the decay of contributions by skilled users could be usefully employed to engender mutual engagement in other group creative tools such as brainstorming, problem solving, and so on.

6. Conclusion

This paper presents observations on the use of a novel group music making tool in two versions: one where musical contributions persist, and one where they decay. We suggest that allowing variable amounts of decay in an interface will allow the challenge of an interface to change to reflect the skills of participants and so hopefully more flow experiences will occur. Moreover, it will support increased engagement between people as indicated by more convergence and borrowing of other people's ideas. These are useful features for group music interfaces as well as other creative applications.

Additionally, we feel that the 'messy' nature of Daisyphone provides a useful interaction metaphor which informally supports many aspects of the logistics of collaboration including identity, awareness, history, localization, and the development of communicative conventions. We argue that the introduction of explicit support for these features of group interaction is unnecessary and instead suggest that more messy support will encourage people to intuitively develop their own conventions.

The key issue we are going to pursue next is how to flexibly manipulate the decay of contributions both graphically and musically without disturbing the participant, whilst still keeping the challenge, engagement, and flow to the forefront of their experience.

7. ACKNOWLEDGMENTS

This work was supported by Queen Mary, University of London and EPSRC grant GR/S81414/01.

8. REFERENCES

1.Blaine, T., and Fels, S. Contexts of Collaborative Musical Experiences. In *Proceedings of NIME-03* (Montreal, Canada, 2003), 129-134.

2.Bryan-Kinns, N. Daisyphone: Support for Remote Music Collaboration. In Proc. *NIME 2004*,

3.Bryan-Kinns, N. Daisyphone: The Design and Impact of a Novel Environment for Remote Group Music Improvisation. In Proc. of *DIS 2004*, ACM Press (2004), 135-144.

4.Burk, P. Jammin' on the Web - a new Client/Server Architecture for Multi-User Musical Performance. *Presented at ICMC 2000.*

5.Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience.* Harper Collins. 1991.

6.Leeuwen, T. *Speech, Music, Sound.* MacMillan. 1999.

7.Sawyer, R. K. *Group Creativity: Music, Theater, Collaboration*. Lawrence Erlbaum Ass. 2003

JamSpace: Designing A Collaborative Networked Music Space for Novices

Michael Gurevich
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
mgurevich@i2r.a-star.edu.sg

ABSTRACT

An interactive music environment to support real-time jamming by novices and amateur musicians over a network is described. JamSpace takes advantage of the low latency and connectivity of a local area network (LAN) to allow real-time rhythmic collaboration from isolated locations. Several demonstrated needs that motivate the design are discussed in detail. These include technologically-mediated ways of restoring casual social interactions to the domain of music creation and the preservation of anonymity and privacy for amateur musicians in a group setting. JamSpace's design addresses these needs with a novel hardware and software interface incorporating listening, private rehearsal, mixing, looping tracks and real-time jamming. User-configurable levels of interactivity are analyzed in terms of social spaces.

Keywords

Collaborative interface, remote jamming, network music, interaction design, novice, media space

1. INTRODUCTION

Most would agree that music is an inherently social activity [30], but since the introduction of recording and broadcasting technology, people's musical experiences have become increasingly private. Before these technologies, the only way to hear music was to play it yourself or hear others play it "live", which normally entailed attending a performance with other people. Whether concert music or explicitly social music for dancing, it was an experience to be shared. Broadcasting technology allowed people to listen to distant music in their homes. Soon, recorded music could be produced in an isolated studio, to be heard later in a private setting. Multi-track recording allowed musicians to collaborate on a record without ever meeting or playing together. Eventually, portable music players allowed people to have private music experiences anywhere, even in public places.

Private music listening is not necessarily undesirable, but there is evidence that the social nature of music is reasserting itself, often from a grassroots level, and often using technology. The Walkman existed concurrently with portable stereos (boom-boxes) in the 1980s, and was much less expensive at the time. Yet everyone is familiar with the image from that time of a person playing recorded music in public from a large boom-box perched on his shoulder. Regardless of the societal merits of this practice, it represented a clear desire to share the music listening experience, even in the face of cheaper, less cumbersome personal listening technology. In more recent phenomena such as iPod jacking [33], podcasting and sharing playlists, people have leveraged essentially personal technologies to create a social aspect to music listening. These examples imply that while our concept of music may be changing, it possesses some fundamental properties that compel us to share musical experiences.

Until quite recently, however, there have been relatively few examples of technologies designed with the deliberate aim of fostering social musical experiences. Karaoke is probably the most notable historical example. Whether in its sing-along or more intimate karaoke-box form, it brings together groups of people to "create" and listen to music, drawing on a shared knowledge of the popular music repertoire. Even karaoke itself was not designed top-down – it appears to have grown somewhat organically as a technological enhancement of a Japanese tradition of amateur music performance at social gatherings [34].

Just as technologies have been leveraged to create new social modalities for listening to music, they also have the opportunity to reintroduce casual social contexts for *making* music. Emerging technologies are beginning to address needs for technologically-mediated interactive social experiences, musical [6, 27] or otherwise [13], now by design.

The initial motivation for JamSpace was to create a distributed music application for a large local area network (LAN), to be used by amateur or novice musicians for recreation. There are a number of precedent application for music over networks, but most are either created by performers for their own use, or are designed for experienced musicians [1].

The technology of a large LAN, as in a hotel, office building or university campus provides two important features that are leveraged in the design of JamSpace in order to suit novice musicians. These are 1) low latency, and 2) connectivity in isolated locations. The design of JamSpace makes use of these features to offer constrained, real-time rhythmic performance with a user interface that maintains privacy and anonymity. The rationale is that privacy and anonymity allow users to engage the interface at their own pace, without inhibition or intimidation. Furthermore, individual users are given control over the level of interactivity. A metaphor of a flexible, configurable space is developed below to illustrate this idea.

Table 1: Networked Music Systems

System	Locations	Time	Sound
FMOL [20]	WWW	RT	SW synth
Jammin' on the Web [9]	WWW (remote)	Non-RT	SW synth
NetJam [21]	E-mail	Non-RT	MIDI Synth
Network-centric music perf [17]	LAN (local)	RT	Audio
NINJAM [25]	Internet (remote)	Synced Non-RT	Audio
Public Sound Objects [2]	WWW (remote)	Near-RT	SW Synth
SoundWIRE (Jamming) [28]	Internet2 (remote)	RT	Audio

2. CONTEXTS

As an interdisciplinary endeavour, the design of a novel music interface should be situated in terms of relevant contexts in a number of different fields. The design of a collaborative networked music application for novices in the contexts of network music, media spaces and online virtual environments is discussed below.

2.1 Network Music

Networks have been used for making music for some time. Network music systems can be categorized in terms of the locations of the of the performers (local vs. remote), the temporal quality of the interaction (real-time vs. non-real-time), and the nature of the sonic material (audio vs. synthesis). Selected network music systems are summarized in terms of these criteria in Table 1.

2.1.1 Location

Due to bandwidth constraints, most early network music systems used LANs, where data or audio were shared in real-time [4, 16]. LAN applications tend toward customized avant-garde performance systems. While LANs can span buildings or small areas, LAN music systems generally preserve the face-to-face nature of traditional music performance, focusing on new ways for performers to collaboratively synthesize and process sound in real-time. The inherent latency and bandwidth constraints of the Internet limit the degree to which these can simulate same-room interactions, but distributed applications can potentially include large numbers of users and provide radically new modes of interaction [20, 31].

2.1.2 Time

Data transit times of different networks are continuously variable, making it difficult to define a precise boundary between real-time and non-real-time. For the purposes of this discussion, real-time is defined as a best-effort attempt to have a local control cause an immediate response on a remote computer. Real-time systems therefore strive to appear synchronous from the users' perspectives, although this is not always achieved. Average transit times on a LAN are typically around 1 ms [12], and are at least one or two orders of magnitude longer on the Internet or other dedicated long-distance networks, depending on the distances involved. Synchronous real-time performance over Internet is a challenge do that particular network's inherent limitations.

The temporal characteristics of a network have profound influence on the interactions it can support. Tanaka [31] likens this idea to the intrinsic connection between musical genres and the environments in which their performance practice developed. One would not play bebop in a reverberant cathedral, for example. In fact, bebop would probably not have developed as it did if Charlie Parker had not been playing in jazz clubs. Network music systems may account for the temporal characteristics of the network in their sound design – using a slowly-varying synthesis algorithm for example, so that the effects of latency are less pronounced [20]. Some even exploit network delay as an integral part of their operation [11, 24, 29].

2.1.3 Sonic Material

Until recently, real-time processing and delivery of high-fidelity audio were limited by computing power and network bandwidth. Most early efforts therefore used synthesized sound that could be rendered locally on each machine or on a separate synthesizer [4, 21]. The network then only had to transmit much more compact control information, usually in the form of MIDI. Among the humblest systems was NetJam, which allowed users to edit MIDI files by email [21]. A now-defunct system known as ResRocket claimed to allow users to jam in real-time via the Internet using MIDI, though they had to cope with long and unstable delays inherent to the Internet.

Systems using audio either make use of the high bandwidth and low latency of a LAN [17] or dedicated research network [28] in order to facilitate real-time interaction, or else use non-real-time (or "fake-time") technology [25]. While it does not allow jamming with live instruments, synthesized sound has the advantage of engaging users who do not have musical instruments. Furthermore, it allows a deeper level of interaction design, where the sonic content and available controls can be appropriately designed for the spatial and temporal characteristics, as well as the users of the system. This can foster novel interactions, as in [1, 20, 29].

2.2 Media Spaces and Online Games

Media spaces typically consist of isolated locations linked by audio and video connections in order to create the metaphor of a seamless physical space for the purpose of collaborative work. Gaver [14] arrives at a similar conclusion to Tanaka with respect to media spaces, recognizing that virtual spaces do not support the same kinds of interactions as the real world, in spite of the metaphor. This is strongly related to the theory of affordances, which in psychology describes the relationship between properties of a system and the actions that can be taken on it [15]. Extended to the fields of HCI and design, affordances become the actions or rather the "perceived actions" that can take place by a user on an interface or object [26].

In spite of the recognition that technologies can create new modes of interaction, media spaces and many online virtual environments don't try to create new paradigms. For the most part, these try to imitate face-to-face interactions in a virtual world. This has led to criticism in the literature that network-based interactions are "unnatural" or inherently inferior to face-to-face interactions [23]. Many are indeed unsatisfying, but this is most often attributable to the fact that these systems have not exploited the unique possibilities and opportunities of the their underlying technologies. There still exists a preoccupation with simulating real life; creating experiences that are "just like being there", when in fact we could be creating experiences that are entirely not otherwise possible when we are in the same room.

There are many unique features that networks can offer. In many cultures there appears to be a tendency toward inhibition and intimidation with respect to novice or amateur musicians. This can result in an unwillingness for novices or amateurs to engage in public or shared music-making experiences. There is an opportunity for technology network technology to break from the real-world and offer an empowering experience to those who would not otherwise participate in making music.

3. INTERACTION DESIGN

JamSpace's hardware and software components are described and illustrated below. The interaction design is analyzed according to Blaine and Fels's contexts for collaborative interfaces [5], and in terms of spatial metaphors.

3.1 Hardware

Each JamSpace terminal has a JamPads hardware interface consisting of a flat surface with 12 raised pressure-sensitive pads. The pads can be pressed or struck with the user's hand, triggering a note with loudness proportional to force. The pads are mapped to percussion instruments or to the notes of the musical scale, depending on the instrument selected by the user. An LED below each translucent pad is illuminated when the pad is struck. Any track in the JamSpace or another players real-time jam can be assigned to the LED display on the pads, helping the user to visually and aurally learn other parts. Novices can learn to play along, the first step in creating, in very little time.

3.2 Software

The JamSpace software consists of a client GUI application and a separate server application. The client GUI consists of 5 components: a scratch track for the local user, a set of tracks from the JamSpace, an interface for making connections to the server, tempo and metronome settings, and a matrix for managing real-time jams with other clients.

3.2.1 Tracks

Users have one scratch track, into which they can privately record and play back one phrase. A drop-down box allows the user to select a synth instrument (currently using general-MIDI instruments, parametric software synthesis may be added in a future version). For tracks in the JamSpace, this box is replaced by a label. A user may choose to submit her scratch track to the JamSpace. Tracks in the JamSpace have a duration of one phrase, and loop until they expire after a period of time, but may be renewed through a voting mechanism. A server queue manages the finite number of active tracks in the JamSpace. Whenever there are less than 4 tracks in the jam, computer-generated tracks are added. Track data is displayed on a timeline interface.

3.2.2 Tempo

A global tempo is maintained by periodic sync messages from the server. 8 beats make up a phrase, and there is a cycle of 4 phrases. Each user can activate any of 3 click tracks which tick at the phrase, beat, and half-beat time scales. Any user can request a tempo change by typing a new tempo in the box. Changes in tempo are also managed by voting.

3.2.3 Real-Time Jamming

Users can jam to their custom looping track mixes or with other live users in the JamSpace. Users may also broadcast their jams, in which case all other users can choose to listen to them in real-time. The low-latency of the LAN and compactness of synth control data transmitted between jammers assures that jams appear to be synchronous to the users.

Users choose an icon from a pre-defined list to represent themselves in the jam. A unique icon appears beside all tracks that a user has submitted to the JamSpace, as well as beside her place in the connection matrix. This affords direct musical communication and development of taste, style and identity, while protecting privacy and anonymity.

3.3 Contexts of Collaborative Interfaces

Existing networked music systems typically target experienced musicians. Interfaces for novices do exist [2, 8], but these are the exception, as the designers of such systems tend to be musicians themselves. Blaine and Fels [5] point out a similar trend in the NIME field as a whole, particularly with respect to music controllers. They acknowledge a tradeoff between the inherent expressivity of an interface and its complexity, which demands a non-trivial balancing on the part of the designer. In a sense, this is the challenge of any interaction designer, but it is significantly complicated by the fact that in music, the task, and therefore the evaluation of a user's ability to perform that task, are often not clearly defined. Framing this as a tradeoff implicitly renders Wessel and Wright's ideal of "low entry fee with no ceiling on virtuosity" fundamentally paradoxical [32].

Blaine and Fels analyze a number of collaborative music systems in terms of 'constraint' over a variety of design elements of the interfaces, where highly constrained interfaces are generally instantly accessible but provide little room for innovation. They argue that the duration and location of engagement of a collaborative music interface by a user can partly dictate the level of constraint necessary to provide a satisfying experience. Group interfaces for novices situated in public places often engage users for short periods of time and must therefore provide a minimal learning time – the ability to "walk up and play" that a novice almost never gets from a traditional instrument.

For the purposes of trying to maintain a general framework for evaluating collaborative interfaces, I discuss JamSpace with respect to several of the design elements described by Blaine and Fels. Some of these elements are obvious or have been discussed above, therefore what follows does not exhaustively cover Blaine and Fels's list.

Focus distinguishes a performance interface for the benefit of an audience versus a recreational experience for the benefit of the players. While JamSpace is intended to include at least one station in a public location that could support a casual audience, the primary focus is for the enjoyment of the players.

The **scale** of the JamSpace presently allows up to 14 simultaneous users. However, the configurable modes of **player interaction** ensure that each user's experience is unique. Each user can choose her own mix of available tracks, and monitor any number of the broadcasting jammers in real-time. While each user possesses the same interface, the experience and contribution of each is unique.

JamSpace does not explicitly employ **directed interactions**, but the interface does allow users to indirectly learn from each other. Flexible modes of interactivity al-

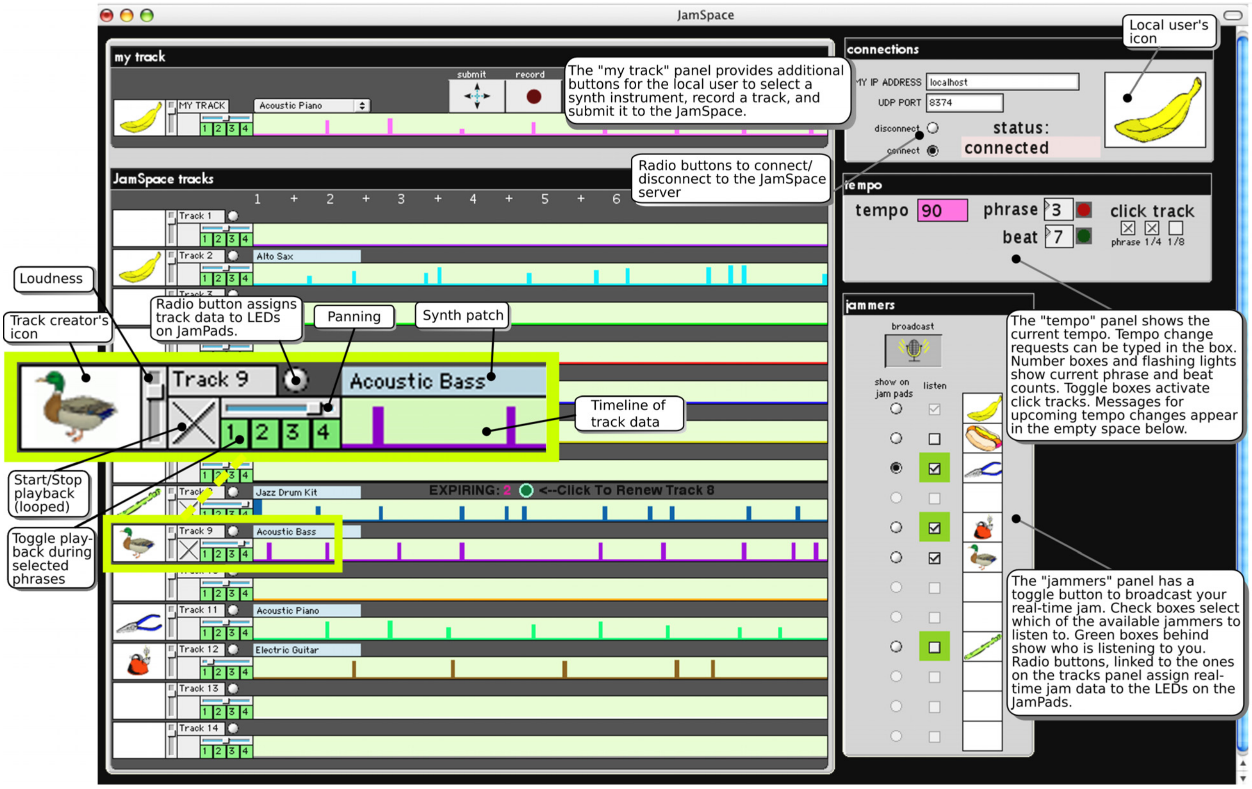


Figure 1: JamSpace client GUI

low users to see what another person is playing in real-time, or the contents of a looping track. By assigning any of these sources to LEDs embedded in the hardware control pads, a user can see what another is playing, directly on her own interface. By disabling broadcast mode, the user can privately play along, develop her own variations, and record them to a track or broadcast to the JamSpace at her own pace.

Blaine and Fels do not consider the particular case of networked collaborative interfaces in detail, and therefore there are unique constraints that apply to JamSpace with respect to several of their design aspects. Of particular importance are the **pathway to expert performance** and **learning curve**. Blaine and Fels argue that a "low entry fee" is paramount, and thus the ideal of "no ceiling on virtuosity" must be tempered. This is mostly due to the pattern of engagement of publicly-situated collaborative interfaces, where users do not have the time to even feel the need to achieve virtuosity, let alone the time to develop it. In order to provide enough rapid satisfaction to ensure further engagement, the public interface must also have a fast learning curve. A privately-situated interface such as the JamSpace is quite different. Users do not have infinite time, but may explore the interface at their own pace, without any public pressure. They are more likely to have the opportunity to return to the interface after some amount of time. This excuses a moderately slower learning time, as self-paced exploration and discovery are part of the design. Looser control over the **musical range** of the material is therefore also warranted, so that users may remain satisfied and develop proficiency over time. JamSpace allows a set of chromatic notes in a constrained octave and free rhythms within the circumscribed metric structure. The interface is simple enough that an amateur musician with a basic knowledge of notes and rhythm can produce meaningful material. Novices can easily grasp and expand on this material or that of computer-generated tracks.

3.4 Spaces

The levels of interactivity within the JamSpace can be analyzed in terms of a spatial metaphor. The different modes of interactivity can be seen to reflect different spaces. In some ways, it also demonstrates characteristics of different spaces at the same time. There is a strong tradition of discussing systems for computer-supported collaborative work (CSCW) and virtual environment in terms of real-world spaces. Jeffrey [19] demonstrates the applicability of the psychological concepts of personal space, group space and privacy in virtual environments. The following discussion pertains to JamSpace stations located in isolated locations.

3.4.1 Private Space

Private spaces imply physical barriers that can exclude all but one person. Privacy has different meanings in different contexts. With respect to technology, it is synonymous with security and confidentiality. In psychology, it normally refers to isolation or solitude [7]. In both senses, JamSpace can represent a private space when the user is not connected to the JamSpace server. In this mode, the user is invisible to all others, and may play on his or her own JamPads, and record, play back and play along with one track. The user's identity and presence are invisible to the world, and he or she is assured of an environment that may not be intruded.

3.4.2 Personal Space

Personal space is an individually-variable, context-dependent concept in social psychology that refers to a preferred boundary zone around a person [22]. Distinct from but related to privacy, the boundaries of personal space are transparent - when you are in public, anyone can see you within your own personal space, and others are clearly visible to you. The aim of JamSpace is not to explicitly capture this concept in a collaborative music environment. Rather, it employs a level of interaction wherein the user is aware of the presence of others and vice versa, but they do not interact and therefore do not share space. In connecting to the JamSpace, other connected users become aware of the user's presence, and he or she becomes aware of them. The user may perceive the rest of the world - he or she receives tracks and can listen to other jammers, but cannot actively participate with them until progressing to a further level of interactivity.

3.4.3 Shared Space

Shared spaces are occupied by groups of people. Like personal space, they may exist with transparent borders within a public space. As CSCW began to flourish in the late 1980's, the notion of creating virtual shared spaces or "media spaces" [14] for collaborative tasks became the standard paradigm for remote collaborative work. Benford [3] analyzes spatial approaches to collaborative work according to the criteria of transportation, spatiality and artificiality. Implicit in this analysis is that space is not just a metaphor in these systems, there is a deliberate attempt to produce or reproduce a 3-dimensional space complete with representations of its human occupants. Buxton [10] distinguishes between shared *person* and *task* spaces, where the former refers to an overall sense of copresence and mutual awareness, while the latter is constrained to the domain of a task. Task spaces do not necessarily include the assumption of explicit spatial representations. Harrison and Dourish [18] challenge the pervasiveness of spatial metaphors, arguing that many CSCW systems more closely embody a concept of *place* than they do *space*, and therefore offer a different set of affordances. Breaking the spatial metaphor carries with it the opportunity for conceiving interactions that are not possible in real spaces, but may be otherwise desirable.

In that there are no graphical or explicit spatial representations of the users in the JamSpace, therefore the spatial metaphor is a weak one. The notion of a collaborative task space is more appropriate. By broadcasting a real-time jam and/or submitting tracks to the JamSpace, users can actively share the JamSpace with others. Users can see who else is listening to them, and may choose to listen in turn and engage in jamming. Submitted tracks form an integral part of the jam, to which users collaboratively contribute.

There is a slight distinction between this metaphor and the traditional definition of shared space, particularly with respect to membership and invitation. In social psychology, group membership carries strong consequences, and complex social mechanisms govern membership and belonging. In the JamSpace, there are no explicit ways of communicating invitation or perceiving membership. Anyone with a JamSpace terminal may join at any time, and users don't know the identities of group members, nor can they monitor all of the members' actions. This breakdown of the strict metaphor of social spaces is not seen as a limitation in the design, however. Rather, JamSpace drawing on aspects of different types of spatial interactions in a beneficial way. Real-world privacy and personal space

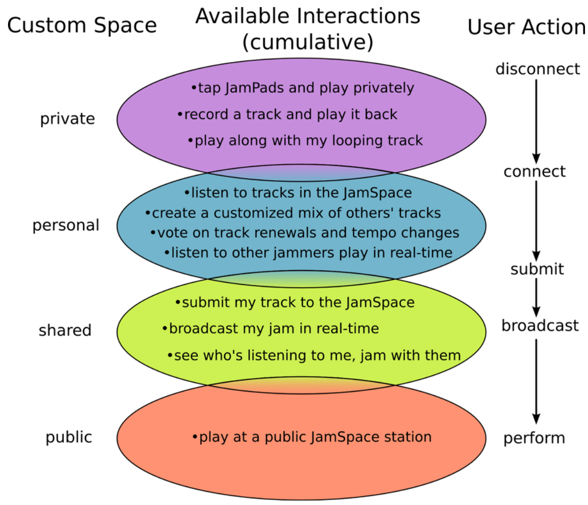


Figure 2: User-configurable space metaphor

are maintained while group interactions are possible.

3.4.4 Public Space

People are mostly free to see and do what they wish in public spaces, within a set of social and cultural norms. They are venues for self-expression and places where people may gather. The design of public spaces must balance the needs for freedom and expression, with those of common decency and protection from offence. In simultaneously embodying the characters of multiple spaces JamSpace assures these ideals partly by the same mechanisms that it ensures privacy. Constraint over the sonic material, and protection of identity ensures that no direct communication between users is possible. Unlike real public spaces, the only possible offence or intimidation that could be perpetrated would be not listening to another person's jam. However, the ideal of expression is maintained. In broadcasting and submitting tracks, a user airs his or her expression in front of all other JamSpace users, regardless of whether they are listening.

JamSpace stations in real public places elevate expression to a different level, providing a venue for the local performer publicly demonstrate his or her musical skills or self-expression in a real-world space. Of course, real-world social conventions and limitations apply here, and the user is subject to the ensuing rewards as well as consequences.

4. CONCLUSIONS

With respect to JamSpace, the spatial metaphor is really just a metaphor. In many CSCW systems, the design seeks to emulate or simulate real spaces and the kinds of interactions that they support. For JamSpace, a spatial metaphor is a useful conceptual way of characterizing the interactivity, but the system does make explicit representations of spaces. Instead, it leverages the characteristics of its technology, users and scenario to create new modes of musical interaction.

The overall design philosophy of JamSpace was to begin with a specific technological platform (local network) and application area (recreational music), and then leverage their affordances to find novel interactions that address the requirements of the scenario. JamSpace is a work in progress. The system is currently being deployed for evaluation with two actual hardware interfaces, along with a number of others using a software emulation of the hardware. Initial impressions show the system to be engaging for both novices and experienced musicians. A systematic evaluation is forthcoming.

5. REFERENCES

[1] A. Barbosa. Displaced soundscapes: A survey of network systems for music and sonic art creation. *Leonardo Music Journal*, 13:53–59, 2003.

[2] A. Barbosa and M. Kaltenbrunner. Public sound objects: A shared musical space on the web. In *Proc. Intl. Conf. on Web Delivering of Music*, 2002.

[3] S. Benford, C. Greenhalgh, G. Reynard, C. Brown, and B. Koleva. Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Trans. on Computer-Human Interaction*, 5:185–223, 1998.

[4] J. Bischoff, R. Gold, and J. Horton. Music for an interactive network of microcomputers. *Computer Music Journal*, 2:24–29, 1978.

[5] T. Blaine and S. Fels. Contexts of collaborative musical experiences. In *Proc. 3rd Intl. Conf. on New Interfaces for Musical Expression (NIME03)*, 2003.

[6] T. Blaine and T. Perkis. The Jam-O-Drum interactive music system: A study in interaction design. In *DIS '00: Proc. 2000 Conf. on Designing interactive systems*, pages 165–173. ACM Press, 2000.

[7] M. Boyle and S. Greenberg. The language of privacy: Learning from video media space analysis and design. *ACM Trans. on Computer-Human Interaction*, 12:328–370, 2005.

[8] N. Bryan-Kinns. Daisyphone: The design and impact of a novel environment for remote group music improvisation. In *DIS '04: Proceedings of the 2004 Conference on Designing Interactive Systems*, pages 67–76. ACM Press, 2004.

[9] P. Burk. Jammin' on the web – a new client/server architecture for multi-user musical performance. In *Proc. 2000 Intl. Computer Music Conf.*, 2000.

[10] W. Buxton. Telepresence: Integrating shared task and person spaces. In *Proc. Conf. on Graphics Interface '92*, pages 123–129, 1992.

[11] C. Chafe. Distributed Internet reverberation for audio collaboration. In *AES 24th Intl. Conf. on Multichannel Audio*, 2003.

[12] C. Chafe, S. Wilson, R. Leistikow, D. Chisholm, and G. Scavone. A simplified approach to high quality music and sound over IP. In *Proc. COST-G6 Conf. on Digital Audio Effects (DAFX-00)*, 2000.

[13] E. F. Churchill, L. Nelson, L. Denoue, J. Helfman, and P. Murphy. Sharing multimedia content with interactive public displays: a case study. In *DIS '04: Proc. 2004 Conf. on Designing interactive systems*, pages 7–16. ACM Press, 2004.

[14] W. W. Gaver. The affordances of media spaces for collaboration. In *CSCW '92: Proc. 1992 ACM Conf. on Computer-Supported Cooperative Work*, pages 17–24. ACM Press, 1992.

[15] J. J. Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, 1979.

[16] S. Gresham-Lancaster. The aesthetics and history of The Hub: The effects of changing technology on network computer music. *Leonardo Music Journal*, 8:39–44, 1998.

[17] X. Gu, M. Dick, Z. Kurtisi, U. Noyer, and L. Wolf. Network-centric music performance: Practice and experiments. *IEEE Communications*, 43:86–93, 2005.

[18] S. Harrison and P. Dourish. Re-place-ing space: the roles of place and space in collaborative systems. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 135–144. ACM Press, 1996.

[19] P. Jeffrey and G. Mark. Constructing social spaces in virtual environments: a study of navigation and interaction. In K. Höök, A. Munro, and D. Benyon, editors, *Workshop on Personalised and Socal Navigation in Information Space*, pages 24–38, 1998.

[20] S. Jordà. Faust music on line (FMOL): An approach to real-time collective composition on the Internet. *Leonardo Music Journal*, 9:5–12, 1999.

[21] C. Latta. Notes from the NetJam project. *Leonardo Music Journal*, 1:103–105, 1991.

[22] K. B. Little. Personal space. *Journal of Experimental Social Psychology*, 1:237–247, 1965.

[23] C. Magerkurth, T. Engelke, and M. Memisoglu. Augmenting the virtual domain with physical and social elements: towards a paradigm shift in computer entertainment technology. *ACM Trans. Computers in Entertainment*, 2(4):12–12, 2004.

[24] Max Neuhaus. `http://www.ubu.com/sound/neuhaus_radio.html`.

[25] NINJAM - Novel Intervallic Network Jamming. `http://www.ninjam.com/`.

[26] D. A. Norman. Affordances and design. `http://www.jnd.org/dn.mss/affordances_and.html`.

[27] J. Scheible and T. Ojala. MobiLenin combining a multi-track music video, personal mobile phones and a public display into multi-user interactive entertainment. In *Proc. 13th ACM Intl. Conf. on Multimedia*, pages 191–198. ACM Press, 2005.

[28] SoundWIRE. `http://ccrma.stanford.edu/groups/soundwire/`.

[29] J. Stelkens. peerSynth: A P2P multi-user software with new techniques for integrating latency in real time collaboration. In *Proc. 2003 Intl. Computer Music Conf.*, 2003.

[30] A. Tanaka. Mobile music making. In *Proc. 4th Intl. Conf. on New Interfaces for Musical Expression (NIME04)*, 2004.

[31] A. Tanaka, N. Tokui, and A. Momeni. Facilitating collective musical creativity. In *Proc. 13th ACM Intl. Conf. on Multimedia*, pages 199–208. ACM Press, 2005.

[32] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *Proc. 1st Workshop on New Interfaces for Musical Expression (NIME01)*, 2001.

[33] Wired News: Feel free to jack my iPod. `http://www.wired.com/news/mac/0,2125,61242,00.html`.

[34] C. R. Yano. The floating world of karaoke in Japan. *Popular Music and Society*, 2:1–17, 1996.

Creating a Network of *Integral Music Controllers*

R. Benjamin Knapp
BioControl Systems, LLC
Sebastopol, CA 95472
+001-415-602-9506
knapp@biocontrol.com

Perry R. Cook
Princeton University
Computer Science (also Music)
Princeton, NJ 08540
+001-609-258-4951
prc@cs.princeton.edu

ABSTRACT

In this paper, we describe the networking of multiple Integral Music Controllers (IMCs) to enable an entirely new method for creating music by tapping into the composite gestures and emotions of not just one, but many performers. The concept and operation of an IMC is reviewed as well as its use in a network of IMC controllers. We then introduce a new technique of Integral Music Control by assessing the composite gesture(s) and emotion(s) of a group of performers through the use of a wireless mesh network. The Telemuse, an IMC designed precisely for this kind of performance, is described and its use in a new musical performance project under development by the authors is discussed.

Keywords

Integral Music Control, Musical Control Networks, Physiological Interface, Emotion and Gesture Recognition

1. INTRODUCTION

The Integral Music Controller (IMC) [1] is defined as a controller that:

1. Creates a direct interface between emotion and sound production unencumbered by the physical interface.
2. Enables the musician to move between this direct emotional control of sound synthesis and the physical interaction with a traditional acoustic instrument and through all of the possible levels of interaction in between.

This paper describes the networking of multiple IMC's, to enable not just one, but many performers to use an IMC and to interact with each other in three ways:

1. The "normal" perceptual path – the performers see, hear, and sometimes even haptically feel the other performer. .
2. The controller interaction path – the performers physical gestures and emotional state, as assessed by the IMC, are used to another performer's electro-acoustic instrument.
3. The integral control path – an entirely new path whereby the emotions or gestures of one performer, as measured by the IMC, are combined with the emotions and gestures of other performers to create an assessment of group gestures and emotions and this is used to control music creation.

2. REVIEW OF INTEGRAL MUSIC CONTROL (from [1])

The term "integral" in "integral music controller" refers to the integration into one controller of the pyramid of interface possibilities as shown in Figure 1. Using an IMC, a performer can move up and down through the interface possibilities.

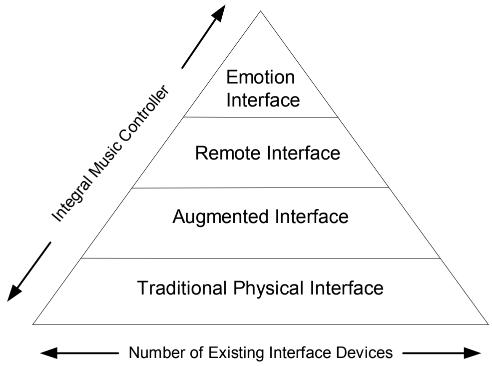


Figure 1: Pyramid of interfaces for controlling a digital musical instrument (categories loosely adapted from [2]). Note the decreasing number of existing interface devices as you move up the pyramid. The integral music controller (IMC) has elements of all interfaces.

As shown in Figure 1, the introduction of direct measurement of emotion to digital musical instrument control represents the completing of the pyramid of possible interfaces. Only with a direct interface to emotion is a truly integral controller possible.

The use of a direct emotional interface also introduces one new feedback path in a musical performance that was never before possible. Figure 2 shows three layers of feedback that can be achieved in musical performance. Layer 1 is the emotional layer. The emotional state of the performer initiates and adjusts the physical gesture being made. This emotional state might or might not be reflective of the intention of the performer. Also, the perception of the sound that is created from the physical gesture elicits an emotional response in the performer and, based on this; the performer may alter the physical gesture. Layer 2 is the physical interface layer. Feedback is achieved through visual cues and proprioception [3]. Layer 3 is the sound generation layer. The physical gestures cause a sound to be created which is heard and possibly used by the performer to adjust the physical gesture [4]. The introduction of a direct emotional interface means that a performer's emotions will directly control the sound generation without passing through the physical interface. The sounds created will effect the emotion of the performer [5] and thus a new feedback path is created.

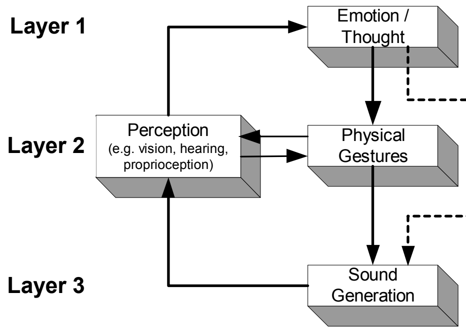


Figure 2: The three layers of performance feedback using an IMC. Layer 1 represents the internal emotion and thoughts of the performer. Layer 2 is the physical interface layer. Layer 3 represents the consequence of the gesture - the creation of music.

There is an extensive body of literature on defining, measuring and using emotion as a part of human computer interaction and "affective" computing (see [6][7][8][9] for a good overview). The emotional reaction to music is so strong that music is commonly used as the stimulus in emotion research [10]. The understanding of the emotional reaction to music, not the categorization or labeling, is critical in using emotion as a direct performance interface. It is clear [3] that this emotional reaction is highly individualistic and thus any synthesis model that uses emotion as an input must have the capability of being customized to an individual performer.

There are many techniques [9] for measurement of emotion including visual recognition of facial expression, auditory recognition of speech, and pattern recognition of physiological signals. For most musical performance environments visual recognition systems would not be appropriate. Thus, physiological signals are the most robust technique for determining emotional state for direct emotional control of a digital music instrument. Physiological signals have been used many times as a technique of human computer interaction in music [11][12][13] for example). Their responsiveness to both motion and emotion makes them an ideal class of signals that can be used as part of an IMC.

3. THE NETWORKED CONTROLLER

The inclusion of networked interaction in electro-acoustic instrument performance introduces a new path for performers to communicate. Networked music controllers can be thought of as a subset of multi-user instruments (see [14] for a summary of such instruments). There are numerous examples of networked controllers used in performance including the MIT Media Lab's Brain Opera [15] and Toy Symphony [16]. In the latter, the BeatBugs [17] allowed the players to enter musical material, then play it or modify it by manipulating sensors on the bug, and/or pass it to another player by pointing the bug at them. A subset of networked controllers is so-called "wearables" and include networked jewelry and clothing [28].

The Princeton Laptop Orchestra (PLOrk) [18] is a recent experiment in constructing an orchestra of sensor connected laptops and speakers. Various composing / performing / conducting paradigms have been investigated, including passing synchronization and other messages related to timbre, texture, etc. over 802.11G using Open Sound Control (OSC). The language ChucK [19] is one of the primary programming mechanisms used by PLOrk, as there is a rich provision for low-latency (10-20 ms.) asynchronous messaging built into the language.

Figure 3 is a block diagram of networked IMC's showing the networked interaction path separated into a physical gesture path and an emotion path. (Note the already existing perceptual path which symbolizes the performers' ability to see, hear, and even feel each others performance.) These new networked interaction paths create a way for performers to collaborate with each other at the controller level before the sounds are actually created. Each performer's physical gesture or emotional state is recognized and converted into a control parameter(s) that can be combined with the control parameter(s) of other performers to create a rich and complex means for group performance.

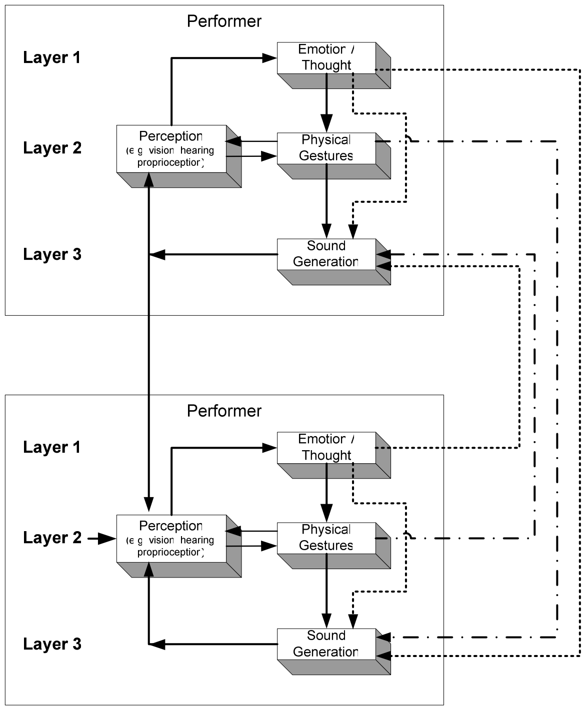


Figure 3: The networking of multiple IMC's. The solid line between each performer represents at the perceptual level. The dashed-dot line shows the physical interaction at the controller level, i.e., how the physical gesture of one performer can effect the sound generation of another performer's instrument. The dotted line shows the emotional interaction at the controller level, i.e., how the emotion of one performer can effect the sound generation of another performer's instrument.

4. THE INTEGRAL CONTROL PATH

Unlike standard networked instruments, the integral control path seeks to combine the physical gestures and emotional state of multiple performers *before* they are categorized and processed into control parameters. The purpose of this is to assess a composite emotion or gesture of multiple performers first, and then to use this as a control input. As shown in Figure 4 this requires a mesh computation of composite signals. Only a completely self-forming, self-aware mesh network topology would enable sets and subsets of different performers to interact with sets and subsets of instruments in real-time.

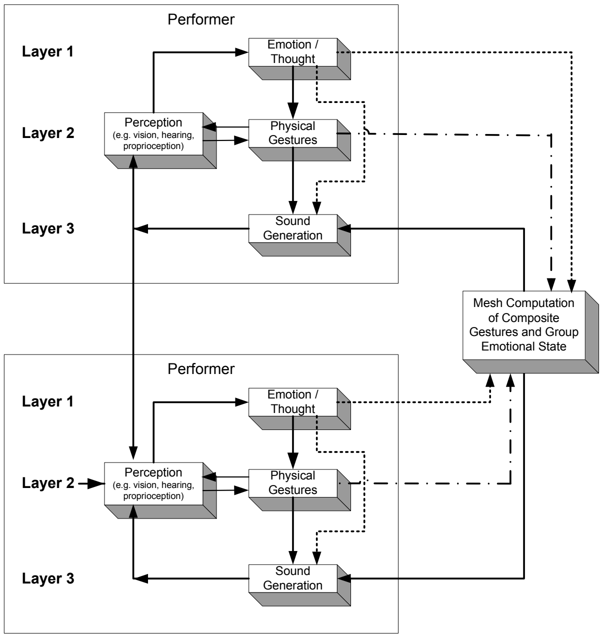


Figure 4: The networking of multiple IMC's using an integral control path as part of a mesh. In this mesh, any performer's physical gestures and emotional state can be composited with any other's.

Both forms of networking can be combined to create a network of integrally networked IMC's. Thus, for example, a performer's emotional state can be assessed by the IMC, combined to create with other performer(s) to create an overall combined emotional state, this state can be used to control the output of a controller within a network of controllers. A detailed example of this will be discussed in section 6 of this paper.

5. IMPLEMENTATION: THE TELEMUSE

There are many systems that wirelessly transmit physiological data including the BodyMedia's SenseWear [20], NASA's Lifeguard[21], and MIT's LiveNet[22]. There are several sensor systems that use wireless mesh networking, and more specifically, a network layer protocol known as ZigBee™. ZigBee™ is designed to use the IEEE 802.15.4 standard, a specification for a cost-effective, relatively low data rate (<250 kbps), 2.4 GHz or 868/928 MHz wireless technology designed for personal-area and device-to-device wireless networking [23]. There are several companies that have ZigBee™-based sensor units including those made by Crossbow [24], Dust [25], and MoteIV [26]. Harvard's CodeBlue [27] uses Crossbow's ZigBee™ compliant motes to create a mesh network of physiological sensors. None of these interfaces are designed specifically as human-computer interfaces, let alone musical instrument controllers, and therefore none of the designers incorporated the use of an integral control path.

The TeleMuse system shown in Figure 5 integrates physiological signal sensors, motion sensors, and a ZigBee™ wireless transceiver into one band designed for human-computer interaction and music control. The TeleMuse can be worn:

- on the limbs to measure muscle tension (EMG), Galvanic Skin Response and motion (dual axis accelerometers)
- on the head to measure brain activity (EEG), muscle tension (EMG), eye motion, and head motion (dual axis accelerometers)
- on the chest to measure heart activity (EKG) and respiration

Figure 5: The Telemuse Wireless Mesh Network IMC

The TeleMuse is the next generation of Integral Music Controller replacing the Wireless Physiological Monitor (WPM) [29] in a smaller more ergonomic design. Like the WPM, the TeleMuse uses dry electodes to sense physiological data. Unlike the WPM, each TeleMuse is its own node in a mesh network and can communicate with any other node in the network. Computation of physical gestures and emotional state, based on physiological signals and accelerometer data, can be distributed among any of the nodes and any computers on the network.

6. VACHORALE: A PIECE FOR PLORK, TELEMUSE AND SINGERS

One use of networked IMC's will be investigated in a project entitled the Virtual/Augmented Chorale (VAChorale). The Virtual/Augmented Chorale (VAChorale) project will investigate the compositional and performance opportunities of a "cyber extended vocal ensemble" and will use the Princeton Laptop Orchestra (PLOrk), the TeleMuse, and ChucK.

6.1 Augmenting The Singer

The VAChorale project will outfit a small choir of (eight) singers with several Telemuses and microphones, coupling each human singer to a laptop, multi-channel sound interface, and multi-channel hemispherical speaker. As an obvious first step, the system will use digital signal processing to modify and augment the acoustical sound of the singers. Further, we will use networked TeleMuses to control various algorithms for modifying and extending the choral sound. The most revolutionary component will be using the TeleMuse to control various sound (primarily voice/singing) synthesis algorithms, in order to extend, and even replace the acoustic components of the choir. The singers will thus be able to "sing" without phonating, controlling the virtual choir with facial gestures, head position, breathing, heart rate, and other non-acoustic signals. An assessment of each singer's emotional state, as well as the choir's composite emotional state will be used as well.

We plan to fully realize the IMC concept, with the physical gestural "instrument" being a singer, and we will create an ensemble of multiple IMC-outfitted Virtual/Augmented singers. The continuum from the "dry" choral sound, through the digitally augmented acoustic sounds of the singers, to the completely virtual sound of the biological sensor-controlled synthesized singing, will provide a rich compositional and performance space in which to create new music.

6.2 Building The "Instruments"

The first goal of the project is to integrate the existing hardware and software systems for biological signal acquisition and processing, acoustical signal processing, and voice synthesis, with the PLOrk (Princeton Laptop Orchestra) "workstations" to create a new augmented singer "instrument." These instruments, hereafter called VAChS (Virtual/Augmented Choral Singers, pronounced "vax") will be identical in technical capability, but can take on various forms based on configuration, programming, and control.

First, the PLOrkStations provide the basic computational and acoustical technical capabilities. Built with support from the Princeton University Council on Science and Technology, the Princeton Freshman Seminar Program, the Princeton departments of Music and Computer Science, the Princeton School of Engineering and Applied Science, and Apple Computer, each of the 15 existing workstations consists of a 12" Mac Powerbook, an Edirol multi-channel FireWire digital audio interface box, six channels of amplification, and a custom-built six-discrete-channel hemispherical speaker.

Second, the TeleMuse will couple each singer in the ensemble to a networked hardware workstation. Physiolgical signals will be captured and processed by each TelMuse node and shared with the rest of the mesh network. As mentioned previously, these signals can be used to determine not only singing gestures, but also the emotional state of the performers. Additionally, each box contains a two-axis accelerometer, so head/body tilt and orientation can be measured. Additional sensors can be used to measure absolute body and head position and orientation.

ChucK was specifically designed to allow rapid, on-the-fly audio and music programming and will be used to synchronize the multiple composited controller streams.

6.3 Virtualizing the Singer

The physiologically-derived emotion signals of the IMC can be mapped to signal processing such as adding echoes and reverberation, shifting pitch, controlling spatial position, etc., and compositional processes such as note generation and accompaniment algorithms. But the IMC can also be mapped to the parameters of physical synthesis models, creating a truly integral controller. In fact, indirect emotion mapping already exists in many acoustic instruments. The nervousness of a singer or violin player already shows in the pitch jitter and spectral shimmer of the acoustical instrument. The heartbeat of the singer modulates the voice pitch because of modulation of lung pressure.

Synthesis by physical modelling lends naturally to control from physical gestural parameters. Signals such as those that come from an IMC can easily be detected and mapped to similar, or totally different (brightness, spatial position, etc) parameters in a physical synthesis model. With higher-level control and player-modeling inside the model, emotional parameters might make even more sense than raw gestural ones. A large variety of parametric physical instrument synthesis models exist in ChucK, with many holding much promise for control from singer gestures and emotional parameters. The models that hold the most interest for this project, however, are those that mimic the human singing voice.

Older proven models for voice synthesis, such as formant filter synthesizers, and articulatory acoustic tube models, already exist in ChucK as native "unit generators." As such it will be easy to perform a number of different mapping experiments, and produce a variety of human-like (and quite inhuman) sounds based on control from the singer sensors.

New models of the human voice such as Yamaha's Vocoloid (constructed with UPF Barcelona) allow for control of vocal quality parameters such as growl, breathyness, and raspiness, and more semantic qualities such as "bluesyness" and "sultryness". These also seem completely natural for control by emotional parameters, and will be exploited in the Virtual/Augmented Chorale project.

6.4 The Performance

The goal of the Virtual/Augmented Chorale project is to compose and rehearse a number of choral pieces, aimed at the production of several concert performances. The repertoire will range from traditional early music augmented by the virtual acoustics of the VAChS, through some contemporary a capella vocal literature, but with the human ensemble augmented by virtual singers, and one or two brand new pieces composed specifically to exploit the maximum capabilities of the Virtual/Augmented Chorale.

7. REFERENCES

[1] Knapp, R.B. and Cook, P.R., The Integral Music Controller: Introducing a Direct Emotional Interface to Gestural Control of Sound Synthesis, In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, September 5-9, 2005.

[2] Wanderley, M.M. "Gestural Control of Music", IRCAM – Centre Pompidou, 2000.

[3] Askenfelt, A. and Jansson, E.V. "On Vibration Sensation and Finger Touch in Stringed Instrument Playing," Music Perception 9(3), 1992, pp. 311-350.

[4] P. Cook, "Hearing, Feeling, and Performing: Masking Studies with Trombone Players," International Conference on Music Perception and Cognition, Montreal, 1996.

[5] Panksepp, J. and Bernatzky, G. "Emotional Sounds and the Brain: the Neuro-affective Foundations of Musical Appreciation," *Behavioural Processes*, 60, 2002, pp. 133-155.

[6] Holland NN, "The Brain and the Book, Seminar 7: Emotion", http://web.clas.ufl.edu/users/nnh/sem04/memo-s04.htm, February 9, 2004.

[7] Hudlicka, E. "To Feel of not to Feel: The Role of Affect in Human-Computer Interaction," *International Journal of Human-Computer Studies*, 59, 1-32, 2003.

[8] Picard, R.W., Vyzas, E., and Healey, J. "Toward Machine Emotional Intelligence: Analysis of Affective Physiological State," *IEEE Transactions on Pattern Analysis and Machine Intellligence*, 23, 10, October 2001, pp. 1175-1191.

[9] Nasoz, F., Alvarez, K., Lisetti, C. L., & Finkelstein, N. "Emotion Recognition from Physiological Signals Using Wireless Sensors for Presence Technologies," *Cogn. Tech. Work*, 6, 2004, pp. 4-14.

[10] Steinberg, R. (Ed.). *Music and the Mind Machine*, Springer, Berlin, 1995.

[11] Tanaka, A and Knapp, R.B., “Multimodal Interaction in Music Using the Electromyogram and Relative Position Sensing,” *Proceedings of the New Interfaces for Musical Expression (NIME) Conference, Media Lab Europe, Dublin, Ireland,* 2002.

[12] Knapp, R. B. & Lusted, H.S. “A Bioelectric Controller for Computer Music Applications,” *Computer Music Journal*, 14 (1), 1990, pp. 42-47.

[13] T. Marin Nakra, “Inside the Conductor’s Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture,” PhD Dissertation, MIT Media Lab, 2000.

[14] Jorda, S. "Multi-user Instruments: Models, Examples and Promises, " *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada, pp. 23-26.

[15] Joseph A. Paradiso, "The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance", *Journal of New Music Research*, Vol. 28, No. 2, pp. 130-149.

[16] http://www.toysymphony.org

[17] Aimi, R. and Young, D. "A New Beatbug: Revisions, Simplifications, and New Directions," *Proceedings of the International Computer Music Conference (ICMC 2004)*, Miami, Florida, November 1-6, 2004.

[18] http://plork.cs.princeton.edu

[19] Wang, G. and P. R. Cook. "ChucK: A Programming Language for On-the-fly, Real-time Audio Synthesis and Multimedia," *Proceedings of ACM Multimedia 2004*, New York, NY, October 2004.

[20] http://www.bodymedia.com/index.jsp

[21] http://lifeguard.stanford.edu

[22] Sung, M., Marci, C., and Pentland, A., “Wearable Feedback Systems for Rehabilitation,” *Journal of NeuroEngineering and Rehabilitation* 2005, 2:17.

[23] http://www.zigbee.org/

[24] http://www.xbow.com/

[25] http://www.dust-inc.com/

[26] http://www.moteiv.com/

[27] Lorincz, K, et.al. “Sensor Networks for Emergency Response: Challenges and Opportunities,” *IEEE Pervasive Computing,* October-December, 2004, pp 16-23.

[28] http://www.turbulence.org/blog/archives/cat_wearables.html

[29] Knapp, R.B. and Lusted, H.S. “Designing a Biocontrol Interface for Commercial and Consumer Mobile Applications: Effective Control within Ergonomic and Usability Constraints,” *Proceedings of HCI International*, Las Vegas, NV, July 22-27, 2005.

Perturbation Techniques for Multi-Performer or Multi-Agent Interactive Musical Interfaces

Matthew Burtner
VCCM / University of Virginia
112 Old Cabell Hall
Charlottesville, VA 22903 USA
mburtner@Virginia.edu

ABSTRACT
This paper explores the use of perturbation in designing multi-performer or multi-agent interactive musical interfaces. A problem with the multi-performer approach is how to cohesively organize the independent data inputs into useable control information for synthesis engines. Perturbation has proven useful for navigating multi-agent NIMEs. The author's *Windtree* is discussed as an example multi-performer instrument in which perturbation is used for multichannel ecological modeling. The Windtree uses a physical system turbulence model controlled in real time by four performers.

Keywords
Multi-performer, multi-agent, interface, mapping, movement, music composition, perturbation

1. INTRODUCTION
The study of multi-agent systems focuses on systems in which intelligent agents cohere around a particular problem or task . Such systems can incorporate asynchronous computation, independent and varied modes of action for each agent, aspects of randomness at the global control level, and decentralized data structures [14]. These characteristics can be desirable for musical systems in which rich and multidimensional control data can be rendered as organic and complex music. Here the application of multi-agent design techniques into NIME development serves as a tool for organizing musical systems. This is done using perturbation, allowing mutual dependency between the performers, bounding their performance by the group behavior. This approach allows for expressive micro-level data to be pulled by larger tendencies of the whole group. Such nested control structures may provide new techniques for mapping.

2. MULTI-PERFORMER/MULTI-AGENT SYSTEMS
As in artificial intelligence, in the area of interactive computer music, agency is largely organized around single-performer systems, structures in which data input is centralized and synchronized. Even in multi-performer interactive music the system is often separated into independent but coexisting agents. In addition, interactive single-agent systems may be multi-modal (that is having more than one type of control input), but these modes are synchronized and codependent. Multi-agent and multi-performer interactive systems however, offer the possibility for new complex behaviors in interactive musical interfaces. Specifically they can yield complexly organic structures similar to ecological systems.

Given the rapidly expanding field of interactive interfaces and real time synthesis systems, relatively few multi-performer approaches exist. The merger of digital controller data into complex mapping strategies suggests multi-performer controllers, but relatively few have been developed. Thus the area of mapping multi-performer controller data remains relatively unexplored. Recent developments in this area point to new possibilities for musical creation. New interfaces such as the WiSe Box [16] and WISEAR [3] are specifically designed as interfaces for multiple performers, primarily because several interfaces can be used simultaneously on stage with data sent wirelessly to a single synthesis engine. These interfaces suggest new multi-performer possibilities arising in the near future. The Tooka [7] beautifully couples two performer input by using a single pressure sensor mounted at the center of an open tube requiring the regulation of air flow from performers positioned on each end. The Tooka is the most codependent and successful of the multi-performer NIMEs this author has experienced.

Musically, groups such as the Hub's data network project [2] and Sensorband's SoundNet project [4], extend multi-performer systems into the field of NIMEs. Examples of earlier work with multi-performer electroacoustic performance include Stockhausen's *Mikrophonie 1* in which multiple performers play, sample and mix a single tam tam in a classic example of the multi-performer instrument.

Musical Multi-agent systems shift focus more broadly to the discreet agency of performers whether they be artificial intelligences or human performers. The expanded notion of an "agent" is articulated here for musical purposes to define a system in which computers or combined human-computer intelligences share musical decision making responsibility.

Allowing multiple-agency in the design of NIMEs introduces problems of mapping because of the potential amount of conflicting control data. Smoothing the data is helpful to reduce noise and create bounded input, but it works against the inherent richness of the multi-performer system and is therefore not useful beyond a certain point. Perturbation is thus proposed as a technique for mitigating control data.

3. FORMATIVE WORK
The work here with multi-performer and multi-agent systems is inspired by a body of research in the field of artificial intelligence in combination with the author's experience performing with interactive NIMEs.

NIME 06, June 4-8, 2006, Paris, France.

3.1 Distributed Artificial Intelligence

Outside the field of music, multi-agent collaborative systems have been explored extensively as distributed artificial intelligence (DAI) [3]. Applications of DAI include modeling market behavior [15] in which many independent variables coalesce in a complex outcome. The multi-agent approach has proven useful in modeling swarming behaviors such as those exhibited by bees, birds and ants [7]. Distributed agency, suggested from this work, is also useful for multi-performer musical systems in which the synthesis engine is not aware of the performer's individual goals even if these goals are shared.

3.2 Multi-User Interfaces

Many of the challenges of multi-performer systems can be related to research in multi-user interfaces. For example, we can see in the *Cognoter* module of Xerox Parc's *Colab* a relevant example. The Colab, collaborative environment in which computers facilitate human team interaction, implements a WYSIWIS (What you see is what I see) foundational abstract.

3.3 Instrument Controller Substitution

Research into Instrument Controller substitution [5] explored the effects of combining the control interface of one instrument with that of a different synthesis instrument, such as performing a bowed string with the keys of a wind instrument. An interesting byproduct of this work was the use of mappings involving multiple modes of a controller yielding controlling a single synthesis parameter. Such many-to-one mappings suggested an approach to mapping in which data streams were merged prior to assignment and allowed to exert a mutual influence on one another. In the Metasax composition *S-Trance-S* [6], eight keys of the saxophone function as continuous input to the string model.

3.4 MICE: a multi-performer ensemble

MICE (Music for Interactive Computers Ensemble) has been exploring multi-agent systems for several years at the University of Virginia (since 2001). The group grew out of this author's Interactive Media seminar as an exploration of network performance, data management, mapping, artificial intelligence and shared expressive structures. MICE involves several computers and human performers with shared responsibilities. The model of human-computer interaction in MICE is viewed as a multi-agent approach to expressive sound, designed to share agency between performers and between human intentionality and computer intelligence.

4. PERTURBATION MAPPING

Multi-performer NIMEs introduce problems of mapping because of the large amount of control data generated for a single task. The inherent richness of the multi-performer system can become overwhelming if some relationship between the performers is not defined at the mapping level. At the same time, smoothing and interpolating between individual agents can loose the richness and complex dynamic of the system.

Perturbation can be used as a technique for navigating multi-performer human-computer interfaces. Perturbation is the use of mitigated influence from one agent on the others. Individual performers or agents simultaneously influence and depend on the others. In such systems, the data input from the modes of action are operated on as a group, and this new value is used to attenuate the input data from the individual performers or agents such that some operation, T, acts as a mitigating force on each of the other performers.

For example, let

$$T = I_1 + \Delta_{I_2} + \Delta_{I_3} \ldots + \Delta_{I_m}$$

Where m is the number of inputs, I. And let Δ represent the difference of the input data in time a window t such that

$$\Delta = \frac{y_2 - y_1}{x_2 - x_1} \text{ where } x_2 - x_1 = t,$$

x_1 being defined as j_α and x_2 being defined as $j_\alpha + t$;

and where $y_2 - y_1$ defines the change in sensor input.

Thus a change of Δ_1 occurs inside window t_1. And let T be such that

$$I_T = I_1 + \sum_{m=2}^{m} \Delta_{I_m}$$

where the perturbation function T sends a collection A – the set of all inputs ($I_1 + I_2 + I_3 + \ldots I_m$) – to the master signal B, expressed as I_T. As such we can view the system as a perturbation machine traversing the vector spaces A and B over Δ_{I_T} in time t where I_T is the master output signal, the result of the many to one mapping as illustrated in Figure 1.

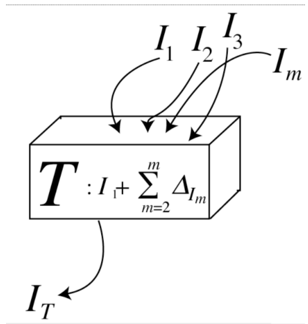


Figure 1: a generalized approach for applying perturbation to many-to-one mappings.

In this example, the effect of the perturbation T decreases for each iteration of m. This mitigating influence between agents can be carefully controlled. The method shows how an expanding system may still incorporate perturbation without necessarily losing distinct agency. Through a variety of asymptotic techniques (realized as T) applied to performer agency (such as those discussed in [10]) a wide range of perturbations are possible.

5. NEW INTERFACE APPLICATIONS

The large-scale interactive multimedia work *Windcombs/Imaq* for voices, instrumental ensemble, movement art, video, and 6-channel computer sound was composed at IRCAM as a commission for the Quincena Festival/Musikene, San Sebastian, Spain. The piece required the creation of a new musical interface called the *Windtree,* an interactive light sculpture for four performers whose movements are combined into a physical system model for sound synthesis (Figure 2).

The Windtree is a light sculpture made out of metal, translucent plastic, and cloth with a light projecting from the inside.

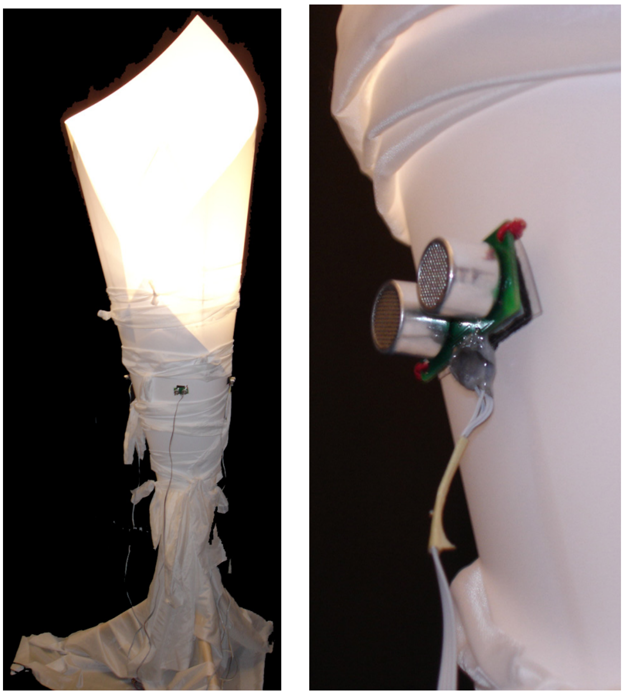


Figure 2: The *Windtree* is an interactive multi-performer light sculpture. The left image shows the sculpture from a distance. The right detail view shows one of the directional sensors

The instrument uses directional sensors pointing in four directions from the cone of the sculpture to capture movement of performers situated on each side. The Devantech SRF04 ultrasonic range finder was used because it provides distance measurements in the desired range (8cm to 2m), and requires low voltage.[1] Four S4F04s are used, each pointing in a different direction. This configuration allows the continuous measurement of four distinct performers, virtually *tethered* in the four directions from the sculpture. The beam pattern of the SRF04 (figure 3) shows that if the performer strays from a direct line from the sculpture the sensor will not give good data.

The high degree of directionality is not a strength in many movement-based applications. The tethering phenomenon limits the use of this sensor for human motion, especially in contexts in which the movement may drift out of the beam pattern. However, this directionality allows for the use of multiple sonars without cross-talk interference. The use of multiple sonars pointing in different directions is idiomatic only for an open space or on the stage, because in enclosed spaces the reflections from the walls create pulse interference noise between different beams.

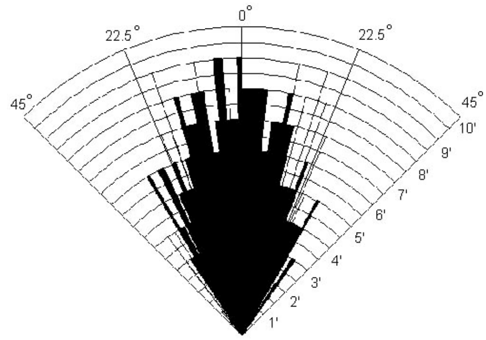


Figure 3: Devantech SRF04 Beam Pattern showing the virtual tethering of the sonar, a highly directional sensor. [12]

This tethering aspect of the SRF04 was desirable for the Windtree because the performer's individual movements are coordinated to an ecological wind model, supporting an artistic notion of directionality related to the North/South/East/West winds in the story related by *Windcombs/Imaq*.

The physical design of the sculpture emulates a portal. In the story of the wind, relayed in the composition *Windcombs/Imaq,* a shaman travels to the four directions and looks through portals into different worlds each with their own character. The shaman sews the portals closed, allowing some of the wind to come through. The cloth bindings around the sculpture evoke this sewing or constricting of the wind, a constriction that is further evoked by the turbulence model in the synthesis.

The Windtree controller is thus closely coupled with a specific synthesis engine, which is further specific for a particular composition.

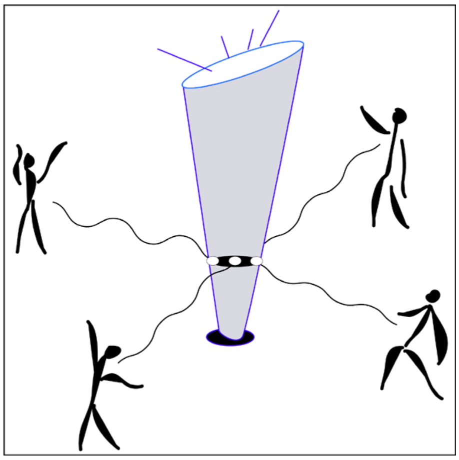


Figure 4: Beam pattern tethering of four performers to the Windtree

6. SYNTHESIS APPLICATIONS

The Windtree sensor data is fed into an ecological physical system model employing wind turbulence modeling. The wind model uses filtered white noise and involves the definition of and interaction between independent, variable bursts of energy. *Levels of nested and banded randomness applied to frequency, duration and amplitude define gustiness*. Turbulence is thus defined as the energy variability within the constraints of the input settings.

---

[1] The published range of the SRF04 [12] is 3cm to 3m but because of noise introduced as a result of the beam dispersal pattern, a practical usable range for dance was found to be closer to 2 meters.

6.1 Turbulent wind modeling

Methods for modeling wind turbulence are defined by scattered documents in the areas of aerospace and are largely concentrated into two approaches, called the Von Kárnán and Dryden approaches [9] [11]. Both of these rely on modeling the effects of wind by using guidelines developed by the U.S. Military, found in the MIL-F-8785C and MIL-HDBK-1797 guidelines. These documents list the differential finite digital filter equations and transfer functions associated with the two approaches. In both, the approach to turbulence modeling involves passing white noise through a series of forming filters.

6.2 Parameter reduction

The difficulties in modeling true turbulence further include issues of angle trajectory and altitude. In the model, these parameters are static, leaving a simplified range of parameters to be controlled by the Windtree.

6.3 Modularity

There are four modules corresponding with the four Windtree performers (Figure 5). These four modules are related to the libretto of *Windcombs/Imaq* that specifies the "Four Winds" as characters in the drama. The main interface (Figure 5) reveals how the independent agency of the four performers is maintained while the perturbation is applied in the mapping subpatch.

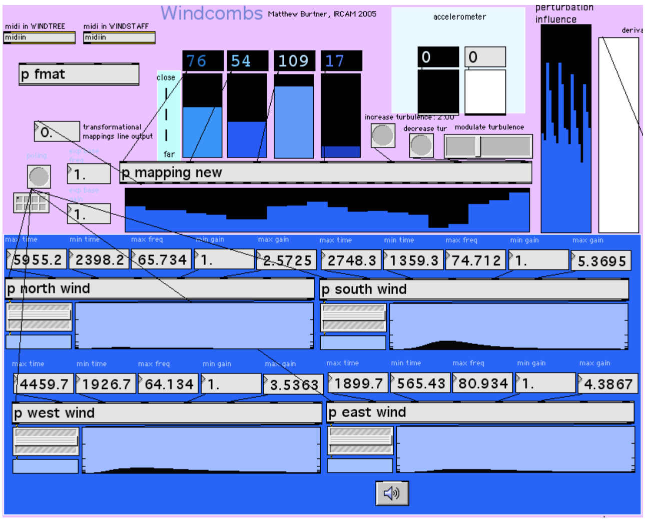


Figure 5: The Windtree interface in Max/MSP.

Each performer's movement is captured independently and sent into one of the wind models. Each wind module contains a distinct turbulence instrument (Figure 5).

Each module has the possibility of continuously modulating the

1) Maximum time values range of an individual gust
2) Minimum time values range of an individual gust
3) Upper limit of the frequency
4) Lower limit of amplitude
5) Upper limit of amplitude

In addition the log interpolation can be set for the frequency and amplitude creating different kinds of interpolations.

The eight filters are then cascaded into a parametric filter. This filter defines the characteristic of that single wind. In the mapping stage, the four winds are combined using perturbation to create the final dynamic turbulent system.

7. MAPPING APPLICATIONS

The Windtree mapping strategy involved several considerations. The nature of the interface (only four inputs) meant that a one to many mapping strategy would be needed to control the model. Specifically a one to five configuration was used. At the same time, the use of perturbation acts as a many to one system as described above. Finally, the composition required a system that would evolve over time. Global conditions of the instrument evolve and this is accomplished by interpolating between matrices over time. Dynamic matrix-based mapping allows for complex data structures such as sequences, mapping matrices, dictionaries, etc. to be passed between objects in Max/MSP. These tools were implemented as FTM by the Real Time Applications (ATR) research group at IRCAM [13] and [1].

7.1 One-to-many mapping

Each of the four performers has only one continuous control input. The synthesis model however requires five continuously varying streams of data, which are further used to control eight independent parameters each. The four input variables thus affect 160 parameters of the synthesis engine.

Figure 6: Four dancers performing on the Windtree. Using a one-to-many mapping, the four input values are mapped into 20 independent parameters controlling 160 variables of the wind turbulence model

7.2 Matrix interpolation

Interpolation between matrices allows the mappings to undergo continuous transformation, changing the effect each input is having on the synthesis engine. The impermanence of the mapping generates constant and gradual change in the system, a characteristic of other environmental models.

7.3 Multi-performer perturbation

Matrix interpolation brings a unity to the multi-performer system by providing the system with a global tendency defined by the mapping. In order to create cohesion, perturbation is used to mitigate the independence of each performer.

Perturbation is applied to the system in an attempt to create cohesion in the multi-performer instrument, and to increase the complexity of turbulent interaction at the synthesis level.

In *Windcombs/Imaq*, each input sensor (I_m) is also a mitigating factor in the determination of the other sensor's value (as I_j) such that each T_a is defined as $((I_1 + I_2 + I_3 + I_4) / 4) + I_m$ for window t at Δ_a.

The output is thus a weighted sum of the inputs such as:

$$T_a = I_1(3/4) + ((I_1 + I_2 + I_3 + I_4)/4)$$

The *real* variable for each input closely follows one of the performers but is shaped by the group as a whole.

It is important to reiterate that this example represents one possibility for *T,* among many possible perturbations. The generalized approach discussed above will support multiple functions of *T* including more complex, transformational operations.

8. Future Directions

In *Windcombs/Imaq,* the performers were considered to be equal and nonhierarchical. More complex asymptotic techniques could be implemented for multi-performer systems. Even in applications employing simple multi-modal systems, perturbation could be used to define gesture from codependent inputs, a situation that reflects certain musical interfaces but is not normally considered in the mapping stage. The many to one mapping for example could benefit from such a perturbation strategy. Future work will involve designing more complex operations such as implementations defining the perturbation machine *T* as a complex of T_1, T_2... etc. such that modulations of the operations themselves can be applied to musical parameters.

9. Acknowledgments

Special thanks to Norbert Schnell, Frédéric Bevilacqua, and the entire Real Time Applications Team (ATR) at IRCAM who created FTM, Gabor and MnM and who hosted me as an invited researcher at IRCAM in 2005 where *Windcombs/Imaq* was developed.

10. REFERENCES

[1] Bevilacqua, F.; Muller, R. and Schnell, N. 2005. "MnM: a Max/MSP mapping toolbox" *Proc. of the 2005 NIME Conference.* Vancouver, Canada.

[2] Bischoff, J., Gold, R. & Horton, J. (1978). "Microcomputer Network Music." *Computer Music Journal* 2(3):24-29. (Reprinted in C. Roads and J. Strawn, eds. 1985. Foundations of Computer Music. Cambridge, MA: MIT Press.)

[3] Bond, A. H. & Gasser, L. 1988. *Readings in Distributed Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann.

[4] Bongers, B. 1998. "An Interview with Sensorband". *Computer Music Journal.* Volume 22, Number 1: Spring 1998. MIT Press, Boston, MA.

[5] Burtner, M., and Serafin, S. 2002. "The Exbow Metasax: Compositional Applications of Physical Models Using Instrument Controller Substitution." *Journal of New Music Research*, vol. 31. Swets & Zeitlinger, Lisse, The Netherlands.

[6] Burtner, M. 2002. "The Metasaxophone: Concept, Implementation and Mapping Strategies for a New Computer Music Instrument." *Journal of Organized Sound,* Volume 7 No 2, Cambridge Press, York, England.

[7] Fels, S. & Vogt, F. 2002. Tooka: Exploration of Two Person Instruments. *Proc. of the 2002 NIME Conference.* Dublin, Ireland.

[8] Fléty , Emmanuel. 2005. "The WiSe Box: a Multi-performer Wireless Sensor Interface using WiFi and OSC." *Proc. of the 2005 NIME Conference.* Vancouver, Canada.

[9] Gage, S. 2003. "Tools Simplify Wind Turbulence Modeling." COTS Journal, November 2003. The RTC Group. San Clemente, CA

[10] Holmes, M. 1998. *Introduction to Perturbation Methods.* Springer. New York, NY.

[11] Lungu, D. and Gelder, P.V. "Characteristics of Wind Turbulence with Applications to Wind Codes." *Proc of ESREL 2003 (European Safety and Reliability Conference).* Maastricht, The Netherlands.

[12] Parallax Inc., 2003. *Devantech SRF04 Ultrasonic Range Finder (#28015).* http://www.parallax.com/

[13] Schnell, N.; Borghesi, R.; Schwarz, D.; Bevilacqua, F.; Müller, R. 2005. "FTM – Complex Data Structures for Max." *Proc of ICMC 2005.* International Computer Music Association. Barcelona, Spain.

[14] Stefik, M.; D.G. Borrow; S. Lanning.; D. Tatar; and G. Foster. 1986. "WYSIWIS Revised: Early Experiences with Multi-User Interfaces". *Proc of CSCW'86,* Computer-Supported Cooperative Work, Austin, Texas.

[15] Sycara, Katia. 2002. "Multiagent Systems." *Artificial Intelligence Magazine,* 19-02-007. American Association of Artificial Intelligence, Menlo Park, CA.

[16] Sycara, K.; Decker, K.; and Zeng, D. 1998. Intelligent Agents in Portfolio Management. In *Agent Technology: Foundations, Applications, and Markets,* M. Wooldridge and N. R. Jennings eds. 267–283. Berlin: Springer.

[17] Topper, D, and Swendsen, P. 2005. "Wireless Dance Control: PAIR and WISEAR." *Proc. of the 2005 NIME Conference.* Vancouver, Canada.

Sensemble: A Wireless, Compact, Multi-User Sensor System for Interactive Dance

Ryan Aylward
Responsive Environments
MIT Media Laboratory
20 Ames St.
Cambridge, MA 01239
1-617-452-5647
aylward@media.mit.edu

Joseph A. Paradiso
Responsive Environments
MIT Media Laboratory
20 Ames St.
Cambridge, MA 01239
1-617-253-8988
joep@media.mit.edu

ABSTRACT
We describe the design of a system of compact, wireless sensor modules meant to capture expressive motion when worn at the wrists and ankles of a dancer. The sensors form a high-speed RF network geared toward real-time data acquisition from multiple devices simultaneously, enabling a small dance ensemble to become a collective interface for music control. Each sensor node includes a 6-axis inertial measurement unit (IMU) comprised of three orthogonal gyroscopes and accelerometers in order to capture local dynamics, as well as a capacitive sensor to measure close range node-to-node proximity. The nodes may also be augmented with other digital or analog sensors. This paper describes application goals, presents the prototype hardware design, introduces concepts for feature extraction and interpretation, and discusses early test results.

Keywords
Interactive dance, wearable sensor networks, inertial gesture tracking, collective motion analysis, multi-user interface

1. INTRODUCTION
Several wireless interfaces have been developed to capture dance gestures over the last decade or two. Some have been sensor systems built into shoes, such as the 1980's Taptronics, featuring piezoelectric pickups at the toe and heel [1] and Expressive Footwear by our group at the MIT Media Lab [2]. Originally realized in 1997, this system was an early implementation of a dense, multimodal wireless sensor cluster (now becoming common in sensor networks) that measured 16 variables including many degrees of both contact and free-gesture control. Other examples of wearable dance instrumentation typically use bendable sensors that span primary joints such as the elbows and knees. Architectures of this sort have been introduced by DIEM in Aarhus [3] and by Mark Coniglio of Troika Ranch in New York [4]. Although these systems have become wireless, they employ a single radio in a beltpack or backpack, hence the various sensors need to be tethered across the body to this central dispatcher. Extreme versions of these types of wearable joint-bend interfaces can be found in full-body motion capture outfits for computer graphics, and flexible fiber-optic angle-sensing systems such as ShapeWrap by Measurand [5].

The systems above were developed for single subjects, and many do not scale well to ensemble performances. For instance, the bandwidth of the Expressive Footwear system was limited 60 Hz full-state updates for two shoes. Furthermore, no provision was included to sense upper body or arm motion. Some of the centralized backpack systems enable more than one dancer to be accommodated, but the wires running from various sensor locations to the central body-worn transmitter are cumbersome.

Another approach to gesture tracking for dancers avoids any body-worn hardware by exploiting computer vision, processing video from a camera or cameras watching the stage. This technique is now well established, and platforms like the Very Nervous System [6], Eyes Web [7], Big Eye, and Jitter are used by many composers. The prevalence of optical tracking methods has even prompted some artists to develop their own video analysis tools, e.g., [8,9]. This approach is processor intensive, and although the underlying technology and algorithms are steadily improving, computer vision is further limited by constraints on lighting and choreography; robustness to occlusion and background noise remains problematic. Hence, obtaining multiple relevant features reliably from a dance ensemble in a performance setting can be difficult.

Accordingly, we have developed a system of compact wireless inertial sensors that can be worn on the hands and feet of a group of dancers to enable real-time gesture tracking over the entire ensemble. This approach has advantages over other techniques in that each point of measurement has a dedicated wireless connection, the system easily scales to a flexible number of performers and number of points of measurement on the body, does not suffer from occlusion, and provides sensor data which is immediately relevant to features of human motion.

2. GOALS
The motivation for this project is the recent opportunity to leverage low-power, high-bandwidth RF solutions and

NIME 06, June 4-8, 2006, Paris, France.

compact inertial sensors to create a wearable wireless motion sensing system meeting the demands of many points of measurement and high data rates. Our goal is to implement such a system for an interactive dance ensemble, which is in some ways an ideal situation for pushing high performance requirements. A highly active environment of human motion demands an unrestricting yet sturdy wearable design. Obtaining detailed information about the movement of the human body and the interaction of multiple human bodies demands many points of measurement. Most importantly, using this information as a vehicle for interactive performance, specifically with musical feedback, demands rapid data collection and analysis to achieve a response with a sufficiently low latency. In the broader scope, we hope to test the applicability of this system to other applications, such as analyzing the dynamics of team sports, physical therapy, biomotion measurement and analysis, or personal physical training.

3. HARDWARE DESIGN

The current hardware design has its roots in the Stack [10], a modular system, including full IMU card, developed by our research group several years ago as a compact and customizable alternative to our earlier Expressive Footwear design. However, the data radio used at the time was limited to only 115 kbps, far too low for our application. Assuming we would like to outfit an ensemble of five dancers wearing sensors on wrists and ankles, with full state updates at 100Hz, the inertial sensors alone generate:

$$6\text{sensors} \times 12\text{bits/sensor} \times 20\text{nodes} \times 100\text{Hz} = 144\text{kbps}.$$

If we wish to transmit additional information from the capacitive sensors, and account for the increased overhead costs associated with sending small frequent packets for low-latency, five dancers could easily require up to 400kbps in practice.

Although compact sensor clusters have been developed at other institutes, none have the characteristics that we need in terms of combining low power and small size with such high data rates. Motes are quite established for sensor networks, but most support mainly peer-peer routing at lower data rates than needed here. Likewise, the Smart-Its and its descendants [11] are designed to work at data rates similar to the Stack. Flety and collaborators at IRCAM [12] have built wireless sensor networks that use a similar transceiver as used in the Stack (and hence also exhibit limited data rate) and others that use the WiFi 802.11 standard, which tends to be much too power hungry for efficient continuous operation with a modest battery. Emmanuel Tapia of the MIT Media Lab has designed very compact wireless accelerometer sensors capable of higher data rates [13], but our application requires more sensor degrees of freedom.

The design presented here includes a full six axis IMU, node-to-node capacitive proximity sensing, and flexible expansion capabilities, combined with a low power 1Mbps radio. The sensor node (Fig. 1) measures 4cm × 4cm × 2cm, not including the protruding antenna and external battery pack. As shown, with the battery included, the weight is approximately 45 g. We chose to decouple the battery from the main circuit board, so that it could be affixed to the strap rather than adding to the bulk of the sensor package. This makes the node more comfortable to wear, provides easy access to the battery, and allows for flexibility in the choice of battery pack.

The nRF2401A data radio we utilize is a small, low power, 2.4 GHz device providing up to 1Mbps data rates. Our communications protocol is a TDMA scheme [14] in which a basestation polls the network for data at the sampling rate, and each node responds within a preprogrammed time slot. The basestation then transmits the data to a central computer via USB for processing. Using this scheme, one basestation can handle full state updates at 100Hz for over 25 nodes. This is a significant performance improvement over previous designs. The workable RF range on these devices appears to be on the order of 50 feet, depending on the local RF environment.

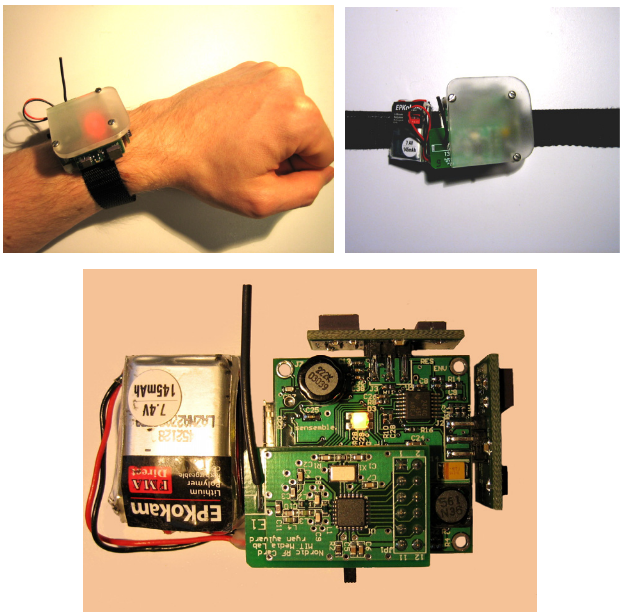


Figure 1. Sensor node on wrist (upper left), removed (upper right), and exposed circuit board (bottom).

The IMU is made up of Analog Devices ADXRS300 rate gyros and ADXL203 accelerometers, as well as associated analog circuitry. Sensor signals are collected by the 12-bit analog to digital converter built into the onboard processor, a TI MSP430F14x. This microcontroller was favored because of its low power consumption, capable A/D, and ample I/O, as well as its use in several of our group's ongoing projects.

The node-to-node capacitive proximity sensor operates by alternating transmit and receive modes on each of the sensor nodes, with only one node transmitting at a time, while the body is grounded. Because of timing constraints, it is not feasible to record measurements for every pair of nodes; rather, several simultaneous transmit nodes and several concurrent receive nodes can be selected in software. During transmit mode, the microcontroller drives an LC oscillator, which generates a high amplitude pulse (tens of volts peak-to-peak) at 91 kHz. During receive mode, the pulse is picked up by the receiving node, amplified, and sampled in quadrature to estimate its amplitude without the need for phase coherence. The nodes are able to use the same electrode for both transmit and receive modes, thanks to an efficient amplifier circuit inspired by the School of Fish, an electric field sensing tool designed several years ago by a former Media Lab student [15]. Capacitive sensing requires an electrode with sizeable area – this could possibly be integrated into the strap securing the sensor package to the body using highly conductive textiles such as Bekiweave [16].

Additional capabilities include a free digital input for interfacing with a Polar heart rate monitor, a free SPI interface for connecting with other digital devices, and a free analog input with associated signal conditioning circuitry for handling an additional resistive sensor, such as a pressure sensor, bend sensor, or light sensor. All of these optional signal lines are broken out to a compact expansion port, which also acts as the programming interface.

Power consumption is always of prime importance in the design of wireless devices; the power source tends to be the largest and most cumbersome component of the system. Unfortunately, our desire to operate continuously with three rate gyros prevents this design from meeting traditional low-power requirements. Each gyro may consume up to 30mW, and their slow setup time prevents them from being power cycled. The data radio is also comparatively power hungry, consuming up to 60mW in receive mode and 40mW in transmit mode, but this can be managed in code by minimizing the amount of time spent in active modes. Ultimately, we chose to operate the system with lithium polymer batteries because they are lightweight, compact, and rechargeable. With two compact 145mAh cells in series, as pictured above (Fig. 1), the node can operate for four hours on one charge.

4. RESULTS

The major advantage of having enough bandwidth to operate multiple sense points on multiple wearers simultaneously is the ability to obtain detailed information about correlated activity within a group. In the context of a dance ensemble, time and spatial correlations can be used to determine which dancers are moving together, which groups are leading or lagging, or perhaps which dancers are responding to one another with complementary movements. With this in mind, our preliminary analysis focuses mainly on the feasibility of extracting simple features that can be used to describe general group dynamics.

4.1 Correlated Motion

Previous work has shown that cross-covariance can be used to express both time separation and spatial similarity of gestures performed by multiple users [17]. For example, Figure 2 illustrates pitch gyro data for the hands of three subjects performing a similar gesture in sequence. The locations of the peaks in the associated cross-covariance curves (calculated with respect to subject 1) give the time lags between the three events. In addition, the height of a peak gives a measure of how well the signal shapes are correlated. In this way, we can also obtain a sense for the spatial similarity of the events. Here, subject two does a slightly better job at mimicking the motion of subject one.

One problem with cross-covariance as a feature is that it requires a complete segment of data to calculate. In a streaming situation, windowed cross-covariance must be used, where the window size is chosen to make a tradeoff between latency and the maximum time separation that can be expressed. A feasible use of cross-covariance requiring a short window might be to follow how closely dancers synchronize to music or to a leader, where the delays between their correlated motions are expected to be within a second.

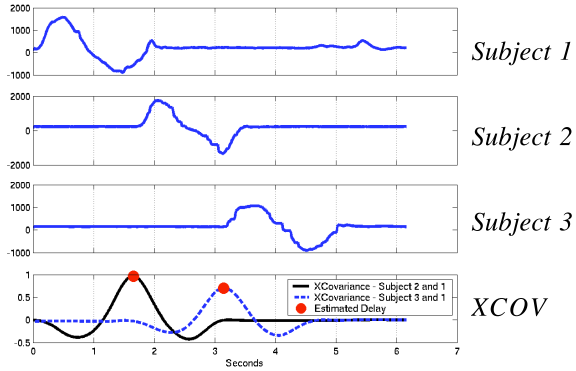


Figure 2. Raw data for hands raised and lowered in sequence (only the pitch gyro is shown) and resulting average cross-covariance.

To test this idea, six sensors were given to three dancers participating in a ballet lesson; each wore one on the right wrist and one on the right ankle. The class then performed an exercise involving a repeated sequence of leg swings executed in unison, to music. Although they were roughly in time with the music, the dancers were not necessarily looking at each other or at an instructor, creating a small but clearly visible delay in their motions (the rehearsal was documented on video for reference). Figure 3 shows a portion of the raw data collected from the leg of each dancer. Because there was very little arm motion associated with this exercise, only leg motion is discussed here. The area from about 35 to 65 seconds corresponds to the synchronized sequence of swings made with the right leg.

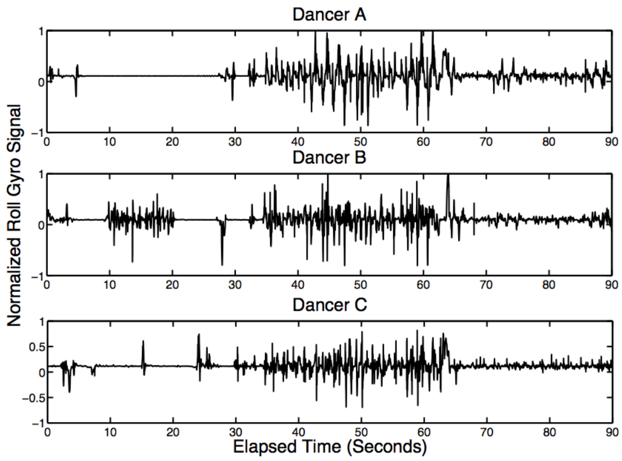


Figure 3. Selected raw data from the ankles of three ballet students performing a sequence of leg swings in unison.

Figure 4 shows the result of windowed cross-covariance analysis on this data segment with a window size of 1 second and a step size of 0.25 seconds. That is to say, at each interval of 0.25 seconds, a window of data was considered, the cross-covariance vector was computed individually for each sensor value, and then the individual vectors were averaged to produce a result. Note that the area of peak cross-covariance, shown in white, tends to waver around the baseline as time progresses. This is consistent with the dancers slowly leading and lagging with respect to one another by small amounts. Because the step size is small enough, individual leg swings and their synchronicity across the ensemble can be picked out. It is clear from the relatively stable middle plot that Dancer A and Dancer C were closely synchronized for the duration of the exercise, while Dancer B fluctuated from about 0.3 seconds ahead of Dancer A to 0.3 seconds behind Dancer A. This fluctuation reflects accurately what is visible in the video. Interestingly, it turns out that Dancers A and C were facing each other during the exercise, while Dancer B had her back turned to the others.

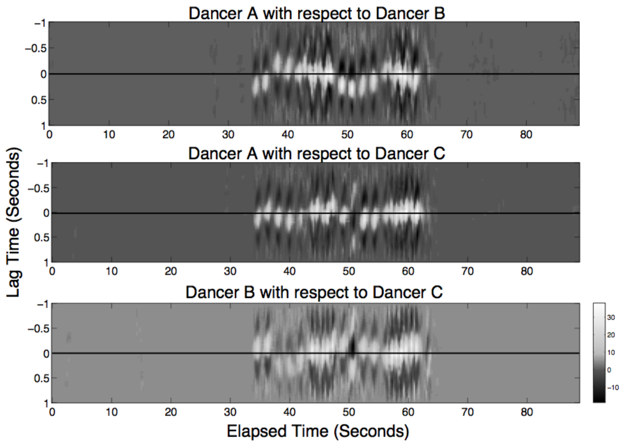


Figure 4. Windowed cross-covariance (averaged across sensor values) between pairs of dancers, for the data segment presented in Figure 3.

4.2 Quantifying Activity

In addition to extracting correlations between the activities of a group, it is important to obtain information about the properties of the activities being observed. These properties might include variations in the overall activity level of an individual or group at different time scales, principal axes of movement, or other features extracted during an interval of high activity.

One approach to activity measurement involves computing the average running variance for various combinations of sensors on individual nodes. If the separation between gestures is long enough, variance spikes can be used to delineate them. In other cases it might be useful to use a lowpass filter to obtain an envelope on the running variance, in order to determine slower trends in the activity level. For example, data was collected from the right wrist and ankle of a ballet student performing a sequence of motions in which slow kicks with the right foot transitioned into fast, tense kicks (in ballet terminology, petit battement). The full sequence is framed with a stylistic tension and release of the right arm at the beginning and end, respectively. Figure 5 shows a portion of the raw data from this segment along with four different activity envelopes obtained from the windowed variance of both upper and lower body movement. Accelerometer activity here denotes the average variance across the accelerometer axes, while rotational activity denotes the average across the gyro axes. One can clearly see a marked increase in activity as leg motion transitions to faster kicking. The role of the arm movement is apparent in the activity envelope as well.

Similar conclusions can be drawn from figure 6, which illustrates the activity envelopes of leg motion for each dancer during the period of correlated activity highlighted earlier in figures 3 and 4. Two areas of peak activity across the ensemble appear around 50 and 60 seconds into the sample, corresponding to repeated leg swings over the full range of motion from front to back and back to front. The general trend of activity is increasing over the segment from 30 seconds to 60 seconds, as the instructor urges the dancers to make each leg swing "successively higher". Finally, we see activity for Dancer B in the interval from 10 to 20 seconds that is not reflected in the movements of the other dancers, corresponding to a few "warm-up" leg swings by Dancer B. Comparison of the activity levels is all that is required to flag this unique period of activity, at which point it could be analyzed more closely, or used as evidence that Dancer B should be clustered in a different subgroup from Dancers A and C. Note that the cross-covariance analysis shown in figure 4 is unable to compare the warm-up leg swings with motion occurring later in time, because the window is only 1 second long. Given enough storage and computing power, one solution is to save interesting data segments for correlation with future data, or to monitor running cross-covariance on multiple time scales.

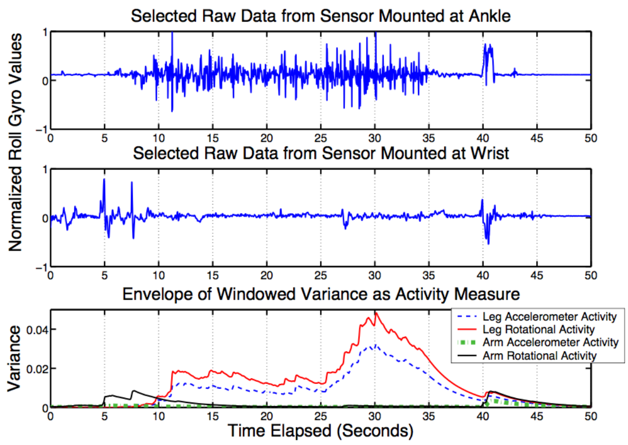


Figure 5. Selected data and resulting activity envelopes as dancer transitions from slow kicks to rapid tense kicks. Sequence of leg motions is framed by stylistic arm motion.

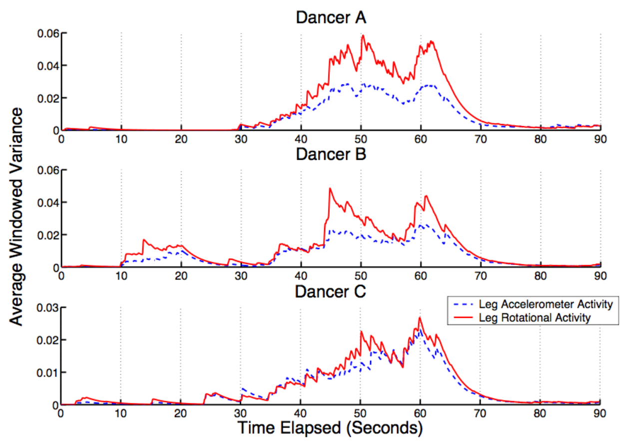


Figure 6. Activity envelopes for the synchronized leg movement highlighted in Figures 3 and 4.

Looking at figure 5 and 6, it would seem as if there is no reason to distinguish between accelerometer and gyro activity. Indeed, sensor activity on a single node is often highly correlated, because human motion is unlikely to occur along only one axis. The accelerometers are also subject to gravity and centripetal acceleration, so rotations will be picked up strongly in some cases. It should be possible to use the gyro signals to help isolate translational acceleration from other types of movement picked up by the accelerometers. However, if one wishes to identify specific classes of activity, it may be more important to compare motion along each axis than rotational versus translational motion. One approach is to keep track of which sensor has the highest variance on each node or on each individual, with the goal of analyzing activity one person at a time. A more efficient approach might be to create a group feature such as mean activity on each sensor axis, for each limb, across the entire ensemble, to determine the predominate axes of collective motion.

For example, figure 7 demonstrates the results of a group of three people raising and lowering their right hands in unison. The bottommost plot indicates the variance on each sensor axis for the right arm, averaged across all three subjects. Note that the average variance of the pitch gyro dominates. This supports our intuition that the act of raising and lowering the hand involves mostly a rotation in pitch. Extracting this information from average windowed variance may simplify the task of detecting specific gestures by determining which sensor signals are most important, or by defining a subgroup that is performing a similar gesture before applying heavier analytical techniques. One can also imagine a situation in which the correlation measurements discussed above are desired, but it is unclear who should be interpreted reasonably as a "reference" for the rest of the group. By comparing the average group variance to the individual variance, one can determine if the motions of a specific subject are characteristic of the entire group, or lie outside the norm.

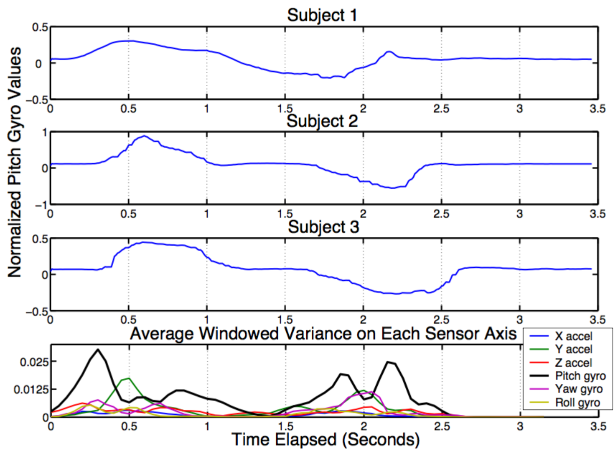


Figure 7. Right arm pitch gyro signals and windowed variance averaged across subjects for each sensor axis, as hands are raised and lowered in unison.

4.3 Capacitive Sensor

One of the limitations of small-scale inertial sensing is that it is extremely difficult to obtain a reference frame for any sort of position tracking. Yet, the shape of the body may be a more intuitive communication tool than the dynamics of the body. To supplement inertial data with information even as simple as "arms together" and "arms apart" would add significant depth to the interface. This was the idea behind the node-to-node capacitive proximity sensor.

Initial performance evaluations have determined that the capacitive system suffers from a very nonlinear response, which, coupled with high noise levels, limits its useful range (Fig. 8). Despite this, nodes grounded through the user's body can be sensed up to a spacing of 30cm with a 16cm^2 electrode. Nodes that do not share a ground, i.e. worn by different individuals, have reduced range but can still be detected. Past attempts at similar sensing systems have achieved better range, possibly due to higher voltage output on the transmitting electrode [15,18]. It may thus be possible to improve the performance with minor adjustments.

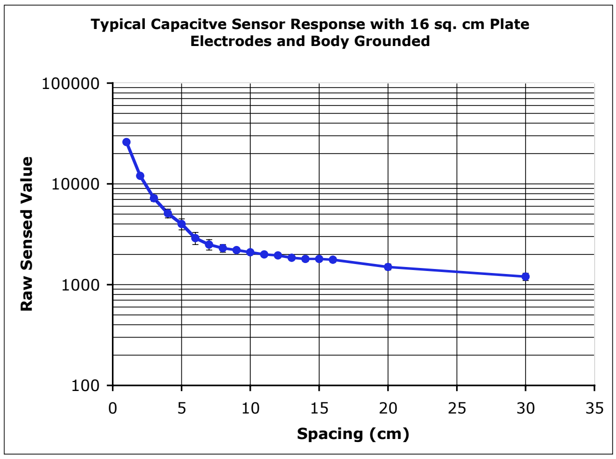


Figure 8. Typical response of the capacitive sensing system.

In the current version, because of the reduced sensitivity beyond about 5cm, it may not be efficient to transmit a full 12-bit value for every capacitive measurement. Rather, the signal should be compressed to fit a range of 8 or fewer bits with a more linear response. Another possibility is to use the existing response to form a simple one-bit indication of close versus distant. Until improvements can be made, this scheme fulfills the minimum requirements. In either case, data reduction will enable the transmission of capacitive measurements from more nodes without compromising the bandwidth available for higher priority sensor data.

5. GENERATING MUSICAL FEEDBACK

To demonstrate the utility of the system as a multi-user interface for interactive performance, it will be necessary to map extracted activity features to musical sound in a satisfying way. In a traditional free gesture interface, each degree of freedom might be mapped directly to a specific continuous control or set of event triggers. In this system, however, there are at least six degrees of freedom per node provided by the inertial sensors, and typically four nodes per user, making direct mapping impractical. Taking the first step towards a practical strategy for musical mapping in this framework, we have been focusing on forming descriptions of motion at the group level rather than at the individual level. As suggested above, simple group features can express a whole range of useful information, such as who is leading and who is following, degree of correlation across the ensemble, changes in activity level across the ensemble, the existence of subgroups or clusters within the ensemble that could be considered separately, principal axes of activity within subgroups, the location of an event unique to one individual, or relationships between levels of upper body motion and lower body motion. In turn, the treatment of the ensemble as an organic unit offers new possibilities for musical interpretation.

However, the potential amount of information expressed by these group features alone is still too large for a direct mapping to music. The problem can be simplified by interpreting group dynamics in the context of a specific piece. For example, the music can be generated from a loose framework or score designed alongside the choreography. At a given point in the score, one may be looking for a specific set of possible changes in the dancers' movements that signal musical events such as changing timbral qualities, the entrance of a new melodic line, or a shift to a new section. By placing contextual limits on the decision space, pattern recognition algorithms can be trained on a specific performance to streamline the control process. Although the dancers do not actually generate music directly under this model, they are able to freely control their progression through sections of the score, alter their interpretation of the context, and add embellishments. This approach should provide a balance between musical continuity and the sense of causality between the movements of the dancers and the generated sound, which is essential for an engaging interactive performance.

One limitation to address is the fact that many of the features discussed in this paper are slowly varying, or have a significant amount of latency associated with them. This is unsuitable for triggering sudden events or percussive sounds, as the human tolerance to latency in this case is quite low. It may be possible to train a state-based gesture-tracking model that would allow for rapid activity detection by predicting future states, but applying a simple threshold on one or more continuous features may be a better option.

6. CONCLUSIONS & FUTURE WORK

In this paper, we have presented a compact, wearable sensor system enabling real time collective activity tracking for interactive dance. The sensor node comprises a full 6-axis inertial measurement unit with supplementary capacitive node-to-node proximity sensing. Preliminary results demonstrate that our design is viable for analyzing a wide range of collective activity parameters in a dance setting. As the current ±1.7g accelerometer was found to have insufficient range to capture certain quick motions, it will be replaced by the ±10g ADXL210E. We also hope to increase the range of the capacitive sensor. Future work will focus on adding to the feature set developed here, assessing real-time operation with special attention to low-latency requirements, and developing a more specific framework for music generation with implementations in Max/MSP or PD.

7. REFERENCES

[1] di Perna, A. Tapping into MIDI. *Keyboard Magazine* (July 1988), p. 27.

[2] Paradiso, J., et al. Design and Implementation of Expressive Footwear. *IBM Systems Journal, 39(3&4)* (October 2000), pp. 511-529.

[3] Siegel, W. and Jacobsen, J. The Challenges of Interactive Dance: An Overview and Case Study. *Computer Music Journal 22, No. 4* (1998), 29-43.

[4] Coniglio, M. The MidiDancer system, see: http://www.troikaranch.org/mididancer.html

[5] See: http://www.measurand.com

[6] Zacks, R. Dances with Machines. *Technology Review* (May/June 1999), pp. 58-62.

[7] Camurri, A., et al. EyesWeb – Towards Gesture and Affect Recognition in Dance/Music Interactive Systems. *Computer Music Journal, 24(1)*, pp. 57-69.

[8] Downie, M. *Choreographing the Extended Agent: Performance Graphics for Dance Theater*. Ph.D. Thesis, MIT Media Lab, September 2005.

[9] Wechlser, R., Weiss, F., and Dowling, P. EyeCon – A motion sensing tool for creating interactive dance, music, and video projections. In *Proc. of the SSAISB Convention,* (Leeds England, Mar. 29, 2004).

[10] Benbasat A.Y. and Paradiso, J.A. A Compact Modular Wireless Sensor Platform. In *Proc. of the 2005 Symposium on Information Processing in Sensor Networks (IPSN)*, (Los Angeles, CA, April 25-27, 2005), pp. 410-415.

[11] Holmquist, L.E., et al. Building intelligent environments with Smart-Its. *Computer Graphics and Applications, IEEE* (Jan/Feb 2004), pp. 56-64.

[12] Flety, E. The WiSe Box: a Multi-performer Wireless Sensor Interface using WiFi and OSC. In *Proc. of NIME 05,* (Vancouver Canada, May 26-28, 2005), pp. 266-267.

[13] Munguia Tapia, E., et al. MITes: Wireless portable sensors for studying behavior. In *Proceedings of Extended Abstracts Ubicomp 2004: Ubiquitous Computing* (2004).

[14] Lovell, S.D. *A System for Real-Time Gesture Recognition and Classification of Coordinated Motion*. M. Eng. Thesis, MIT EECS Dept., Jan. 2005.

[15] Smith, J.R. *Electric Field Imaging*. Ph.D. Thesis, MIT Media Lab, February 1999.

[16] See: http://www.bekaert.com/bft

[17] Aylward, R., Lovell, S., Paradiso, J. A Compact, Wireless, Wearable Sensor Network for Interactive Dance Ensembles. Submitted to *Int. Workshop on Wearable and Implantable Body Sensor Networks*, November 2005.

[18] Paradiso, J., Gershenfeld, N. Musical Applications of Electric Field Sensing. *Computer Music Journal, 21(2),* (Summer 1997), pp. 69-89.

The ZKM Klangdom

Chandrasekhar Ramakrishnan
ZKM | Institut für Musik und Akustik
Lorenzstr. 19
D-76135 Karlsruhe
+49 (0)721 8100 1630
cramakri@zkm.de

Joachim Goßmann
ZKM | Institut für Musik und Akustik
Lorenzstr. 19
D-76135 Karlsruhe
+49 (0)721 8100 1633
jo@zkm.de

Ludger Brümmer
ZKM | Institut für Musik und Akustik
Lorenzstr. 19
D-76135 Karlsruhe
+49 (0)721 8100 1601
ludi@zkm.de

ABSTRACT
The Klangdom is an audio spatialization instrument developed at the Institut für Musik und Akustik at the ZKM. It is made up of 39 Meyer Sound loudspeakers hung on four sliding tracks, allowing for easy re-configuration of the speaker setup. The audience sits inside the Klangdom, which can be controlled either directly via a mixer, by externally developed software, or by a sequencer for sound movement, Zirkonium, developed at the ZKM. Zirkonium can accept and spatialize audio generated by other applications (even on remote machines) and can simulate the Klangdom over alternate speaker setups to aid composition and dissemination (e.g., in stereo or 5.1).

Keywords
Sound Spatialization, Ambisonics, Vector Based Additive Panning (VBAP), Wave Field Synthesis, Acousmatic Music

1. INTRODUCTION
At the Institut für Musik und Akustik (IfMA) at the ZKM | Zentrum für Kunst und Medientechnologie Karlsruhe, we are focused on commissioning, producing, and presenting electroacoustic music. To this end, we have a variety of ateliers, studios, and concert spaces, one of the better known being the Blauer Kubus (the Blue Cube), which is the IfMA's configurable environment for concerts as well as a sophisticated production facility. The Kubus has always been well equipped for presenting music in standard formats such as quadraphonic, 5.1, octophonic, or 16 channels. Though we have in the past had some speakers hung overhead, above the audience, the majority of the loudspeakers were positioned around the outside edge of the Kubus, limiting our ability to create a completely immersive sound experience. Moreover, as the number of channels increases, the control of a multi-loudspeaker environment seems to call for a different instrument than a mixing console. It was the desire to create such a system which led to the Klangdom project.

2. HISTORY
The spatial distribution of sound events is a parameter used in many musical genres and has played an important role in electroacoustic music in particular since its very beginnings.

As early as 1951, the studio at the Radiodiffusion-Télévision Française (RTF) employed a quadraphonic spatialization system with two front channels, one channel in the back, and one above the listeners and developed a controller for the system, the *pupitre d'espace*[16]. The pupitre d'espace was used as an instrument for live control of spatialization. As we can see in this example, spatialization is accomplished through the inter-working of three different components: loudspeaker position, technique for routing sound to the loudspeakers, and a controller for positioning the sound sources.

Figure 1. The Blauer Kubus

A thorough description of the development of multi-channel sound in electroacoustic music is unfortunately not possible in the space permitted. We refer the interested reader to Zvonar 2005[16] and Küpper 1984[9] for discussions of various aspects. To summarize the developments, we can broadly observe two approaches develop which we will label the acousmatic approach and simulation approach.

The acousmatic approach can be thought of as an extension of musique concrète. Similar to how musique concrète focuses on the capabilities of the tape machine as a source of generating musical material, the acousmatic approach focuses on the loudspeaker and its qualities as a way to organize sound in space. This is best illustrated by the acousmonium[3], developed at GRM in the mid-1970s. An acousmonium is made up of different types of loudspeakers distributed throughout a room. An acousmonium is usually played live by a composer/performer who routes the audio of a piece (often 2-channel) to the different loudspeakers, taking advantage of the sound reproduction characteristics and physical placement of the loudspeakers to realize a live performance. Another system with a similar philosophy is the sound dome of the sort championed by Leo Küpper[9] and exemplified by the German Pavilion at the World Expo '70 in Osaka, Japan. Though a sound dome is typically assembled from one particular brand/model of loudspeaker, in both it and the acousmonium, the composer/performer does not directly control sound position in 3D space as a parameter — rather she controls the routing of the sound to loudspeakers, which by their very location define the virtual position of the sound source.

The simulation approach, by contrast, uses 3D space as parameter, hiding the internal routing to the loudspeakers and uses signal processing and psycho-acoustic properties to produce the illusion of a sound emanating from a particular point. Whereas the acousmatic approach can be realized in the analog domain with just a mixer, the simulation approach more or less requires the intercession of a computer. One of the early pioneers of this approach was John Chowning, who in his 1971 article[2] described techniques for simulating moving sound sources over a quadraphonic speaker setup. The Spatialisateur or Spat[7] developed at IRCAM is an evolution of this idea and lets the user specify a sound source's position as well as its reverberation characteristics, which are also important for the perception of sound localization.

Other recent work along the same line has focused on Wave Field Synthesis and Ambisonics. Wave Field Synthesis[4] uses a large number of small loudspeakers to synthesize an approximation of the wavefront that would be observed were there a sound source at the specified position. Ambisonics similarly approximates the sound field at a point, but does so using ordinary speakers[6].

3. CONSTRUCTION

The goal for the Klangdom was to be able to bathe listeners inside the Kubus with a wash of sound from all directions. To accomplish this, it was important that we be able to place loudspeakers around the public, covering the entire 360 degrees of azimuth and 90 degrees of zenith (the top half of a sphere). However, flexibility was also a key consideration. We wanted the placement of loudspeakers to be configurable and the individual components to be usable outside of the Klangdom.

Figure 2. The Interior of the Kubus

Since we were working with an existing space, the Kubus, and did not have the luxury of building a new space for our spatialization environment, practical requirements played a major role in the design. Any infrastructure we built needed to augment and play well with our existing capabilities. This determined our choice of loudspeaker and scaffolding for the dome.

3.1 Speaker Type

Several factors influenced our choice of speakers. First of all, we wanted the flexibility to use the speakers in the context of both a simulation as well as an acousmatic strategy. We also wanted to be able to achieve concert volumes without having multiple loudspeakers project the same signal because of the resulting sound coloration. Additionally, we have a heavy schedule of concerts and guest artists and can always use a few extra concert loudspeakers. Thus, one of the main requirements was that the loudspeakers be powerful and of a good enough quality to be usable outside of the spatialization system. We settled on Meyer Sound UPJ-1 as a high-quality speaker. We have 33 UPJ-1, augmented with four CQ-1 and two CQ-2.

Using these speakers ruled out Wave Field Synthesis, since WFS functions best with small loudspeakers.

3.2 Speaker Positioning

A requirement for the Klangdom infrastructure was that it permits the loudspeakers to be quickly moved. There are both theoretical and practical reasons for this requirement.

First of all, given the range of music we present, we need to be able to accommodate a variety of loudspeaker setups. Though the default setup for the Klangdom is a "balanced" arrangement of speakers (more on this below), a composer may intentionally, for artistic reasons, desire an alternate speaker setup.

Additionally, works we present in the Kubus may have a visual component. We need to be able to rearrange the speakers to ensure that the visual aspects are properly represented — e.g., allowing the projector unfettered access to the projection screen. And since one concert program may involve works requiring different speaker setups, the Klangdom needed to be reconfigurable quickly, within the time allotted by a brief intermission.

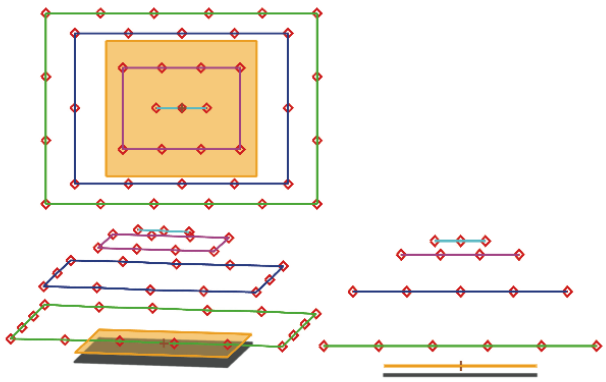


Figure 3. Distance-oriented distribution of speakers

Though quick reconfiguration and support for exotic setups was a consideration, we wanted the Klangdom to also be able to accommodate the ideal loudspeaker setup. What is the ideal loudspeaker setup? The answer is dependent on the technology employed for sound spatialization as well as compositional aims, but one reasonable answer is to distribute loudspeakers as "equally" as possible over the surface of a sphere. This is known in mathematics as a spherical code[14] and is, for example, the approach taken by Ambisonics[13].

However, distributing the speakers on the surface of a sphere is at odds with the geometry of the Kubus. As the name suggests, the Kubus is a rectangular prism and to build a spherical structure inside it would result in a considerable amount of unusable space. One option would be to distribute the speakers over an ellipsoid, as done by Küpper in Linz, but constructing curved rails to hang the loudspeakers upon proved to be considerably more expensive than building straight ones. Thus we decided to work with the geometry given by the Kubus and distribute the speakers on tracks parallel to the walls of the Kubus. Within these requirements, we created two canonical configurations — one distributes the speakers evenly in space (the distance between two speakers), the other evenly with respect to the angle between two speakers.

Figure 4. A Klangdom trolley

We built three rings, along with one straight segment directly overhead, upon which to hang the loudspeakers. The two outermost rings are comprised of track into which we can insert a trolley. The speakers are attached to the trolley by two cables to prevent them from rotating unwantedly. The speakers on these two tracks can easily be shifted laterally. The innermost ring is a one-piece ring, which does not facilitate the easy lateral movement of the speakers, but can be raised or lowered via a winch.

4. CONTROL

In keeping with our motto of flexibility, we have tried to stay as open as possible with regards to control for the dome. Although Wave Field Synthesis cannot work well with the given hardware, we can support other standard panning algorithms, such as Ambisonics and Vector Base Amplitude Panning (VBAP)[12].

Since the loudspeakers have built-in amplifiers, they are attached directly to a Yamaha DM2000 mixer and any computer with an audio interface capable of feeding 39 channels can connect to and drive the system. Thus, if a composer/performer comes to us with her own preferred spatialization software, she can easily plug that in.

However, for the cases where the composer does not have spatialization software prepared, we have developed our own software solution, Zirkonium.

4.1 Zirkonium

Zirkonium is software for sound spatialization based as heavily as possible on components provided by Apple as part of CoreAudio[1]. This ties us to Apple hardware but allowed us to develop the software more quickly, incorporating more functionality, than if were to have taken a cross-platform approach.

To use Zirkonium, the user defines a set of resources they want to pan. A resource is compromised of one or more channels of audio and may come from a sound file (AIFF, WAV, MP3, AAC, among other formats), from an audio interface, an AUNetSend input (an AudioUnit plug-in[1] that receives audio over a network) or a Jack[10] source. Having AUNetSend and Jack sources make it possible for other programs such as a digital audio workstation (DAW), Max/MSP, or SuperCollider to send audio into Zirkonium.

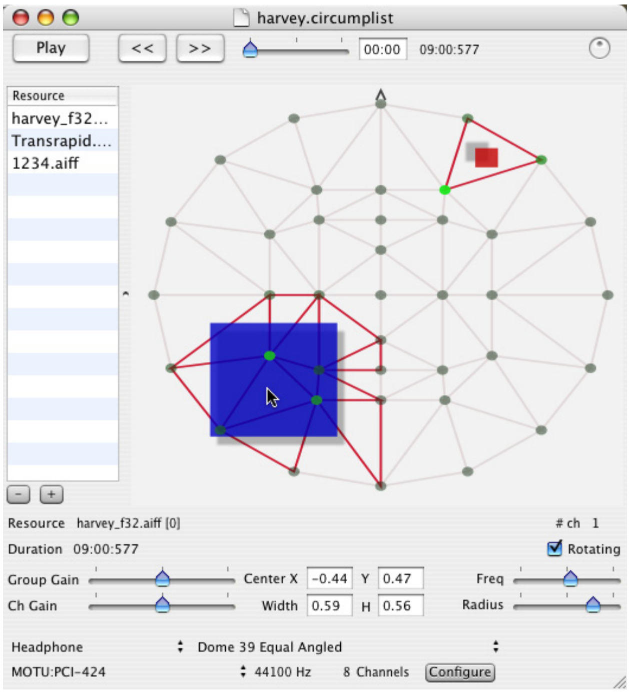


Figure 5. The Zirkonium interface

Once in Zirkonium, each channel of a resource is made available as a sound source which can be panned around the dome. We employ a panning technique that we have developed and call Sound Surface Panning (SPP). SPP is not actually a panning algorithm itself, but a technique for expanding the capabilities of an underlying panning algorithm such as Ambisonics or VBAP. Ambisonics and VBAP are oriented to controlling the position of point sources. SPP extends the underlying panning algorithm to make "sound size" a variable of control. Though a better solution needs to incorporate decorrelation [8][11], SPP takes a naive approach and decomposes a "rectangle source" into a mesh of points regularly distributed inside the rectangle, applies the underlying panning algorithm to these points, and then scales the output so the resulting acoustic power remains 1.

Using SPP, each sound source has 4 control parameters: X and Y position and width and height. These may be controlled in the GUI with the mouse, by a HID controller such as a joystick, or from another program via OSC[15]. Zirkonium can also read a panning score file and schedule the events described therein. We have also created a plug-in to control the pan parameters from within a DAW, better integrating spatialization into the composition process.

As hinted above, AudioUnits can also be hosted in Zirkonium. The Apple 3D in particular is utilized to facilitate some of the software's functionality. Zirkonium can produce a mix designed for listening over headphones or a 5.1 setup. This is done using the Apple 3D Mixer and specifying one of HTRF, Kemar Head, or 5.1 as the spatialization target and setting the positions of the loudspeakers as the locations of virtual sound sources. Such a mix may not be optimal for all purposes, but it may nonetheless be useful.

5. FUTURE WORK

Our work with the Klangdom is still in the early stages. Given our flexibility in loudspeaker positioning, we plan to investigate the utility of various loudspeaker configurations. One aspect of this work is to characterize the distortion incurred because of the non-ideal position of the speakers and look for ways to improve this. We also intend to investigate loudspeaker setups that have a psycho-acoustic foundation, rather than a pure mathematical one. Continuing the development of the SPP technique is something else that we see as important. For example, we are looking at ways to incorporate decorrelation. Designing and utilizing controllers is another area of research we are undertaking. As mentioned above, HID devices can be used to control the Klangdom and we would like to take advantage of force-feedback and haptics as a means of making the use of the Klangdom instrument more natural. To this end, we have also begun the design of tangible interfaces for controlling the Klangdom and expect to see interesting results. In the near future, we will be commissioning composers to realize compositions for the Klangdom. We expect the experience of working with composers using the system will bring about new insights, challenges, and changes.

6. ADDITIONAL AUTHORS

Bernhard Sturm, ZKM | Institut für Musik und Akustik; Lorenzstr. 19, D-76135; +49 (0)721 8100 1631, bs@zkm.de

7. ACKNOWLEDGMENTS

The Klangdom project was supported by the other members of the Institut für Musik und Akustik: Götz Dipper, Achim Heidenreich, Caroline Mössner, Marc Riedel, and Susanne Wurmnest. The project is generously funded by the Zunkunftsoffensive initiative of Baden-Württemberg.

8. REFERENCES

[1] Apple Computer. Core Audio Technologies. http://developer.apple.com/audio/coreaudio.html

[2] Chowning, John M. The Simulation of Moving Sound Sources. *J. Audio Eng. Soc. 19,* 1971, 2-6.

[3] Bayle, François. À propos de l'Acousmonium, Recherche Musicale au GRM. *La Revue Musicale 394-397,* 1986, 144-146.

[4] Berkhout, A.J. A Holographic Approach to Acoustic Control. *Journal of the Audio Engineering Society, 36(12),* 1988, 977-995.

[5] Boone, Marinus M. Acoustic rendering with wave field synthesis. Position paper for *ACM SIGGRAPH and EUROGRAPHICS Campfire* (Snowbird, Utah, May 26-29, 2001). Available online at: <http://www.siggraph.org/conferences/campfires/acoustic2001/abstracts/boone.pdf>

[6] Daniel, Jérôme. *Représentation de champs acoustiques, applications à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimedia.* Phd Thesis, Université Paris VI, 2001.

[7] Jot, Jean-Marc and Warusfel, Olivier. Spat~: A Spatial Processor for Musicians and Sound Engineers. In *CIARM: International Conference on Acoustics and Musical Research,* (Ferrara, Italy. 1995). Available online at: <http://catalogue.ircam.fr/articles/textes/Jot95a/>

[8] Kendall, G. S. The decorellation of audio signals and its impact on spatial imagery. *Computer Music J.* vol. 19, no. 4 (1995), 71-87.

[9] Küpper, Leo. Die Klangkuppel bei der Ars Electronica 1984 in Linz. In *Ars Electronica Festival-Katalog 1984 (Band 01).* 1984. Available online at < http://www.aec.at/de/archiv_files/19841/1984_016.pdf>

[10] Letz, S., Fober, D., Orlarey, Y., and Davis, P. Jack audio server: MacOS X port and multi-processor version. In *Sound and Music Computing '04* (Paris, France, October 20-22, 2004).

[11] Potard, Guillaume and Burnett, Ian. Decorrelation techniques for the rendering of apparent sound source width in 3D audio displays. In *Proc. of the 7th Int. Conf. on Digital Audio Effects* (DAFx'04) (Naples, Italy, October 5-8, 2004).

[12] Pulkki, V. Virtual sound source positioning using vector base amplitude panning. *J. Audio Eng. Soc.* 45(6) (June 1997), 456-466.

[13] Ward, Darren B. and Abhayapala, Thushara D. Reproduction of a plane-wave sound field using an array of loudspeakers. *IEEE Transactions on Speech and Audio Processing,* vol 9., no. 6 (September 2001).

[14] Weisstein, Eric W. Spherical Code. From *MathWorld -- A Wolfram Web Resource.* http://mathworld.wolfram.com/SphericalCode.html

[15] Wright, Matthew, Freed, Adrian, and Momeni, Ali. Open sound control: state of the art 2003. In *Proc. of the 2003 Conf. on New Interfaces for Musical Expression* (NIME-03) (Montreal, Canada, 2003).

[16] Zvonar, Richard. A History of Spatial Music. *eContact! 7.4.* 2005. Université Concordia. 27 Oct. 2005. <http://cec.concordia.ca/econtact/Multichannel/spatial_music.html>

A framework for immersive spatial audio performance

Mike Wozniewski
McGill University
Centre for Intelligent Machines
3480 University Street
Montreal, Quebec
mikewoz@cim.mcgill.ca

Zack Settel
La Société Des Arts Technologiques
1195 Saint-Laurent boulevard
Montreal, Quebec
zack@sat.qc.ca

Jeremy R. Cooperstock
McGill University
Centre for Intelligent Machines
3480 University Street
Montreal, Quebec
jer@cim.mcgill.ca

ABSTRACT

Traditional uses of virtual audio environments tend to focus on perceptually accurate acoustic representations. Though spatialization of sound sources is important, it is necessary to leverage control of the sonic representation when considering musical applications. The proposed framework allows for the creation of perceptually immersive scenes that function as musical instruments. Loudspeakers and microphones are modeled within the scene along with the listener/performer, creating a navigable 3D sonic space where sound sources and sinks process audio according to user-defined spatial mappings.

Keywords

Control paradigms, 3D audio, spatialization, immersive audio environments, auditory display, acoustic modeling, spatial interfaces, virtual instrument design

1. INTRODUCTION

Virtual environments (VEs) have received considerable interest from researchers in fields as diverse as scientific visualization, cognitive perception, and medical therapy. Despite continued improvements in the quality of VE audio components, there has been comparatively little use of this technology in the areas of musical performance and interactive sonic display. This is perhaps due to the simplistic and impoverished audio representations, geared largely toward gaming applications on consumer-grade systems, that dominate the field. In these simplistic models, virtual sound sources are associated with fixed models of propagation, with audio rendered identically for every source. The demands of musical composition and performance, however, require far greater flexibility of control over sound propagation and mapping strategies. Musicians are typically less interested in having perceptually accurate acoustic models, and would rather have the power to manipulate or exaggerate those affects for artistic purposes. The research framework described here is among only a few (eg. [11, 12, 14]) that have explored VEs from the perspective of musical performance. It allows for the control of various features on the basis of individual sound sources and sinks, thereby facilitating the design of rich 3D audiovisual scenes that function as virtual musical instruments.

The framework allows a user to create a navigable 3D scene where his position and orientation are modeled in space, as well as the positions of the loudspeakers with which he will be rendering the scene. This allows the user to control the progression of a musical piece by traveling through virtual space, and to switch easily between different auditory display technologies. The content of the scene is composed of spatially positioned sound sources and sinks that process audio according to user-definable DSP routines. Any form of input device can be used and easily mapped to affect DSP or visualization parameters. Moreover, sonigenic mappings can be employed to visualize the sonic properties of various scene elements. The user's ability to control how and where audio propagates within the scene suggests a powerful control paradigm, building on the human strength of spatial understanding.

2. AUDIO IN VIRTUAL ENVIRONMENTS

By incorporating diegetic[1] audio into a VE, we provide users with additional information for localizing objects and events in the scene. This allows them to perform more parallel and simultaneous tasks, since the cognitive load imposed on the user is lessened. Furthermore, the psychological immersion that the user perceives parallels that of the real world, hopefully leading to a more pleasant experience.

2.1 Auditory Display Technologies

Research in the area of virtual audio environments is mainly concerned with the challenge of perceptually localizing sounds in 3D space. This is known alternatively as 'audio spatialization', 'audio rendering', or 'sound imaging'. Many techniques exist (see [4, 10, 13, 18] for an overview), but all these methods can be grouped into three main classes: binaural methods, amplitude differencing, and wave field synthesis (WFS).

Binaural methods use filters to simulate frequency-based modifications to the audio signals based on a head response transfer function (HRTF). The filter coefficients for sounds originating at various coordinates are acquired by empirically measuring the audio responses on a model human head [7]. This way, one can spatialize a sound by filtering two audio signals and rendering them through headphones, or through a stereo speaker setup. Headphones are obviously the best display method, since the audio signals are isolated to each ear and crosstalk cannot not occur. There are many methods for crosstalk cancellation [6], but the user is required to stay motionless, otherwise the perceptual effect is ruined.

The consumer-grade systems that many gamers and cinemaphiles have deployed in their homes use several speakers placed equidis-

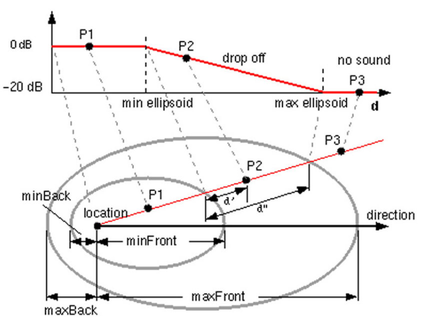
[1]The term *diegetic sound* is common in the film industry and describes a sound that is presumed to be present and localized in the scene. If a character in a movie plays an instrument, this is diegetic music, whereas the omni-present background music (that the character is unaware of) is non-diegetic.

tant from the one central point, often called the *sweet spot*. These systems typically use amplitude differences between loudspeakers to make sounds appear from a certain direction. The commercial-grade systems are however usually pantophonic (localize sounds only in a horizontal plane), and require very precise speaker arrangements. Some work has made periphonic (fully 3D) methods possible. Ambisonics [8] for example, encodes audio signals with directional components (x,y,z,w), and decoders exist that reproduce the necessary sound signals on multiple speakers to simulate sounds arriving at the user with apparent direction. Vector base amplitude panning (VBAP) [15], groups speakers into triplets, and a direction vector can be computed for any virtual sound source located within the loudspeaker triangle. Both ambisonics and VBAP support rendering of 3D sound with variable numbers of speakers, though ideally the speakers should be uniformly distributed and equidistant from the user to minimize error.

Contrary to amplitude panning techniques, where the sound field is accurately produced only in the sweet spot, WFS attempts to recreate the sound field throughout an entire volumetric space. This requires an array of many speakers, and much computational power, in order to compute the propagation of each boundary signal. The sound field may not be completely accurate in any one spot, but the error field is minimized and uniformly distributed, allowing the listener to move about the space without loss of perceptual auditory immersion.

2.2 3D Audio Scene Description

In the case of real-time interactive systems such as games and virtual reality systems, there have been many attempts at modeling virtual sound sources. Several toolkits available for developing such applications, including Microsoft DirectX, OpenAL, and X3D. Creative EAX can be added to these to simulate room reverberation effects complete with occlusions, exclusions, and obstructions. These toolkits allow developers to move sounds sources interactively around a virtual scene and have them properly rendered. However, support only exists for standard surround formats, so arbitrary speaker configurations for ambisonic or VBAP rendering are not yet possible.



1: The X3D sound node model [1].

However, the most limiting factor of these toolkits is the fact that they use very simplistic spatial models of audio sources. Most APIs have no method to specify directivity of sounds (i.e. all sounds are considered omni-directional), usually a maximum of one listener is supported, and very little power is provided to manage more complex sonic interactions between sound nodes. In fact, the only API that supports directivity of sound sources is X3D, which models directivity with two ellipsoids (as shown in Figure 1). The inner ellipsoid indicates the commencement of attenuation based on distance, which proceeds linearly until full attenuation occurs at the outer ellipsoid.

The problems with this method are abundant. First, there is no method to represent more complicated directivity patterns, such as that of traditional musical instruments. Similarly, directivity applies only to sound sources. Sound sinks (i.e. listeners) cannot be modeled with directional sensitivity, thus cardioid patterns of microphones cannot be modeled. Finally, the linear attenuation model does not accurately relate to real-world physics, where sound in fact decays exponentially.

There have been a few projects that have gone to much further extents in modeling sound source propagation for interactive systems. For example, the DIVA (Digital Interactive Virtual Acoustics) project [17] uses ray casting (similar to methods in 3D graphics) to compute reflection paths of audio signals. Additional virtual sources are created at the reflection points, and by modeling surfaces with absorption parameters, they achieve a much more realistic response of the environment. Tsingos et al [19] augment this technique, by also modeling the diffraction of sound around the edges of 3D objects using the uniform theory of diffraction (UTD). These methods still however simplify the directionality of sound sources in order to make their more complicated rendering approaches tractable in real-time.

3. A SPATIAL AUDIO FRAMEWORK FOR MUSIC & SOUND APPLICATIONS

Although the traditional methods presented earlier do offer control mechanisms for interactivity, these often pertain to spatial rather than sonic arrangement. In traditional methods, sounds can be moved individually in real-time, but their acoustic parameters (directivity pattern, roll-off, etc.) cannot be manipulated dynamically. Rather, these parameters remain fixed and are usually defined for the entire scene rather than an individual node. For example, it is not possible to include some sources in the computation of reverberation, while excluding others. Furthermore, the virtual sound sources tend to map to external audio streams (sound files, networked audio, etc.) rather than signal processing objects that can interact with other nodes according to their spatial arrangement.

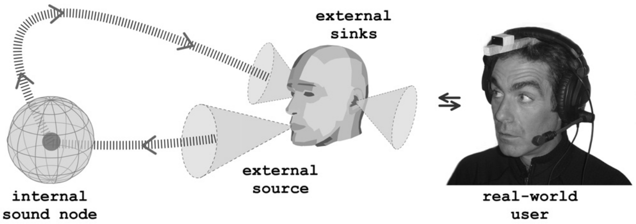
For the purpose of musical creation and performance, perceptually accurate models of sound propagation may not always be desired. The user (or rather, musician) may desire to exaggerate certain acoustic effects for artistic purposes, and use the 3D propagation of audio as a medium for musical expression. The audio scene representation that we propose allows for this level of control by making *sound nodes* much more generic entities, and explicitly requiring a *sound connection* to be made between nodes. With proper management of these nodes and connections, one can create a navigable 3D space, that functions as a control interface for an immersive virtual instrument.

3.1 The Sound Node

Sound nodes in our representation can be either *sources* (emitting sound), *sinks* (absorbing sound), or both. The latter case is particularly important in the context of musical performance since we can allow a node to absorb sound, apply DSP, and then emit the result back into the scene. By doing so, we create *internal* nodes, which become virtual effects units that process audio at some location in space. Each node is also represented with a parametric directivity pattern, giving the user highly accurate control over objects with which the node can interact.

Notice also that the concept of a listener within the scene is somewhat more abstract; While traditional systems allow for only one listener, our representation supports many. A listener is just a particular type of sink node (or group of sinks), where the audio absorbed by the sink is written to hardware buffers (eg. for headphones and speakers). We call this an *external* sink, since it relates to external or real-world entities. External sources also exist; they represent input from microphones, sound files, or other audio streams.

Figure 2 illustrates a simple example of how internal and external sound nodes function together. The audio output of the scene is delivered to the user through two external sink nodes, which correspond to the real world headphones being worn. These are positioned in space relative to his virtual head position. When a user sings, their voice is propagated directionally into the scene and interacts with any internal nodes within range. Those nodes process the audio and emit the result back into the scene. Methods for tracking the user's head orientation and position relative to that of the virtual world are discussed in Sections 5.2 and 5.3.



2: Simple example: A singer's voice propagates directionally into the scene, gets processed, then propagates back toward the singer.

In addition to the advantage of supporting multiple simultaneous listeners, it is also possible to easily change the type of listening hardware by creating the appropriate nodes. This would be done for example, when the user switches from headphones to a multi-channel speaker configuration.

3.2 The Sound Connection

As mentioned earlier, a musical performance context benefits from the addition of various parameters to more finely control the audio propagation between nodes. Rather than having an identical propagation model for every node, we construct *sound connections* between every source-sink pair in the scene. This allows users the ability to enable and disable various connection features.

This is important for several reasons. The fact that a node has the ability to process audio and re-emit the result means that there is no guarantee of diminishing sound energy as audio makes its way through the scene. To prevent the occurrence of intensifying feedback loops, the the user must ensure that only acyclic connection chains exist. Conversely, feedback may be desirable from an artistic perspective, so the user may wish to create such loops among only certain nodes.

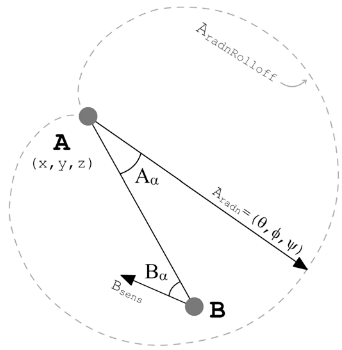
DISTANCE DECAY	The attenuation of the sound signal with distance
ANGULAR ROLL-OFF	The amount of decay relative to the angle of propagation
DOPPLER EFFECT	A variable delay as a function of distance
PROXIMITY EFFECT	A low-shelf filter for small distances (close proximity)
ABSORPTION FILTER	A low-pass filter simulating air absorption as a function of distance
INCIDENCE FILTER	A low-pass filter to simulate frequency-dependent attenuation with angle
REVERB	A filter with an impulse response representative of the current scene size

1: Features of a sound connection.

The *connection features* are described in Table 1. The user can specify the intensity of each feature in a given connection. This allows for some interesting special cases; for example, the user could "teleport" a sound signal between two very distant nodes by removing the effects of distance decay and Doppler. Additionally, diminishing the Doppler effect can be useful to prevent detuning of pitched material in musical passages.

3.3 Computation of audio propagation

Once a sound connection is established between two nodes and the various connection features have been specified, the computation of audio propagation is quite simple. Though the virtual sound nodes we saw in Figure 2 provided cone-shaped patterns of propagation, this is a simplification of more the elaborate mechanism shown in Figure 3.



3: A virtual sound node in more detail.

A source node $\mathbf{A}$, with a more complicated cardioid[2] directivity pattern has several properties, including a 3D position (x, y, z) and the orientation with which it *radiates* sound: $\mathbf{A}_{\text{radn}} = (\theta, \phi, \psi)$. It also has an angular roll-off, $A_{\mathbf{radnRolloff}}$, that describes how the radiation of sound is affected at various angles of propagation. The sink node $\mathbf{B}$, has similar properties though it's directivity is called $\mathbf{B}_{\text{sens}}$ and it's rolloff (not shown) is called $B_{\mathbf{sensRolloff}}$ to imply *sensitivity* rather than radiation.

3.3.1 Distance Decay

Sound decays exponentially according to the inverse square law [16]. We compute the distance gain, G_{dist}, that the signal should be multiplied by to simulate this decay:

$$G_{\text{dist}} = \frac{1}{(1 + |\mathbf{B} - \mathbf{A}|)^{\beta}} \tag{1}$$

The additional control parameter, β, is used to control the steepness of the exponential decay. When $\beta = 2$, this model is identical to natural sound decay.

3.3.2 Angular Roll-off

The angle of incidence between a node's orientation and the vector connecting the source and sink can be computed with basic vector mathematics. For example, A_α is found by:

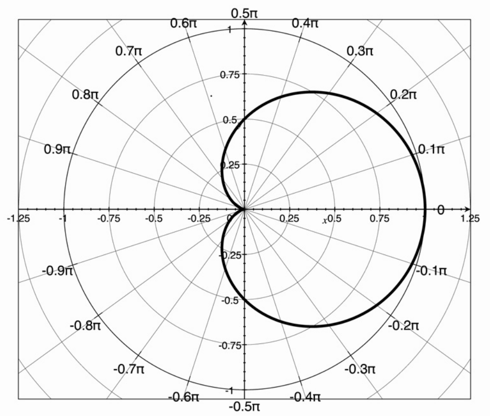
$$A_\alpha = \cos^{-1}\left(\frac{\mathbf{A}_{\text{radn}} \bullet (\mathbf{B} - \mathbf{A})}{|\mathbf{A}_{\text{radn}}|\,|\mathbf{B} - \mathbf{A}|}\right) \tag{2}$$

B_α is computed in the same fashion. Then we compute a gain value that simulates the sensitivity of the sink and radiation of the source with:

$$G_{\text{rolloff}} = A_{\mathbf{radnRolloff}}(A_\alpha)\, B_{\mathbf{sensRolloff}}(B_\alpha) \tag{3}$$

[2]It's worthwhile to note that such directivity patterns are commonly found in the field of acoustics. Most directional microphones exhibit sound sensitivity similar to cardioids, and many traditional musical instruments have lobed radiation patterns [16].

While angular roll-off can be described by a mathematical equation, we choose to represent this with a lookup table for greater flexibility to experiment with custom roll-off functions. The cardioid pattern from Figure 3 is more nicely depicted in Figure 4. For each angle of incidence (i.e., A_α), the gain value is shown that indicates the level of attenuation based on incidence. At an angle of zero, the entry in the table returns 1.0. As the incidence increases to 90°, the gain drops to 0.5 until it finally reaches 0.0 at an angle 180°.



4: A rolloff function for cardioid sensitivity/radiation.

3.3.3 *Doppler*

Doppler is the apparent frequency shifting of a sound as a function of changing distance. This is accomplished using a variable delay embedded in each connection. The delay time corresponds to the amount of time it would take a sound to travel the distance between the connection's source and sink.

3.3.4 *Other spatial filters*

We attempt to model frequency-specific audio attenuation effects based on distance and angle. For example, high frequencies are quite directional and usually much quieter behind the source while low frequencies tend to wrap around and are heard equally in all directions. We simulate this with the use of a low-pass filter (which we call an *incidence filter*), whose cutoff frequency increases with the angle of incidence. High frequencies are also lost due to air absorption as the distance between source and sink increases. This too is modeled with a low-pass filter, called an *absorption filter*.

A common effect known to sound recording engineers, the *proximity effect*, models the boost of low frequency energy when the distance between source and sink is very small. This is accomplished with the use of a low-shelf filter, but only when when a threshold closeness is attained. Though this is a side effect of directional microphones in the real world, it serves as a good cue for spatial proximity. Furthermore, it is a familiar effect that most people will recognize, particularly performing musicians.

Another spatial effect that we consider is a simulation of reverberation by filtering with an impulse response function. While this impulse function should ideally be computed directly from the scene geometry, this has not yet been implemented in an automatic fashion. For now, the user must specify which type of room is closest to the desired effect, similar to Creative EAX filter choices such as hallway, arena, or bathroom. In future work, we plan to examine ray casting tecnhiques similar to the DIVA project [17] mentioned earlier, where additional sound sources would be automatically created at important reflection points. The balance of direct sound, reflections, and diffuse/reverberant sound will greatly add to the user's spatial awareness. The fact that our framework allows the user to tune each of these components is quite interesting. For example, a user could diminish directivity effects (angular roll-off and incidence filtering) when the distance between source and sink is larger than the reverberation radius. The reverberant sound would thus dominate and the sense of directivity would decrease, which is a common psychoacoustic effect. On the other hand, reverb may be heavily exaggerated or made to contradict directivity for an interesting artistic effect.

4. THE ROLE OF GRAPHICS

A further important distinction of our approach from other projects is the rich visual component. In our framework, graphics are not only used as an artistic medium, but are essential components of the editing and authoring environment. When dealing with large amounts of data and many control parameters, it is important to have suitable feedback. For example, the designer of a complex scene can visually perceive the spatial arrangement of sound nodes, when doing so by ear might be difficult or impossible. In performance, graphical feedback provides a high degree of precision when directing a sound towards a particular sink, especially when either are in motion.

4.1 Graphical Engine

Since users of the system are creating artistic content, we would like to provide a graphical engine that supports beautiful and convincingly believable content. This means that our engine should allow designers to develop 3D graphics with the most cutting edge tools on the market. Ideally, we would like to allow graphics development with any commercial application that designers might be comfortable with, and then provide a simple mechanism to incorporate these graphics into the virtual world.

The scene that is displayed will typically be composed of complicated, independently designed 3D models such as: people, musical instruments, audio equipment, and architectural elements such as furniture, walls, and floors. The spatial interactions between these elements will likely be high-level involving simple movements like translation, rotation, or scaling. We anticipate that the need to control these models at the vertex level will be rare, although the ability to animate various components of these models will be required - for example, articulating the body parts of a human avatar. With these constraints in mind, we have adopted a scene-graph representation for the virtual world.

A scene-graph is simply a data structure that organizes the logical and spatial elements of a 3D scene. Each element is represented as a node in a tree structure, such that any spatial transformations performed on a parent node will be propagated automatically to all children nodes. Scene-graphs are particularly useful for representing rigid structures and managing hierarchical scenes where groups of objects are animated together. By applying operations to the parent node, we can affect all other grouped elements without specifically indicating them. This holds not only for spatial commands, but also for logical commands such as hiding a group, pruning it from the scene, and propagating other higher-level parameters.

A custom 3D engine has been developed using the OpenSceneGraph (OSG) [2] graphics toolkit, that allows interactive control of various 3D models, with several audio-centric graphical elements made available to the user. The user is also provided with tools to distribute the rendering over several screens, so that a truly immersive environment can be created.

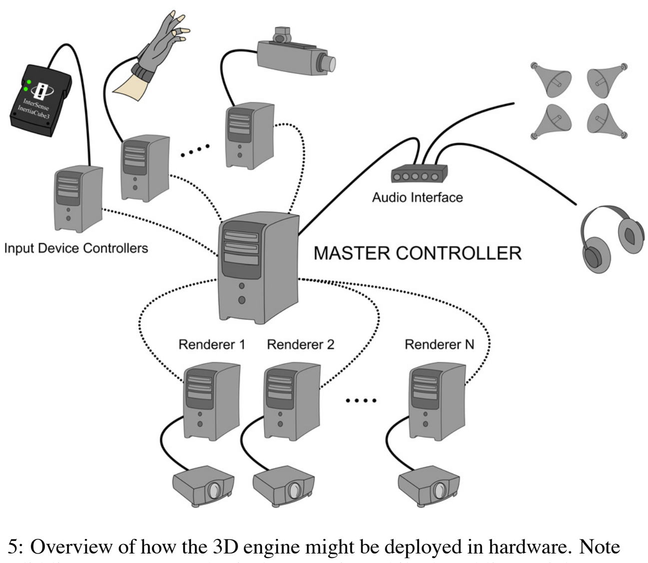
4.2 Sonigenic Graphical Display

When dealing with audio propagation on such a controllable level, it may become difficult to conceptualize what processes are taking place. The possibility of meaningful visualization is thus highly appealing. We provide users the ability to visualize the directivity of their sound nodes with a 'laser beam', wireframe mesh, or lighting affects. Each approach has its respective benefits and drawbacks. The laser beam only shows the direction vector of a sound node, while the wireframe mesh shows the radiation or sensitivity pattern as well. Lighting, though computationally expensive, can act as an appealing analog to the behavior of sound as it propagates. Light has similar decay properties (in terms of both distance and angle), but can only be used to visualize monotonically decreasing roll-offs.

For each sound node, an animated intensity meter is also available which displays the current sound energy contained within the node, similar to a home stereo VU meter. Other custom animations are possible using keyframe indexing of 3D models. The animations can be created using any 3D modeling application such as Maya or 3D Studio Max, and then assigned a control signal that determines the time index. This provides a powerful mechanism for sonigenic display, where audio parameters such as energy, envelope, or even beat detection can be rendered visually. Further details of this process can be found in Section 5.3.

5. CONTROL ENVIRONMENT

This system is developed with artists and designers in mind, and thus we aim for simple control mechanisms to allow such users to modify and extend the functionality. This is primarily accomplished with the inclusion of the PureData (Pd) [3] patcher programming language. Pd has a large community of music-focused programmers, who are continually developing new DSP and control methods with this language. Hence, we believe that it is the ideal candidate to avail powerful control in an extensible fashion.



5: Overview of how the 3D engine might be deployed in hardware. Note solid lines represent a physical connection while dotted lines might represent a network connection via TCP/UDP.

Figure 5 illustrates a sample hardware architecture, including various sensors and rendering technologies. This is similar to a CAVE-like [5] immersive environment.

5.1 Digital Signal Processing (DSP)

Any type of complex DSP can occur within a sound node. In fact, every node has a Pd abstraction with an incoming and outgoing signal, where users can insert their own DSP. For example, a user can filter the signal, add echo or delay, or process the sound according to some physical model. A node's spatial parameters (eg. position or orientation) can be used in the DSP computation, increasing the audio-visual coherence.

Some special DSP cases are worth pointing out. For example, if the user chooses not to send a signal back out, and rather writes the incoming signal to soundcard buffers, then they have created the external sink that was mentioned in Section 3.1. This is how real-world loudspeakers are modeled. On the other hand, if the user ignores the incoming signal and instead generates only an outgoing signal (perhaps by playing a sound file, or reading from the soundcard's input buffer), then they have created an external source. Input from the real world is modeled this way.

We should note that sound nodes are mono-channel audio objects. Multi-channel audio effects may be created with the 'grouping' mechanism, where sound nodes are arranged together in a scene-graph. As a result, any operation applied to the parent will propagate to the entire group. This applies not only to typical spatial transformations (such as translation or rotation), but also to sonic operations (such as establishing connections or distributing audio input). If we consider again the example in Figure 2, we note that the user's headphones and microphone should be grouped together, as children of a 3D head model. By translating the head, the source and sinks will automatically translate with it.

5.2 Input Daemons

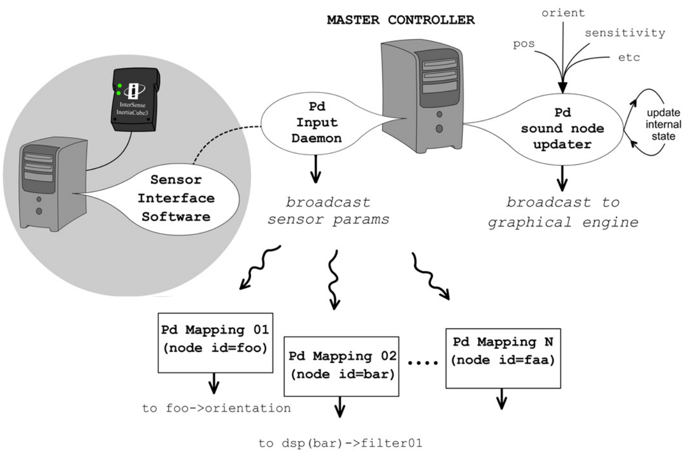
Creating virtual musical instruments is of little value unless they can also be played. We believe that an excellent input mechanism is natural body motion. The human body is directly modeled in 3D and various control mappings can be implied from body posture or motion (gestures). Such control makes direct use of what we have already learned through daily practice with our bodies. In particular, kinesthetic feedback informs us of our body's orientation in space. This provides additional information in another modality channel, which again decreases the cognitive load imposed on the user, and allows for natural and expressive control [20].

Currently, body tracking solutions are being investigated using computer-vision based approaches, although we note that performance of these methods remains disappointing due to a lack of both temporal and spatial precision. It has been demonstrated several times that musical systems are particularly demanding of low-latency control [21]. Thus we must include the possibility of using other input devices with lower latency and greater fidelity. Examples include data gloves, orientation sensors, accelerometers, and other such devices. Each device is interpreted by a 'Pd input daemon', which formats input signals into a standard message format (in the range of [0,1] for each parameter). These messages are available for control of DSP or more complicated mappings, as described in the following sections.

5.3 Mapping Considerations

One of the trickiest aspects of creating powerful musical interactions is the mapping of control parameters to musical effect. Many of today's so-called computer music interfaces either end up being too simple to achieve virtuosic playability in the long term, or too complicated to learn in the short term. Wessel & Wright [21] have made this point clear, and suggest that controllers have mappings to higher-level musical behaviors such as navigation through timbre space, or any multidimensional synthesis control. Hunt & Kirk [9] likewise suggest that 'holistic control' of a parameter space is better than sequential control using one-to-one mappings. They also mention the need for a 'performance mode', where the system "stays constant whilst the focus is on the improvement of the player". Our system takes inspiration from these ideas by offering a spatially immersive control paradigm.

Since users are both geometrically and perceptually encompassed within the instrument, they have the ability to explore complicated parameter configurations in a holistic manner. Multimodal feedback is provided so that users can easily learn how the system reacts to certain actions, and there are typically many methods to achieve similar results. We offer a 'performance mode', though the distinction between this and the 'editing mode' is fuzzy. An initial configuration of nodes can be created during an authoring phase, but this configuration can change in realtime during performance. For example, a user may choose to 'grab' a node and move it to a new location, or even animate the node in an orbiting fashion. Thus mappings are dynamic and manipulable, providing users with a dynamic mapping interface rather than a static performance system.



6: How mappings can be defined.

Figure 6 shows how Pd can help to realize more complex mappings. The node updater on the right receives parameters relating to a specific node id. The input daemon formats sensor readings into a standard format and broadcasts that information to any Pd mapping abstraction that a user might create. For example, one mapping might receive values from an orientation sensor and control the 3D orientation of a node while a different mapping uses those values to compute coefficients for a DSP filter.

6. CONCLUSION

We have presented an initial research framework for creating 3D scenes that function as spatial musical instruments. In addition to a more thorough representation of the 3D audio scene, we have introduced the concept of spatial navigation through a musical 'piece'. Powerful control mechanisms are provided that allow users to map parameters (either from input devices or generated by sonic events) to meaningful audio-visual outcomes. We believe that natural spatial mappings that humans typically encounter in the real world will make this paradigm particularly effective for musical expression.

7. ACKNOWLEDGEMENTS

The authors wish to acknowledge the generous support of NSERC and Canada Council for the Arts, which have funded the research and artistic development described in this paper through their New Media Initiative. In addition, the authors are most grateful for the resources provided to the project by La Société Des Arts Technologiques and to the Banff Centre for the Arts for offering a productive and stimulating environment in which several of the applications described here were prototyped.

8. REFERENCES

[1] Extensible 3d (x3d) iso standard (iso/iec 19775:2004). http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/.

[2] Openscenegraph. www.openscenegraph.org.

[3] Puredata. www.puredata.info.

[4] D. R. Begault. *3-D sound for virtual reality and multimedia*. Academic Press Professional Inc., 1994.

[5] C. Cruz-Neira, D.J.Sandin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of SIGGRAPH '93*, pages 135–142. ACM Press, 1993.

[6] W. G. Gardner. *3-D Audio Using Loudspeakers*. Kluwer Academic Publishers, Norwell, MA, 1998.

[7] W. G. Gardner and K. D. Martin. Hrtf measurements of a kemar. *J. Acoustics Soc. of America*, 97(6):3907–3908, 1995.

[8] M. Gerzon. Periphony: With-height sound reproduction. *J. Audio Eng. Soc.*, 21(1):2–10, 1973.

[9] A. Hunt and R. Kirk. Mapping strategies for musical performance. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. Ircam - Centre Pompidou, 2000.

[10] J. M. Jot. Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia Systems*, 7(1):55–69, 1999.

[11] T. Maki-Patola, J. Laitinen, A. Kanerva, and T. Takala. Experiments with virtual reality instruments. In *Proceedings of NIME*, pages 11–16, 2005.

[12] A. Mulder, S. S. Fels, and K. Mase. Mapping virtual object manipulation to sound variation. In *Proceedings of the IPSJ-SIGMUS*, pages 63–68, Tokyo, Japan, Dec 1997.

[13] M. Naef, O. Staadt, and M. Gross. Spatialized audio rendering for immersive virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 65–72. ACM Press, 2002.

[14] J. Pressing. Some perspectives on performed sound and music in virtual environments. *Presence*, 6(4):482–503, 1997.

[15] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. *J. Audio Eng. Soc.*, 45(6):456–466, 1997.

[16] T. D. Rossing. *The Science of Sound (2nd edition)*. Addison-Wesley, 1990.

[17] L. Savioja, J. Huopaniemi, T. Lokki, and R. Vaananen. Creating interactive virtual acoustic environments. *J. Audio Eng. Soc.*, 47(9):675–705, 1999.

[18] G. Steinke. Surround sound - the next phase : An overview. *J. Audio Eng. Soc.*, 44(1):651.

[19] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of SIGGRAPH '01*, pages 545–552. ACM Press, 2001.

[20] R. Vertegaal, T. Ungvary, and M. Kieslinger. Towards a musician's cockpit: Transducers, feedback and musical function. In *Proceedings of ICMC*, 1996.

[21] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *Workshop at NIME*, 2001.

An Architectural Framework for Interactive Music Systems

Alexandre R.J. François and Elaine Chew
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA
{afrancoi,echew}@usc.edu

ABSTRACT

This paper introduces the Software Architecture for Immersipresence (SAI) framework to the computer music community. SAI is a software architecture model for designing, analyzing and implementing applications that perform distributed, asynchronous parallel processing of generic data streams. The most significant innovation of SAI is its ability to handle real-time DSP, interactive control, and data-centered representations in a unified model. This generality facilitates the design and implementation of complex interactive systems that combine music analysis, synthesis and on-line control. Two examples illustrate the use of SAI in the design and implementation of interactive music systems: MuSA.RT, a system for real-time analysis and interactive visualization of tonal patterns in music, and ESP, a driving interface (wheel, pedals and display) for creating expressive performances from expressionless music files.

Keywords

Software Architecture, Interactive Systems, Music software

1. INTRODUCTION

This paper introduces the Software Architecture for Immersipresence (SAI) framework [15] to the computer music community. SAI is a software architecture model for designing, analyzing and implementing applications that perform distributed, asynchronous parallel processing of generic data streams. Two examples illustrate the use of SAI in the design and implementation of interactive music systems.

The design principles underlying SAI are rooted in François' past and ongoing cross-disciplinary research on interactive systems at the Integrated Media Systems Center (IMSC), an NSF Engineering Research Center at the University of Southern California, and on computer vision systems, at USC's Institute for Robotics and Intelligent Systems. Although originally motivated by the Immersipresence vision–that is, combining immersion and interactivity [20]–, SAI provides a general formalism for the design, analysis and implementation of *complex software systems of asynchronous interacting processing components.*

SAI defines architectural level abstractions that are consistent with a general model of computation. These abstractions resolve a class of seemingly related fundamental issues, that are characerized by Puckette as a divide between processing and representation paradigms [26], and by Dannenberg as the difficulty of combining (functional) signal processing and (imperative) event processing [9].

Interactive systems are particularly interesting and challenging, as they require on-line (real-time) analysis and synthesis of data media of different nature. SAI is designed explicitly to address the limitations of traditional approaches in this context. For historical reasons, SAI was first applied to the design and implementation of real-time and interactive computer vision systems [17]. Its relevance in the context of interactive music systems was explored and established over the past few years through collaborations between the authors, started at IMSC.

The remainder of this paper is organized as follows. Section 2 relates the SAI approach to major landmarks in the rich computer music history. Section 3 describes the main features of the framework: the architectural abstractions that form the SAI style, important properties, and design and implementation tools. Section 4 demonstrates the use of the framework with two interactive music systems. Finally, Section 5 offers concluding remarks and outlines research and development directions for ongoing and future work.

2. RELATED WORK

Amatriain's thorough review and analysis of the audio and music processing software landscape in his thesis [1] testifies to the richness and diversity of this field.

Existing approaches can be characterized according to various criteria. For example, levels of abstraction range from programming languages to code libraries, to application frameworks, to programming environments (possibly visual), and graphical applications. Another criterion is the primary objective, which can be music or audio analysis, synthesis, or composition. Requirements may include real-time constraints or interactivity (i.e. low latency).

Focusing on interactive systems and their underlying models of computation separates *on-line* from *off-line* approaches. In the on-line group, synthesis-oriented efforts adopt models from Digital Signal Processing (DSP), and are often concerned with scheduling (because of the hard real-time requirement). This category comprises of the lineage rooted in the Music N languages [18], which includes Csound [28, 27, 13], the Max paradigm [25] in its various forms (e.g. Max/MSP [19] and Pure Data [24]), and others. All of these assume a process oriented dataflow model, which is not adapted to interactive manipulation. Researchers have introduced various mechanisms for interactive control in on-line DSP-style systems (either synthesis- or analysis- oriented). For example, in Aura [9] a message-passing model complements the dataflow model. This hybridation occurs at a rather low level of abstraction (language).

The off-line group comprises of analysis and composition oriented systems that focus on representation. The interactive nature of these systems, if any, comes from the existence of a visual graphical user interface (GUI) to manipulate the structures or their computations (note that GUIs are typically built on message passing models). OpenMusic [22, 3, 4] is a representative member of this category. Its visual interface adopts the popular patch metaphor used in Max/MSP and most visual environments for music processing, but it does so with very different semantics: as observed by Puckette in [26], Max patches contain dynamic process information, while OpenMusic patches contain static data. The dataflow model of OpenMusic maps to the functional programming approach, and has no model for representing or handling real-time (or on-line) events.

Both semantics are useful for different purposes, and part of their appeal is that they define high-level abstractions. Unfortunately, the two models are incompatible. Assayag and Dubnov's improvisation system [2] provides a concrete example. Off-line versions of the learning and generation algorithms are implemented in OpenMusic. The on-line implementation of the system, OMax [11], utilizes the OpenMusic implementation in conjunction with Max/MSP to handle the on-line aspects, such as real-time control, MIDI and audio acquisition and rendering. Communication between, and coordination of, the two subsystems requires the use of a special interaction protocol, OpenSound Control [23].

SAI introduces abstractions that aim to reconcile real-time DSP, interactive control, and data-centered representations in a unified model. SAI can therefore be seen as a hybrid architectural style (the approach taken here), or as a more general model in which the ones listed above can be characterized as *architectural patterns*.

SAI's abstractions are high-level (architectural) ones that are independent of the programming models used to implement them. SAI is therefore different in nature from programming and application frameworks such as CLAM [1]. (The relationship between SAI and code libraries will be addressed below.) SAI abstractions are also independent of any visual metaphors that might be employed to manipulate them.

3. THE SAI FRAMEWORK

SAI constitutes a framework in the sense that it defines a set of concepts and abstractions that together constitute a model of some application domain. The model can be instantiated to represent a particular application. In an attempt to providing a unifying model, the application domain targeted for SAI is, by choice, extremely general. The SAI model builds on three pillars: (1) an explicit account of time both in data and processing models; (2) the distinction between persistent and volatile data; and, (3) asynchronous parallelism. This section describes SAI's abstractions, formalized as an *architectural style*; a few interesting properties of the style; and, existing design and implementation tools.

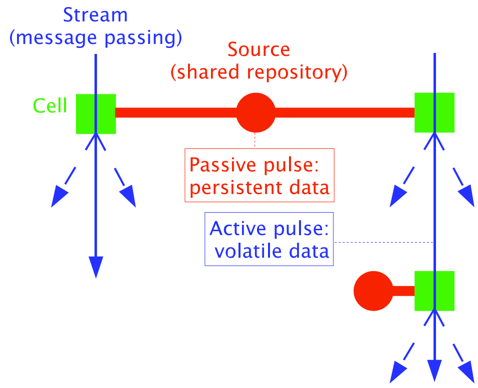


Figure 1: SAI elements and graphical notation.

3.1 Architectural style

SAI specifies a formal architectural style [16] comprising of an extensible data model and a hybrid (shared memory and message-passing) distributed asynchronous parallel processing model. Figure 1 presents an overview of SAI defining elements in their standard graphical notation.

In SAI, all data is encapsulated in *pulses*. A pulse is the carrier for all the synchronous data corresponding to a given time stamp. Information in a pulse is organized as a mono-rooted composition hierarchy of *node* instances. The nodes constitute an extensible set of atomic data units that implement or encapsulate specific data structures. Pulses holding volatile data flow down streams defined by connections between processing centers called *cells*, in a message passing fashion. They trigger computations, and are thus called *active* pulses. In contrast, pulses holding persistent information are held in repositories called *sources*, where the processing centers can access them in a concurrent shared memory access fashion. Processing in a cell may result in the augmentation of the active pulse (input data), and augmentation and/or update of the passive pulse (process parameters). The processing of active pulses is carried in parallel, as they are received by the cell. Data binding is performed dynamically in an operation called *filtering*. Active and passive *filters* qualitatively specify, for each cell, the target data in respective pulses. This hybrid model combining message passing and shared repository communication, combined with a unified data model, constitutes a universal processing framework.

A particular system architecture is specified at the conceptual level by a set of source and cell instances, and their inter-connections. Specialized cells may be accompanied by a description of the task they implement. The logical level specification of a design describes, for each cell, its active and passive filters and its output structure, and for each source, the structure of its passive pulse.

In the graphical notation, cells are represented as squares, sources as circles. Source-cell connections are drawn as fat lines, while cell-cell connections are drawn as thin arrows crossing over the cells. When color is available, cells are colored in green (reserved for processing); sources, source-cell connections, and passive pulses are in colored in red (persistent information); and, streams and active pulses are colored in blue (volatile information).

3.2 Architectural properties

By design, the SAI style shares many of the desirable

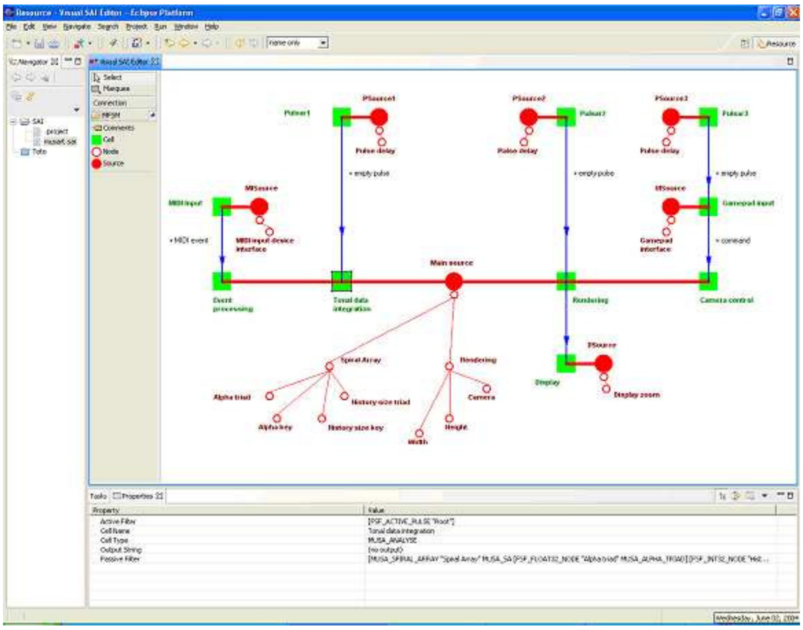


Figure 2: VisualSAI: a prototype Integrated Visual Architecture Design and Analysis Environment. The graph shown is that of the MuSA.RT system.

properties of classical dataflow models. The explicit representation of the flow of data allows for the intuitive design and fast high level understanding of a system's components and their interactions. The modularity of the model facilitate distributed development and testing of particular elements, and easy maintenance and evolution of existing systems. SAI also naturally supports distributed and parallel processing. The SAI style provides unified data and processing models for generic data streams, allowing for the simultaneous modeling of DSP, control, and data-centered approaches.

The underlying asynchronous parallel processing model promotes designing for optimal (theoretical) system latency and throughput. SAI abstractions do not impose arbitrary hard real-time constraints, but rather suggest (and enable) the implementation of synchronization mechanisms *only* when needed. On current hardware, most interactive systems do not require hard synchronization, but rather best-effort performance for low-latency, complemented by time consistency checks. Modeling systems as inherently dynamic, and with explicit account of real-time, leads to the natural expression and efficient implementation of such requirements.

The distinction between volatile and persistent data is key to providing a consistent model for both process- and data-centered approaches. The resource and instance models described in [10] illustrates the collapse of these two notions in classical models: the instrument is a persistent model while a note generated by the instrument is a volatile message. The examples described in the next section offer other illustrations of persistent and volatile data.

SAI spawns a continuum of intermediate-level representations from conceptual to physical specifications, a property reinforced by the graphical notation. SAI promotes the encoding of system logic in the structural organization of simple computing components rather than in the complexity of the computations carried by individual components. SAI designs exhibit a rich variety of structural and functional architectural patterns, whose systematic study will produce tools for assisting in design and re-use.

3.3 Design and implementation tools

An open source architectural middleware called Modular Flow Scheduling Framework (MFSM) [21], developed in C++, provides cross-platform code support for SAI's architectural abstractions, in the form of an extensible set of classes. MFSM is (heavily) multi-threaded and transparently leverages hyper-threading and multiprocessor architectures. A number of software modules regroup specializations that implement specific data structures, algorithms and/or functionalities. They constitute a constantly growing base of open source, reusable code, maintained as part of the MFSM project. In particular, the MFSM framework facilitates the leveraging of existing third-party code libraries through encapsulation. Extensive documentation, including a user guide, a reference guide, and tutorials, are available on the project Web site.

The graphical and compositional nature of the SAI model suggests the creation of integrated visual design and analysis environments. The formal nature of the model make it suitable for automatic static analysis and code generation (currently under development).

Figure 2 shows a screen shot of VisualSAI [29], a prototype of the user interface for such an environment, implemented as a plug-in for the Eclipse platform [12]. (The SAI graph shown is that of the MuSA.RT system described below.) As already noted, the SAI model is not linked to any visual metaphor, so that custom environments dedicated to specific activities are possible.

4. EXAMPLE SYSTEMS

This section describes two examples of interactive music systems designed using the SAI model. Both exhibit architectural patterns typical of interaction [14]. They are implemented using the MFSM middleware and share a sig-

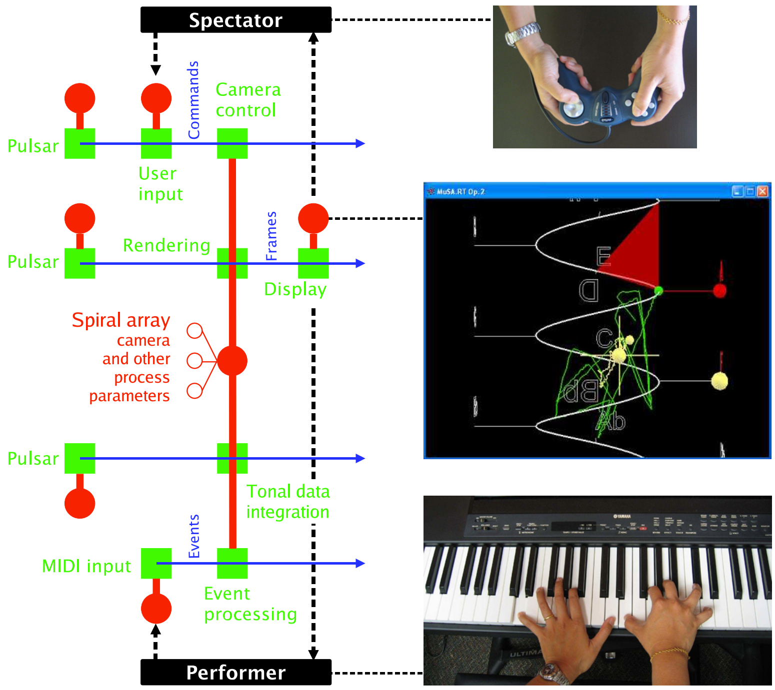


Figure 3: MuSA.RT synopsis with conceptual level system architecture in SAI notation.

nificant amount of code that exist in the form of MFSM modules.

4.1 The MuSA.RT System

MuSA.RT (Music on the Spiral Array . Real-Time) [7, 6] is a system for real-time analysis and interactive visualization of tonal patterns in music. Figure 3 presents a synopsis of the project with a conceptual level system architecture in SAI notation.

The system processes and analyzes MIDI input, for example captured during a live performance, in real-time, and maps the data to the Spiral Array [5], a 3D model for tonality. A center of effect (CE) summarizes contextual tonal information by mapping any pitch collection to a spatial point, and any time series of notes to meaningful trajectories, inside the Spiral Array. CE trajectory analysis allows the viewer to infer the presently active set of pitch classes, and higher level constructs, such as the current chord and key, revealed through real-time 3D rendering. An operator can concurrently navigate through the Spiral Array space using a gamepad to zoom in and out, tilt the viewing angle and circle around the spiral to get a better view of the tonal structures. An automatic pilot option seeks the best view angle and centers the camera at the heart of the action.

The system therefore combines on-line analysis of music data, computation of dynamic music structures, and real-time synthesis of visual data with interactive parameter adjustments. These computations are defined by four independent streams: (1) MIDI input and event processing; (2) tonal analysis (real-time CE algorithms); (3) rendering of the Spiral Array structures; and, (4) control device (gamepad) input and camera manipulation. These four streams potentially operate according to different modalities (e.g. push or pull input models) and at different rates. The Spiral Array structure, processing and rendering parameters are persistent (yet dynamic) data; the MIDI messages and the rendered frames for visualization are volatile data.

The precise scheduling and synchronization of multiple data streams processed and synthesized in real-time would constitute a major challenge in creating the MuSA.RT system adopting a traditional approach. Instead, the SAI model provides the tools to design an architecture that ensures best achievable latency between input and visual feed-back to the spectators.

From an engineering point-of-view, the complexity of such cross-disciplinary experiments is traditionally limited by actual system integration, which is the main source of unforeseen problems. Using SAI and MFSM greatly simplified system design, implementation and integration. From research point-of-view, the MuSA.RT system constitutes a platform for testing and validating the different modules involved. Each functional module can be replaced by a functionally equivalent module, allowing strictly controlled comparisons in an otherwise identical setting.

4.2 The ESP System

ESP(The Expression Synthesis Project) [8] is a driving interface (wheel, pedals and display) for creating expressive performances from expressionless music files. The use of a compelling and intuitive metaphor makes high-level expressive decisions accessible to non-experts. Figure 4 presents a synopsis of the project with a conceptual level system architecture in SAI notation.

The performer drives the car through a road in a virtual world. The driving (specifically, the state of the dynamic car model) directly controls the tempo and loudness of the music as it unfolds over time. The turns in the road are based on music structure, and guide the user's expressive choice.

The ESP system combines analysis of user input, on-line computations of the dynamic model, and real-time synthesis of synchronous visual and music data. These computa-

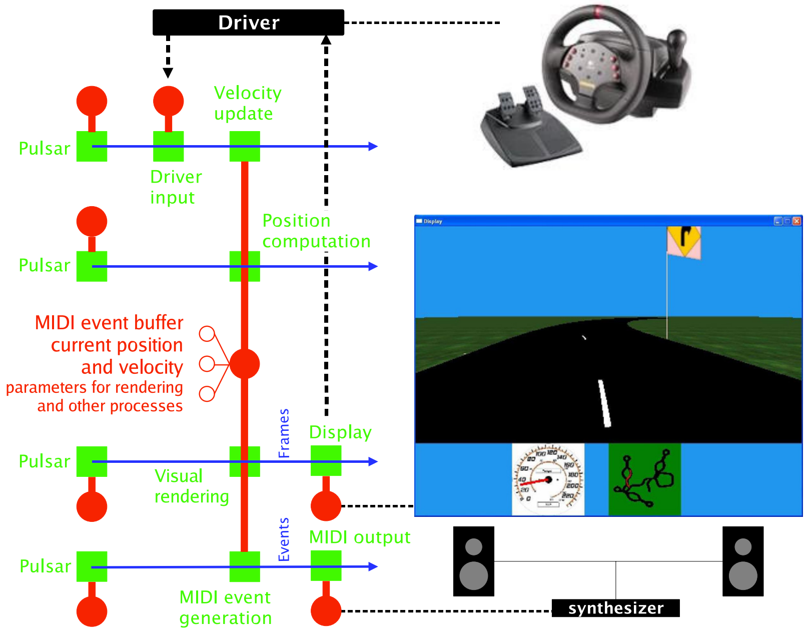


Figure 4: ESP synopsis with conceptual level system architecture in SAI notation.

tions are defined by four independent streams: (1) control device (driving wheel and pedals) input and velocity update for the car model; (2) discrete time integration of the dynamic equations for the car model; (3) visual rendering of the driver's view (including dashboard instruments); and, (4) aural rendering (MIDI event generation).

The architecture of the ESP system exhibits similarities in structure and function (patterns) with that of the MuSA.RT system. The score (MIDI event buffer), car model, processing and rendering parameters are persistent (yet dynamic) data; the MIDI messages and the frames rendered for visualization are volatile data.

The ESP system does not explicitly implement any hard synchronization mechanism to ensure that the music synthesized and visual data are presented with some fixed temporal error bound. Although by no means ruled out by the SAI model, such a mechanism would be computationally expensive, and difficult to certify, given the complexity of the network of external software and hardware elements involved in the final result. Rather, the design minimizes latency on each stream. "Synchronous effect" is achieved when the time discrepancies are below the threshold of human perception, a performance level usually achievable on modern commodity computing platforms.

5. SUMMARY AND FUTURE WORK

This paper introduced the SAI framework in the context of computer music systems. SAI defines high-level abstractions that form a general formalism for the design, analysis and implementation of complex software systems of asynchronous interacting processing components. The open source architectural middleware MFSM provides an extensible set of classes that implement SAI's architectural abstractions.

The most significant innovation of the SAI framework is its ability to handle real-time DSP, interactive control, and data-centered representations in a unified model. This generality facilitates the design and implementation of complex interactive systems that combine music analysis, synthesis, and on-line control. For example, Figure 5 shows a possible architecture for a real-time improvisation system such as that of Assayag and Dubnov [2]. The development of audio- and music-oriented code modules, especially modules encapsulating existing libraries, will help motivate the use of the framework for music system design and implementation.

The SAI framework also aims to facilitate the design and development of interactive music systems that involve other input and output modalities, such as graphics and vision. The collection of available modules pertaining to these domains is more substantial and constantly growing (see for example the open source WebCam Computer Vision project [30]).

As research on theoretical aspects of SAI and tool development continue, it is the adoption of the framework in various interacting fields of research that bears the promise of ground-breaking cross-disciplinary explorations.

6. ACKNOWLEDGMENTS

Jie Liu coded most of the ESP system, after initial coding contributions by Aaron Yang. Gérard Assayag and Belinda Thom provided valuable feed-back on this manuscript.

This work was supported in part by the Integrated Media Systems Center, a National Science Foundation (NSF) Engineering Research Center, Cooperative Agreement No. EEC-9529152; and by the NSF under Grant No. 0347988. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the NSF.

7. REFERENCES

[1] X. Amatriain. *An Object-Oriented Metamodel for Digital Signal Processing with a Focus on Audio and Music.* PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2004.

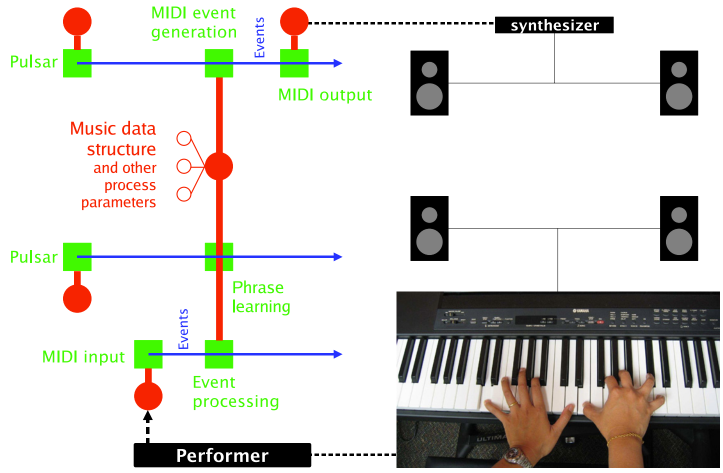


Figure 5: Possible architecture for an improvisation system.

[2] G. Assayag and S. Dubnov. Using factor oracles for machine improvisation. *Soft Computing*, 8:1–7, 2004.

[3] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue. Computer assisted composition at IRCAM: PatchWork & OpenMusic. *Computer Music Journal*, 23(3), 1999.

[4] J. Bresson, C. Agon, and G. Assayag. OpenMusic 5: A cross-platform release of the computer-assisted composition environment. In *Proc. 10th Brazilian Symposium on Computer Music*, Belo Horizonte, Brazil, 2005.

[5] E. Chew. *Towards a Mathematical Model of Tonality*. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[6] E. Chew and A. R. François. MuSA.RT - Music on the Spiral Array . Real-Time. In *Proceedings of ACM Multimedia*, Berkeley, CA, USA, November 2-8 2003.

[7] E. Chew and A. R. François. Interactive multi-scale visualizations of tonal evolution in MuSA.RT opus 2. *ACM Computers in Entertainment*, 2(4), October-December 2005.

[8] E. Chew, A. R. François, J. Liu, and A. Yang. ESP: A Driving Interface for Expression Synthesis. In *Proc. New Instruments for Musical Expression*, Vancouver, B.C., Canada, May 2005.

[9] R. B. Dannenberg. A language for interactive audio applications. In *Proc. International Computer Music Conference*, San Francisco, CA, 2002.

[10] R. B. Dannenberg, D. Rubine, and T. Neuendorffer. The resource-instance model of music representation. In *Proc. International Computer Music Conference*, Montreal, Quebec, Canada, 1991.

[11] S. Dubnov and G. Assayag. Improvisation planning and jam session design using concepts of sequence variation and flow experience. In *Proc. International Conference on Sound and Music Computing*, Salerno, Italy, November 2005.

[12] Eclipse. *www.eclipse.org*.

[13] R. B. (Ed.). *The Csound Book*. MIT Press, Cambridge, MA, 2000.

[14] A. R. François. Components for immersion. In *Proc. IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 2002.

[15] A. R. François. Software architecture for immersipresence. Technical Report IMSC-03-001, Integrated Media Systems Center, University of Southern California, December 2003.

[16] A. R. François. A hybrid architectural style for distributed parallel processing of generic data streams. In *Proc. International Conference on Software Engineering*, pages 367–376, Edinburgh, Scotland, UK, May 2004.

[17] A. R. François. Software Architecture for Computer Vision. In G. Medioni and S. Kang, editors, *Emerging Topics in Computer Vision*, pages 585–654. Prentice Hall, Upper Saddle River, NJ, 2004.

[18] M. Matthews. *The Technology of Computer Music*. MIT Press, Cambridge, MA, 1969.

[19] Max/MSP. *www.cycling74.com*.

[20] D. McLeod, U. Neumann, C. Nikias, and A. Sawchuk. Integrated media systems. *IEEE Signal Processing Magazine*, 16(1):33–43, Jan. 1999.

[21] Modular Flow Scheduling Middleware. *mfsm.sourceForge.net*.

[22] OpenMusic. *recherche.ircam.fr/equipes/repmus/OpenMusic*.

[23] OpenSound Control. *www.cnmat.berkeley.edu/OpenSoundControl*.

[24] M. S. Puckette. Pure data: another integrated computer music environment. In *Proc. Second Intercollege Computer Music Concerts*, pages 37–41, Tachikawa, Japan, 1996.

[25] M. S. Puckette. Max at seventeen. *Computer Music Journal*, 26(4):31–43, 2002.

[26] M. S. Puckette. A divide between 'compositional' and 'performative' aspects of Pd. In *Proc. First International Pd Convention*, Graz, Austria, 2004.

[27] B. L. Vercoe. Extended Csound. In *Proc. International Computer Music Conference*, pages 141–142, Hong Kong, China, 1996.

[28] B. L. Vercoe and D. P. Ellis. Real-time Csound: software synthesis with sensing and control. In *Proc. International Computer Music Conference*, pages 209–211, Glasgow, Scotland, 1990.

[29] VisualSAI. *visualsai.sourceForge.net*.

[30] WebCam Computer Vision. *wccv.sourceForge.net*.

Transmodal Feedback as a New Perspective for Audio-visual Effects

Christian Jacquemin
LIMSI & Université Paris 11
BP 133
91403 ORSAY, France
FirstName.FamilyName@limsi.fr

Serge de Laubier
Puce Muse
2 rue des Pyrénées, Silic 520
Wissous 91320 RUNGIS Cedex, France
sdl@pucemuse.com

ABSTRACT

A new type of feedback is presented that involves both the auditory and visual modalities. It combines an audio resonant bandpass filter, a geometrically constructed mass-spring system and its graphical skin. The system shows a resonant behavior that is detailed in various parameter setups. Complex mass-spring topologies result in a coherent self-sustained audio-visual system that mimics gusts of wind blowing a veil and associated sound effects.

Keywords

Audio-visual composition, Transmodality, Feedback

1. TRANSMODAL FEEDBACK

The twentieth century has seen a very large body of work concerning the connection between the visual and acoustic modalities and, more specifically, between sound, music, light, and image. Most of these works can be classified as *transmodal*: either using images to generate sound, or analyzing sound and music to generate graphics that can in turn be used to modify sound and music [1].

Another line of artistic exploration concerns the connection of one modality with itself: the notion of *feedback*. First considered as a undesirable effect, audio feedback has been appropriated by pop musicians such as The Who and Jimy Hendrix as an interesting ornamentation of their music in which their instrument (a guitar) was used as a control filter. Audio feedback can be considered as an *intra-modal* system that uses sound to generate sound.

Our purpose in this work is to explore the potentialities of the combination of trans- and intramodal communications in what we term *transmodal feedback*. How can a system for audio↔graphic feedback be designed, in which sonic output is used as input for graphical synthesis, that is in turn fed into the sound generator?

We first analyze some transmodal applications which offer interesting insights into the correspondences that can be established between the audio and graphic modalities. Then, a transmodal feedback system that combines physical modeling, graphical rendering, and a sound resonator is presented. Last, several variations are proposed in order to illustrate different parametrizations and renderings.

2. TRANSMODAL CORRESPONDENCES

The correspondences between two modalities tend to be metaphorical when they are used for artistic and creative purposes, and tend to be more literal when they are used for control purposes.

In the metaphorical category, and connecting sound to graphics, is the work of Golan Levin. The sound (the voice) is transformed into illustrative graphical effects inspired from the cartoon world [9]. Similarly, rich graphical environments such as urban models can be easily associated with sonic interpretations [19]. Metaphorical representations introduce a distance between the source stimulus (image or sound) and its perceived effect. For this reason they are not appropriate for feedback effects which require better coherence between input and output.

Literal transmodal correspondences, that are better suited to feedback, are encountered in systems where a modality is used to control another one. One of the motivations behind these works is that human perceptual capabilities depend on the modality. For instance, vision is very good at distinguishing visual patters in large sets of visual data, while audition is good at perceiving very brief sound variations.

Visual representation of music is a literal correspondence between graphics and audio that has its origin in the notation of music through scores. Digital media have offered new perspectives to interactive composition through the graphical representation of musical composition. It can be based on sophisticated musical theories such as Xenaki's theory for Iannix [5] or more abstract representations such as Sonos [16] or Metasynth [11]. Similarly virtual instruments are visual interfaces for music synthesis that focus on playability, direct manipulation, and real-time interaction [8].

Duality of sonic and visual representation is also well illustrated by visual representations of sound databases [15] that can help the user to build a mental map of the soundscape of the sample collection. The reverse combination of sound and graphics is abstract data sonification: the process of representing generic data by means of audio signals [2].

Since our purpose is to close the loop and allow reciprocal transmodal information exchange, we return to the notion of feedback in a resonating system before introducing our model of audio↔graphic feedback.

3. RESONANCE AND FEEDBACK

3.1 Audio and Video Feedback

Unimodal feedback is the process of capturing the signal produced by an emitter in a modality (typically a loudspeaker for sound) and reamplifying it. It is illustrated by Figure 1. It generally involves the contribution of an external trigger source that plays a more important role in video feedback than in audio feedback.

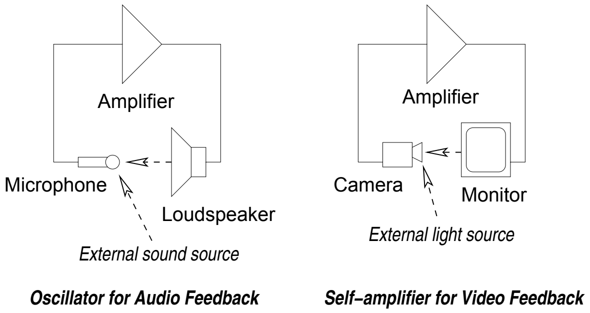


Figure 1: Unimodal Feedback.

The “classic” audio feedback (also known as Larson effect) occurs when an amplifier receives as input its own output. The loop results in an increasingly loud signal until the limits of the amplifier are reached. Audio feedback can be seen as an echo with very short delay defined by the characteristics of the system (distance between loudspeaker and microphone, amplifier, characteristics of I/O devices, room...), and transforms it into an oscillator. The selected frequencies correspond to the Barkhausen effect and are such that the input and output signals are in phase (with additive intensities) and the gain is slightly above 1. Since the amplified signal is mainly controlled by the characteristics of the system, the external sound source plays the role of a trigger and the output pitch is dominated by resonance frequencies.

What is known as video feedback is by nature very different from audio feedback since it relies only on gain and not on oscillation. For this reason, all colors are equally subject to amplification, contrary to audio feedback that amplifies a very narrow band of frequencies. Periodicity in video feedback occurs in space and not in time, and results in tiling or kaleidoscopic effects whose base graphical components are defined by the external signal. (Visual perception occurs in time and in space, but only spatial perception involves resonance and periodicity.)

3.2 Transmodal Feedback

In order to design the architecture of a transmodal feedback system, we must establish a reciprocal communication between a graphical and an audio application so that the signal emitted by one component is accepted by the other one. For this purpose we use networked applications and encapsulate transmitted data via network messages. The overall architecture is given in Figure 2 in which emitters have been preserved for human access to the system output, but sensors (microphone and camera) have been removed because they are not necessary any more (even though unimodal feedback could be combined with transmodal feedback).

The design of an audio↔graphic oscillator is not as straightforward as it is for a pure audio system. First there is a temporal inconsistency between the processing delays of the audio, graphic, and communication systems. The processing delays in a graphic system are higher than or equal to the frame refresh rate (typically 40ms). They cumulate with the communication delays between the audio and graphic system (around 1ms). In an audio system, the delays are close to the period of the sound signal (a few μs). The processing delays of an audio↔graphic system are controlled by the frame rate, and therefore greater than 40ms.

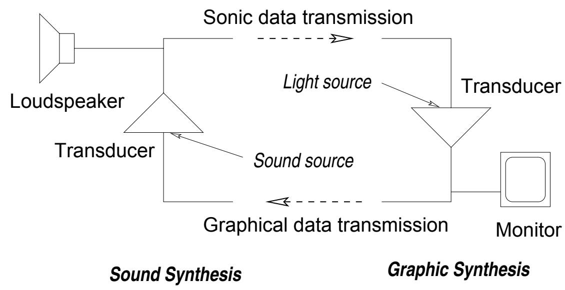


Figure 2: Transmodal Feedback.

A second temporal inconsistency concerns the emitted signals. The phase of the visual signal is several orders of magnitude higher than the phase of the audio signal, which is in turn much higher than the delays involved in a looping audio↔graphic system. The system cannot work as an oscillator as discussed for an amplifier in pure audio feedback.

When comparing unimodal audio and video feedbacks, it appears that audio feedback offers a richer domain of experimentation because of its double nature: phase coincidence (the signal is tuned to the characteristics of the system) and self-reinforcement. It seems therefore desirable to build a system which will act as a resonator. Since we cannot work on the signal directly (because of the second temporal inconsistency), oscillations will concern higher level audio parameters such as envelope or pitch. Because of the first temporal insconsistency, the resonator frequency must be lower than 25hz, possibly much lower.

The architecture proposed in Figure 2 has no reason to be an oscillator if the transmitted data are not periodic. In order to equip the application with a generator of periodic signals, the graphical component is complemented with a mass-spring system (MSS) that directly controls the graphical output, and indirectly the sound generation. We now turn to the implementation of this architecture and its two major building blocks: a skinned MSS on the graphical part, and a resonator related to the MSS dynamics on the audio part. The application is named *GraphSon*.

4. AUDIO↔GRAPHICS FEEDBACK

The architecture of *GraphSon* is made of two networked applications: an audio patch under *Max/MSP* [10] that implements a resonant bandpass filter externally controlled by the speed and acceleration of the graphical elements, and a virtual 3D scene under *Virtual Choreographer* (VirChor) [18] that is made of a skinned MSS parametrized by the sound envelope derived from the audio patch. Data exchange between these components is made through OSC. Figure 3 show the instantiation of Figure 2 in the case of *GraphSon*.

4.1 Graphical and Physical Components

Mapping is considered as an important issue in the de-

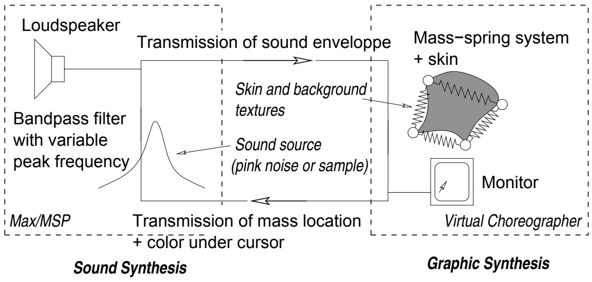


Figure 3: *GraphSon* Architecture.

sign of virtual instruments and concerns the "intelligent" and sensitive association between a musician's gestures and the control of his/her instrument. Mapping tends to be considered not just as an interface, but as an autonomous component in virtual instruments. Because of their intuitive and rich behavior, physical models can be used as mapping devices that produce complex and variable responses to stimuli: for instance, obstructions in particle flows and resulting collisions (*FlowField* [4]), or MSSs and their complex dynamics (*GENESIS* [3] or *PMPD* [12]).

Our interest for such systems in this work is not for the purpose of mapping human stimuli to musical synthesis, but for the introduction of a resonator in our audio↔graphic feedback loop parameter (Figure 3). The MSS associates input sound envelope values to a graphical output through an indirect mechanism.

In a MSS, the equation that controls the dynamics of a mass M_i that is linked to n_i masses $M_{i,j}$, is

$$m_i.x'' = -d.x' + m_i.g_x + \sum_{j=1}^{n_i} k_{i,j}(d(M_i, M_{i,j}) - lini_{i,j}) \quad (1)$$

in which m_i is the mass of M_i, d the viscous damping coefficient, g the gravity, $k_{i,j}$ the spring constants, and $lini_{i,j}$ the lengths of the unstretched springs. Sound envelope e is used to modify dynamically two of the MSS characteristics: its damping factor and the spring elasticity

$$d = k_{damp}.e \text{ and } \forall i, j \; k_{i,j} = k_{elast}.e \quad (2)$$

The audio-visual effect is that high sounds result in a stiff and constrained MSS (mild and sustained wind in a non extensible veil), while low sounds result in a weak and free MSS (strong gusts of wind in a light and extensible veil). In the second case, the potential energy accumulated in the veil can be released suddenly and transformed in kinetic energy. Such a correlation produces perceptually plausible correspondences between audio and graphics [7].

The graphical scene is implemented in *VirChor*. The `<graph>` element describes a MSS, and the `<patch>` element a Bezier patch. At each frame, a script is executed that reconnects the control points of the skin to the masses of the MSS. Two models are designed according to table 1 and illustrated by Figure 4. A quad is used as skin in the simplest model $GraphSon_2$.

The target application is $GraphSon_{4\times4}$ because it offers richer behaviors, and better graphical renderings and animations. It combines a 4×4 MSS with a grid topology and a bicubic Bezier patch defined by 16 control points (masses at nodes, springs for inter-connectivity). The simplest application is used for analyzing the parameter effects and resonating behaviors in section 5 under simpler experimental conditions and fewer parameters.

Table 1: Parameters of *GraphSon* Instances.

Name	Masses (fixed)	Springs	Skinning
$GraphSon_2$	2 (1)	1	Quad
$GraphSon_{4\times4}$	16 (2)	24	Patch 4×4

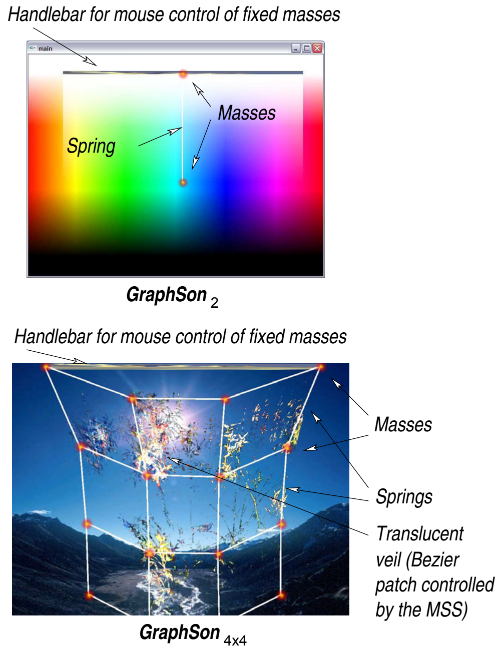


Figure 4: Two instances of *GraphSon*: Gestures are transmitted to the upper masses of a MSS that controls an animated translucent veil.

4.2 Audio Component

For audio-visual coherence purposes, the sound generated from the graphical output is intended to reproduce the noise of a veil in the wind. The effect is obtained by using a pink noise source (the wind) filtered by a digital bandpass filter that produces high pitch noise for strong gusts of wind.

The filter is controlled by its quality Q, its gain G, and its center frequency f_{res}. The higher the quality, the shorter the bandwith, and the higher the output at the resonance frequency. The second order equation used for the filter is

$$y_n = G(x_n - r.x_{n-2}) + c_1.y_{n-1} + c_2.y_{n-2} \quad (3)$$

r, c_1, and c_2 are parameters calculated from f_{res} and Q.

In order to produce a satisfactory audio effect, the resonance frequency is controlled by the acceleration of masses in the bottom line. Strong accelerations of these masses correspond to high pitch output, giving the impression of a strong wind blowing the veil.

The resonance filter is implemented in *Max/MSP* with the `reson~` object that has 4 inputs: an audio signal and 3 digital values G, f_{res}, and Q. Equation (3) is taken from [10]. The frequency f_{res} is a linear function of the acceleration of one of the masses in the MSS. It is computed in the audio patch from the values of the mass location received from the graphical component. The output of `reson~` is the filtered audio input, pink noise produced by the object `pink~`. The envelope of the output audio signal, sent to the graphical component, controls damping and elasticity.

5. FEEDBACK CONTROL AND ANALYSIS

We now turn to the study of the resonating audio↔graphic feedback loop under various parameter values. The behavior of the feedback loop depends on several factors: the topology of the MSS, the parameters of the audio system including the nature of the base sound (noise or sample), the transmission delays through the network, and the motion of the controlled mass by the user. This section is intended to provide better insight of the basic echo resonance in the system in its simplest form: pink noise and a 2-mass 1-spring system. More detail is also provided on the parametrization of the system and its effect on the animation of the graphical scene and the audio output.

5.1 Basic System

If the simplest MSS (*GraphSon*$_2$ presented in 4.1) is connected to the resonance filter fed with a pink noise a periodic behavior is observed, illustrated by Figure 5. In this figure two values are plotted that trace the dynamics of the audio and graphic systems:

- the height of the lower mass, the free mass since the other one is fixed to the handle (dotted line),
- the sound level which is used to control damping and spring coefficient (solid line).

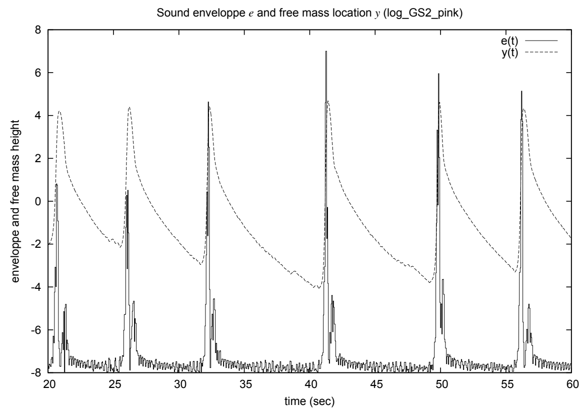


Figure 5: *GraphSon*$_2$ Basic Resonating System (see upper part of Figure 4): 1 Fixed Mass, 1 Vertically Moving Mass, 1 Spring, and Pink Noise.

The basic behavior can be described as follows. When the mass reaches its lowest position (maximal extension of the spring), it slows down, decreases the pitch of the resonance frequency, and increases the Q of the filter. This results in a weaker sound that in turn decreases damping and spring coefficients. Because of low damping values, the MSS becomes more reactive to small movements of the lower mass and the spring then retracts very quickly.

The use of various sound samples does not modify significantly the behavior of the resonator, even though it has a strong impact on the audio output. Several tests were made with various kinds of music: piano romantic music, techno/world music, natural sound effects... but none had a strong impact on the system behavior. Such observations are coherent with resonating audio feedback, in which resonance is controlled by the system characteristics and the trigger sound plays a secondary role.

5.2 Color Parametrization

In order to provide the user with easy access to the parametrization of the audio system (and also indirectly on the graphical system), the red, green, blue components of the color under the mouse cursor are transmitted to the audio patch and associated with parameters of the audio resonator. The associations are made as follows:

- The green value controls a multiplicative factor of acceleration that defines f_{res} and also controls G,
- the red value controls Q (the height and width of its bandwith),
- the red and blue values bring an additional additive factor to f_{res}.

The color can be used in two ways. It can either be used as a control device for the user. If she/he moves the mouse cursor on the background image, various responses are obtained from the system. Color can also be used in a more passive way by placing the mouse cursor on the animated veil. Then the variation of colors under the mouse cursor results in dynamic modification of audio parameters that reciprocally modify the animation and rendering of the audio scene.

Various types of veil colorings are used to produce different color variations and thus different behaviors of the feedback loop. In Figure 4 above, two types of veils are shown. In the upper snapshot, a blended semi-translucent veil is used: from white opaque at the top to translucent at the bottom. The bottom snapshot shows a more complex rendering of the veil that is implemented through shaders: the veil color is the composite of several semi-transparent textures combined with masks. The transparency parameters of textures and masks are computed from dynamic geometrical characteristics of the veil and vary according to its dynamics.

The combined effects of color and veil motion are shown in Figure 6. The color under the mouse cursor is the blending of a red background color and the semi-transparent white color of the veil. Because of the high value of the red channel, the audio resonator is a sharp filter with a narrow bandwith. Because of the veil motion, when the veil drops the color under the mouse cursor becomes whiter, which makes blue and green values higher, and thus tend to reactivate the audio system. The combination of these two effects gives the resonator of the feedback loop a smaller period than it had without the veil (compare Figure 5 for fixed pink color and mouse cursor outside the veil and Figure 6 with mouse cursor over the veil).

5.3 Complex System Behavior

We now turn to GraphSon$_{4\times4}$, a MSS made of a 4×4 grid of masses that controls a bicubic Bezier patch. As for simpler systems, we plot on the same graph the location of the lowest left mass and the audio level.

Because of the more complex internal dynamics of its MSS, the loop resonance is not as clear as it is in the case of a 2-mass 1-spring system. The veil has its own internal short term dynamics that combine with the longer term loop dynamics. The loop resonance is easier to detect for low color values associated with a soft filter (Figure 7).

If the audio system receives a bright color associated with high parameter values for gain and quality, the output level is higher and higher values are sent for damping and spring coefficients. The veil has short amplitude movements with very short periods. Periodicity is much more

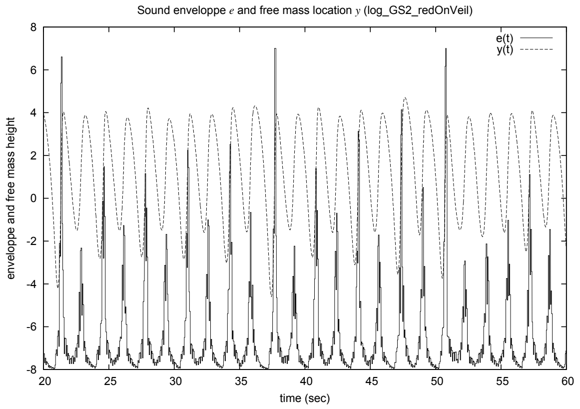


Figure 6: MSS Topology of Figure 5 and Pink Noise, with an Additional Control through the Color under the Mouse Cursor. The Cursor is Located over a Semi-transparent White Veil and a Red Background Color.

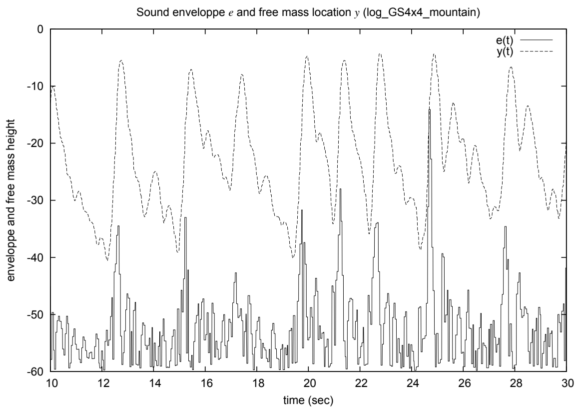


Figure 7: *GraphSon*$_{4\times4}$ A MSS made of a Grid of 16 Masses (see lower part of Figure 4): 2 Fixed Masses, 14 Vertically Moving Masses, 24 Springs, and Pink Noise. Mouse Cursor on Dark Background Color (0,0.02,0.07).

difficult to detect in the resulting motion and audio signal (Figure 8).

As for the previous simpler MSSs, the audio signal does not play an important role in the dynamics of the system. Other parametrizations of veil and sound should be considered if the purpose is to influence more strongly the loop resonance by the audio signal.

5.4 Combination with Gesture

The simplest system (*GraphSon*$_2$ and the mouse cursor on a static pink color) has an autonomous resonance that is shown Figure 5. If this system is manipulated by an operator who controls the location of the fixed mass (the upper mass), the system behaves as follows (see Figure 9):

1. during gesture control, the output follows the constrained motion of the upper mass (the values between the two vertical dotted lines),
2. when the manipulation is completed, the system has a transient chaotic behavior (5 to 10 seconds),
3. finally the periodic resonance restarts and begins by

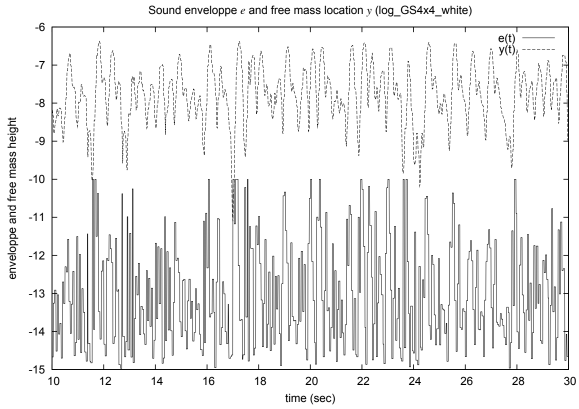


Figure 8: MSS Topology of Figure 7 and Pink Noise. Mouse Cursor on White Background Color (1,1,1).

a decreasing slope followed by a short peak.

These results show that strong gestures can control the system while they are executed and for a short time afterward, but the system quickly returns to its periodic behavior when the excited state is over.

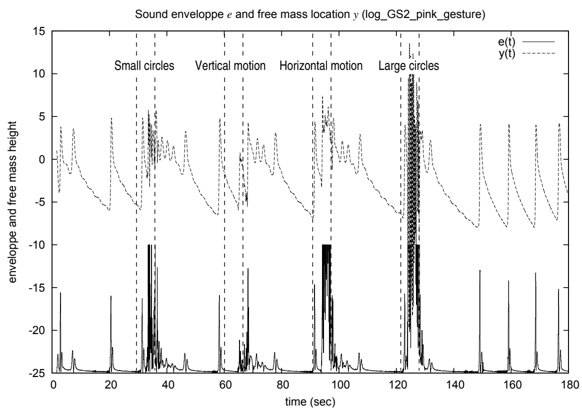


Figure 9: Gesture Mapping with *GraphSon*, Conditions of Figure 5.

6. SYNTHESIS AND PERSPECTIVES

In this study, we have presented a model and an application that build an audio↔graphic feedback loop made of a MSS and its visual skinning, and a resonant bandpass filter. Audio level is used to control the physical system dynamics, while mass acceleration controls the filter characteristics. In addition, color under mouse cursor directly parametrizes the filter and indirectly modifies the MSS reactivity. The loop actually behaves like a resonant system with a period between 2 and 5 seconds. Periodicity is better observed on a simple MSS or in quiet situations (soft filter and dark color).

Further studies could be carried out:

- The system dynamics can be studied formally in the simple case by taking into account the internal characteristics of the audio and graphic systems and the information propagation delays between the two components. The output of the formal study should be then compared with the dynamics observation in the computer model.

- The artistic or industrial applications of such an audio-visual environment for the realistic or non-realistic rendering of natural phenomena such as wind can be further investigated. Current works tend to study separately graphical and sonic modeling [13], but we are convinced that deeper investigations of the perceptual correlations between sound and image in the modeling of such natural phenomena are promising directions of research [7, 6]. It is therefore necessary to design new generations of audio-visual environments such as the one presented in this study to offer a framework for such studies on multi-modal modeling and perception.

- For sound creation purposes, richer parameter sets and richer topologies could be taken into consideration: other MSS topologies such as the ones explored by PMPD for audio-visual composition [12], other audio patches with physical modeling of wind phenomena such as the ones used for musical instruments [17], other color parameters such as hue, saturation, and value, and more complex visual renderings through physical cloth modeling or shaders and BTF textures.

- If the purpose is to design a virtual instrument that uses the feedback resonance for graphical and audio synthesis, gesture-based control should be investigated more deeply, possibly with haptic feedback [14]. High speed in graphical rendering through bitmap animation or decoupling of mass-spring animation and associated skinning would yield higher resonance frequencies in the audio↔graphic loop and produce interesting audio-visual patterns.

7. ACKNOWLEDGMENTS

This study has benefited from a research collaboration on virtual instruments between the authors and Hugues Genevois (LAM), Brian Katz (LIMSI), and Norbert Schnell (IRCAM). Many thanks to Brian Katz, Jean-Baptiste Thiebaut (Queen Mary Univ.), and the three anonymous reviewers for their comments on a preliminary version of the paper.

8. REFERENCES

[1] Audiosculpt. http://forumnet.ircam.fr/349.html.

[2] S. Barrass and G. Kramer. Using sonification. *Multimedia Systems*, 7(1):23–31, 1999.

[3] C. Cadoz, A. Luciani, J.-L. Florens, and N. Castagné. ACROE-ICA: Artistic creation and computer interactive multisensory simulation force feedback gesture transducers. In *Proceedings of New Interfaces for Musical Expression (NIME'03)*, pages 235–246, 2003.

[4] T. Chen, S. Fels, and T. Schiphorst. Flowfield: Investigating the semantics of caress. In *ACM Special Interest Group on Graphics and Interaction (SIGGRAPH'02)*, page 185, 2002.

[5] T. Coduys and G. Ferry. IanniX aesthetical/symbolic visualisations for hypermedia composition. In *Proceedings International Conference Sound and Music Computing (SMC '04)*, 2004.

[6] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. West, G. Pingali, P. Min, and A. Ngan. A beam tracing method for interactive architectural acoustics. *The Journal of the Acoustical Society of America (JASA)*, 115(2):739–756, 2004.

[7] J. K. Hahn, J. Geigel, J. W. Lee, L. Gritz, T. Takala, and S. Mishra. An integrated approach to motion and sound. *The Journal of Visualization and Computer Animation*, 6(2):109–124, 1995.

[8] S. Jordà. Sonigraphical instruments: From FMOL to the ReacTable. In *Proceedings of New Interfaces for Musical Expression (NIME'03)*, pages 70–76, 2003.

[9] G. Levin and Z. Lieberman. In-situ speech visualization in real-time interactive installation and performance. In *Proceedings of NPAR'04*, pages 7–14, 2004.

[10] MaxMSP 4.5 reference manual. http://www.synthesisters.com/download.

[11] MetaSynth. http://www.uisoftware.com/MetaSynth/.

[12] A. Momeni and C. Henry. Dynamic independent mapping layers for concurrent control of audio and video synthesis. *Computer Music Journal*, 30(1):49–66, 2006.

[13] S. Ota, T. Fujimoto, M. Tamura, K. Muraoka, K. Fujita, and N. Chiba. "1/fβ noise-based real-time animation of trees swaying in wind fields. In *Proceedings of Computer Graphics International (CGI'03)*, 2003.

[14] X. Rodet, J.-P. Lambert, R. Cahen, T. Gaudy, F. Gosselin, and F. Guédy. Sound and music control using haptic and visual feedback in the PHASE installation. In *Proceedings of New Interfaces for Musical Expression (NIME'05)*, pages 109–114, 2005.

[15] D. Schwarz. Recent Advances in Musical Concatenative Sound Synthesis at Ircam. In *Workshop Audio Mosaicing: Feature-Driven Audio Editing/Synthesis.* (ICMC'05), Barcelona, 2005.

[16] A. Sedes, B. Courribet, and J.-B. Thiebaut. Visualization of sound as a control interface. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX'04)*, 2004.

[17] J. Smith. Virtual acoustic musical instruments: Review and update. *Journal of New Music Research*, 33(3):283–304, – 2004.

[18] Virtual Choreographer 1.2 reference manual. http://virchor.sourceforge.net/html/.

[19] P. Waters and A. Rowe. Alt-space: Audio-visual interactive software for developing narrative environments. In *Proceedings CADE2004*, 2004.

Screen-Based Musical Interfaces as Semiotic Machines

Thor Magnusson

Creative Systems Lab
University of Sussex
BN1 9QN, East Sussex, UK
T.Magnusson@sussex.ac.uk

ABSTRACT
The ixi software project started in 2000 with the intention to explore new interactive patterns and virtual interfaces in computer music software. The aim of this paper is not to describe these programs, as they have been described elsewhere [14][15], but rather explicate the theoretical background that underlies the design of these screen-based instruments. After an analysis of the similarities and differences in the design of acoustic and screen-based instruments, the paper describes how the creation of an interface is essentially the creation of a semiotic system that affects and influences the musician and the composer. Finally the terminology of this semiotics is explained as an interaction model.

Keywords
Interfaces, interaction design, HCI, semiotics, actors, OSC, mapping, interaction models, creative tools.

1. INTRODUCTION
In our work with ixi software [14][15], we have concentrated on creating abstract screen-based interfaces for musical performance on computers. These are graphical user interfaces (GUIs) that do not necessarily relate to established conventions in interface design, such as using buttons, knobs and sliders, nor do they necessarily refer to musical metaphors such as the score (timeline), the keyboard (rational/discrete pitch organisation) or linear sequencing (such as in step sequencers or arpeggiators). Instead we represent musical structures using abstract objects that move, rotate, blink/bang or interact. The musician controls those objects as if they were parts of an acoustic instrument, using the mouse, the keyboard or other control devices. We have created over 15 of these instruments – each exploring new modes of interactivity where some of the unique qualities of the computer are utilised in fun, inspirational and innovative ways. Qualities such as remembering the musician's actions, following paths, interaction between agents, generativity, randomness, algorithmic calculations and artificial intelligence; all things that our beloved acoustic instruments are not very good at.

Over the course of our work, we have developed a loose and informal language for these instruments – a semiotics that suggest to the musician what the functionality of each interface element is, and what it signifies in a musical context. Human Computer Interface (HCI) research [2][3][17][1][6] is usually con-centrated on the chain of meaning from the software designer to the software user. The user is the *receiver* of information and the aim of HCI is traditionally to make the interaction between the two systems (the human and the computer) intuitive, representational and task based (where the tasks are based on real world tasks). What is lacking is a stronger discussion of the situation where the computer is used as a tool for artistic creation – an expressive instrument – and not a device for preparing, organising or receiving information. In artistic tools we have an important addition, where the signifying chain has been reversed: the meaning is created by the user, deploying a software to achieve some end goals, but this very software is also a system of representational meanings, thus influencing and coercing the artist into certain work patterns.

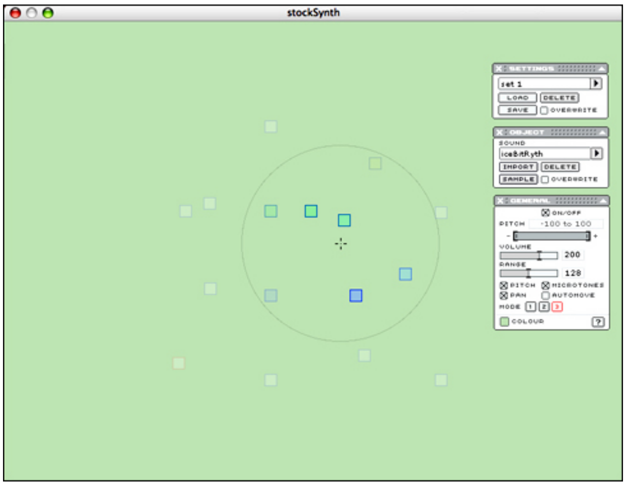


Figure 1: StockSynth. Here the crosshair cursor serves as a microphone that picks up sounds from the boxes that represent sound samples. The mic has adjustable scope (the circle). The boxes are moveable and the mic moves by drawn or automatic trajectories or by dragging it with the mouse.

2. A SHORT NOTE ON INSTRUMENTS
"Even simple physical instruments seem to hold more mystery in their bodies than the most elaborate computer programs" [10]

Both acoustic instruments and music software incorporate and define the limits of what can be expressed with them. There are special qualities found in both, but the struggle of designing, building and mastering an acoustic instrument is different from the endeavor of creating musical software. The acoustic instrument is made of physical material that defines the behaviour of it in the form of both tangible and aural feedback. These material properties are external to our thought and are something that we fight with when we design and learn to play instruments. Such features or characteristics of the material instrument are not to be found in software. Software is per definition programmed (etymology: "pro" = before, "graphein" = written); its functionality is prewritten by a designer or an engineer and the decisions taken in the design process become the defining qualities of the software, determining its expressive scope.

Different languages are based on different paradigms and lead to different types of approaches to solve a given problem. Those who use a particular computer language learn to think in that language and can see problems in terms of how a solution would look in that language.[1] [12]

This is not the place to go into the cognitive processes involved with learning and playing an instrument. But we are faced with an important question: what material (instruments) is the computer musician composing for and where does he or she get the ideas from? In other terms: where does the thinking (or composing) of the computer musician or digital instrument inventor take place? It happens most likely in the form and structure of the programming language in which he or she is working. The environment defines the possibilities and the limitations of what can be thought. But what does it mean to "learn to think in a language"? What are we gaining and what are we sacrificing when we choose an instrument or a programming environment? And what are the reasons for some people preferring one environment for another?

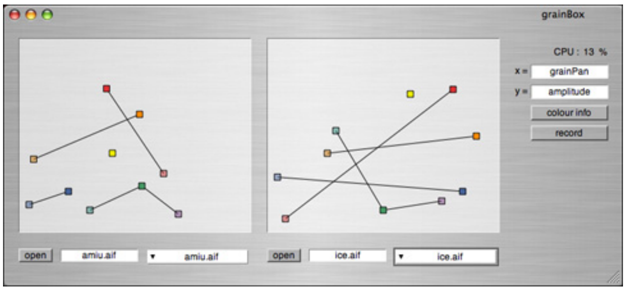


Figure 2: GrainBox. It can be hard to create interfaces for granular synthesis. The GrainBox is a suggestion how to represent the complex parameters as boxes with X and Y dimensions in 2D space and with connections to other parameters such as reverb and random functions.

When musicians use software in their work, they have to shape their work process according to the interface or structure of the software. As with acoustic instruments software defines the scope of potential expression. The musician is already tangled in a web of structured thinking but the level of freedom or expressiveness depends on the environment in which he or she working.[2] To an extent, the musical thinking takes place at the level of the interface elements of the software itself.

It is misleading then to talk of thinking as of a 'mental activity'. We may say that thinking is essentially the activity of operating with signs. This activity is performed by the hand, when we think by writing; by the mouth and larynx, when we think by speaking; and if we think by imagining signs or pictures, I can give you no agent that thinks. If then you say that in such cases the mind thinks, I would only draw attention to the fact you are using a metaphor, that here the mind is an agent in a different sense from that in which the hand can be said to be the agent in writing.

If again we talk about the locality where thinking takes place we have a right to say that this locality is the paper on which we write or the mouth which speaks. And if we talk of the head or the brain as the locality of thought, this is using the 'locality of thinking' in a different sense. [21]

If here I am attempting to find the "locus" of musical thinking/performing in both acoustic instruments and screen-based digital instruments – a discussion that is much deeper than can be delved into here – it is important to consider the difference in embodiment and incorporated knowledge of the player in those two types of instruments. When learning an acoustic instrument, the motor memory does most of the job and your learning "happens" as interaction with the body of the instrument. Due to the material qualities of it, one can never master an instrument, it always contains something unexplored, some techniques that can be taken further and investigated. With software however, it is more or less visual and procedural memory that is involved, as software doesn't have a material body that the musician learns to operate. The only "body" of software is in the form of its interface elements, and they (as opposed to the indicative nature of physical material) are simple, contingent and often arbitrary design decisions.[3] The "body" of the software has to be created and it does not depend upon any material qualities, but rather the style and history of graphical user interface design.

3. HCI AND SEMIOTICS

Designing is essentially a semiotic act. Designing a digital instrument or programming environment for music is to structure a system of signs into a coherent whole that incorporates some compositional ideology (or an effort to exclude it). The goal is to provide the users with a system in which they can express themselves and communicate their ideas in a way that suits their work methods and sometimes provide new ways of thinking and working. But what kind of a tool is the computer and what kind of communication are we talking about here?

3.1 Interaction Paradigms

We can roughly define three primary **interaction paradigms** in computer software as: *computer-as-tool, computer-as-partner,* and *computer-as-medium.* [6] Different research communities address these paradigms. The HCI field investigates the computer-as-tool paradigm but the attention is mainly on how to design understandable and ergonomic software for the user of the tool. What is lacking is a better understanding of creativity itself and how creative and experimental minds use software (and often have to misuse it to get their ideas across). We have learned from user feedback that there seems to be a general need for better sketching environments that can be modified according to the needs of the user. An interesting fact here is that many cutting-edge art works are created by hacking or modifying software or simply creating one's own tools. There are schools of artists that respond to the limitations of commercial software with their own software in the form of software art.[4] [11][16]

[1] Try to replace "language" with "instrument" in McCartney's paragraph above – the same applies for musical instruments as well.

[2] From this perspective SuperCollider and Pure Data are arguably more open and free than Logic, Protools or Reason, to name but a few.

[3] Often made by the wrong people: an engineer and not an ergonomist; a graphic designer and not a musician.

[4] The www.runme.org repository is an excellent source for information and examples of what is happening in the field of software art and generative art. It is closely related to the ReadMe festival, which was the first software art festival.

3.2 The Semiotics of a Creative Tool

The most common of semiotic practises is to look at the signifying channel from the sender to the receiver through some medium such as signs, language, text, or film. [5][9] The "work" here is a static construction that doesn't change after it has been published or released.[5] By contrast, computer-based works are interactive and can be changed or modified after their release either by users themselves or by updates. *Interaction becomes a new sign-feature*.[2] Some studies have been done on this new semiotic quality of the computer [1][2][3][7], but very few in the field of music software or other creative software.

In music software, the user is at the same time *the receiver* and interpreter of information from the designers of the software and *the sender* of information in the form of the music being composed using the tool. This dual semiotic stance is important in all tools (whether real or virtual) but becomes vital in contingently designed tools such as music software. Music software is a sign system in its own right, but the important question here is: which are the relevant layers of signification and communication and from where do the originate? This can be analysed into strata of different practices. The hardware designers, the programmers of the compilers, the language API and the software itself, the designers of the interaction and the programmers of the interface. A creative tool has history of important design decisions all shaping its scope and potential. This is a complex structure, but the user is faced with the question: what is the meaning conveyed in the interface? And is this system of signification not essentially of compositional nature? Who took those decisions and by which criteria?

The contingency of design mentioned above in relation to the digital medium is one of the most definable characteristic of it. We don't have this "contingency problem" when designing acoustic instruments as the properties of the material we work with leads us in our design: closing a hole in a flute increases the wavelength in the resonant tube and the tone deepens; pressing the string against the fingerboard of a guitar – shortening the wavelength – produces a note of higher pitch. When designing screen-based computer interfaces we can choose to imitate physical laws as known from the world of acoustic instruments, we can draw from the reservoir of HCI techniques or we can design something entirely new. It is here that interface design, the interaction design, and mapping becomes very important factor in the creation of interesting screen-based instruments for the computer.

4. INTERFACE ELEMENTS IN IXI

Most modern operating systems are graphical or allow for a graphical front end. The WIMP (Window, Icon, Menu, Pointer) interface [4] has become a standard practice and we have become used to the direct manipulation [20] of graphical objects. The traditional method is to translate work practices from the real world into the realm of the computer, and thus we get the folders, the documents, the desktop and the trash. In music applications we get representations of keyboards, buttons knobs and sliders, rack effect units and cables. This is also suitable where the aim is to translate studio work practices into the virtual studio. But when we are creating new instruments using the new signal processing capabilities and artificial intelligence of the computer, there might not exist any physical phenomena that we can use as source for our interface metaphors.[6]

4.1 Interaction Models

Each of the ixi applications is a prototype or a suggestion and it explores a specific mode of interaction. The whole of our software can be grouped into a specific kind of **interaction model**: a language, a semiotics or a design ideology that informs and en-forms the work. An interaction model can be defined as more operational than an interaction paradigm (computer as tool, partner or medium). [6] It can be evaluated according to the descriptive, the evaluative and the generative power of the model. These dimensions of evaluation are all important when creating an interaction model. The descriptive power is the ability to describe a significant range of existing interfaces; the evaluative power helps us to assess multiple design alternatives; and the generative power is the ability of the model to inspire and lead designers to create new designs and solutions.

4.2 Interaction Instruments

It is the generative aspect of ixi's interaction model that is the subject here. Beaudouin-Lafon's definition of *instrumental interaction* [7] is the closest description the author has found that relates to our work with ixi software. The interaction instrument is a tool that interfaces the user with the object of interest. A scrollbar is an example of such instrument as it gives the user the ability to change the state/view of the document. A pen, brush or a selection tool in a graphics package is also a type of such instrument.

There are three design principles that define the methodology of instrumental interaction: *reification* - the process by which concepts are turned into objects; *polymorphism* - the property that enables a single command to be applicable to objects of different types; *reuse* - the storing of previous input or output for another use. When an ixi application combines all three design principles into a successful interface, we have what we could call a semiotic machine. The interface is multifunctional and can be used in a variety of different contexts.

4.3 The Terminology of ixi's semantics

As explained in earlier papers, [14][15] most of the ixi software applications are controllers that send and receive OSC (Open Sound Control) [23] information to sound engines written in other environments such as SuperCollider [12] or Pure Data [17]. We separate the interface from the sound engine in order to be able to reuse the control structures of the abstract interface in other contexts, for example allowing a sequencing interface to control parameters in synthesis if the user configures it so. These controllers are all made from a common ideology or an interaction model that we see as a semiotic system.

In our work with *ixi software*, the fundamental attention has been on the interaction design and not the interface design. The design of interface elements is often highly (but not exclusively) aesthetic and depending on taste, whereas the interaction design deals with the fundamental structure and ergonomic idea of the software. In the example of SpinDrum [14], for example, the wheels contain pedals controlling beats per cycle, the size of the wheel signifies the volume and the

[5] Post-structuralist thought has rightly pointed out how interpretations of the work change in different times and cultures, but the work itself doesn't change - only people's interpretation and reception of it.

[6] As we can derive from the Peircian semiotics, an interface object can be represented in various ways: *iconically* (where the representation is based on resemblance to an object), *indexically* (where the representation is influenced by an object) or *symbolically* (where the representation is based on convention).

colour accounts for which sound is attached to the object. Here the interaction design clearly affects the interface design (size, number of pedals, colour), but the shape of the pedals (whether a square, a circle or a triangle) is simply an aesthetic decision and of little general importance.

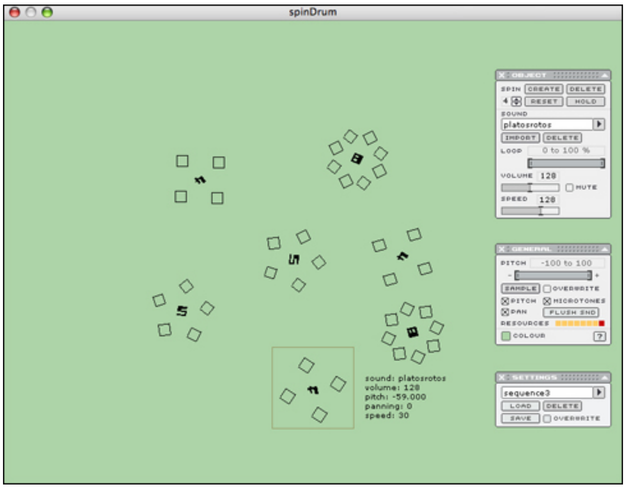


Figure 3: SpinDrum. Each wheel contains from 1 to 10 pedals. The wheels rotate in various speeds, and when a pedal hits top position (12 o'clock) it triggers the sample or sends out OSC info to the soundengine. The X and Y location of the wheels can affect parameters such as pitch and panning.

4.3.1 Actors

The ixi interfaces are pattern generating machines with cogs and bolts of varied significance. To sum up the basic design ideas of ixi software we could say that it was the *reification of musical ideas into abstract graphical objects as control mechanisms that act in time*.[7] We call these abstract objects *actors,*[8] as they are graphical representations of temporal processes that act, enact and react to the user, to each other or the system itself in a complex network of properties, relations and teleology (desired states or end goals). Beaudouin-Lafon calls graphical interface tools "interaction instruments", but we cannot use that metaphor as an ixi application is a musical instrument on it's own but also because of the different nature of the interface units of ixi software. The feature under discussion here is the difference musical applications have from the *ergonomically* "single-threaded" or serial task-processing applications used for painting, text editing, programming, video editing or in architecture. In contrast to these applications, a music application is multi-threaded or parallel, i.e. there are many processes, streams, layers or channels that run concurrently in every composition or performance, all controlled by the user, but, in the case of ixi, usually only one at a time.[9]-[10]

The interface units that we call actors - such as a picker, a spindrum or a virus - are not instruments that the musician uses for some task and then chooses another instrument for the next task. The actors in the ixi software applications are put into use at some point in time and they continue working in a temporal flow (rotating, moving through a trajectory or interacting) until the musician decides to stop or pause their activities.

4.3.2 Context

All actors perform their task in a context. They are graphically represented in a two- or three-dimensional space on the screen and their location might typically influence their properties. The actors move, rotate or blink in this space and are therefore both spatially and temporally active units. The space can have qualities such as temperature, gravity, brightness, etc. which are all qualities that could affect the actor's behaviour or it can contain other actors of different type that influence the behaviour of the message sending actors. Feedback from users of ixi software has shown us that people find the metaphor of an actor presented in time and space useful to represent musical actions and ideas. What the feedback also shows is that people intuitively understand the metaphor of having actors on a stage that perform some tasks that they – the directors of the piece – are controlling.

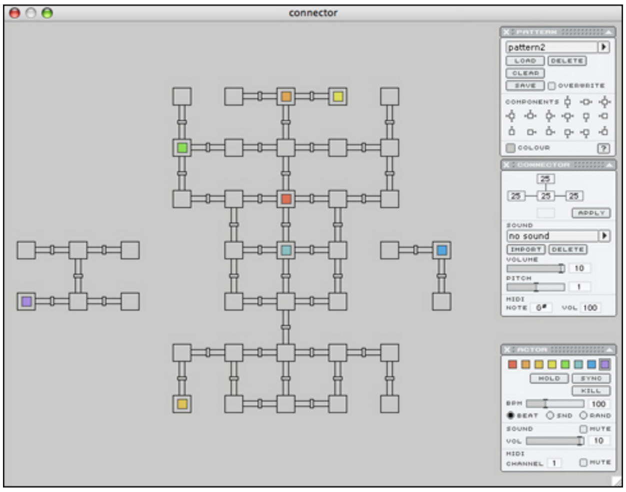


Figure 4: Connector. This software uses generative algorithms to decide where *actors* travel within a network of *connectors*. There are probability charts that decide the next move of an actor and when it enters a connector it triggers a MIDI note and/or a sound sample that is a property of the connector.

4.3.3 Network

When talking about the context and the environment of these actors, we must note the fact that the interface elements are not the only actors in the context of an ixi instrument: the user is one actor, the control hardware (a mouse, keyboard, sensor or controller), the soundcard, the speakers and other communication such as virtual audio cables, MIDI or OSC messages. The whole context of musical action and reaction is the space of the actor, a space in which the heterogeneous network of musical performance takes place. The meaning of the actor is its functionality within the control context and the mapping context. The actor has as many dimensions as it has numbers of control parameters and connections for receiving or sending messages.

---

[7] Musical idea here meaning any pattern generating structure.

[8] We thought about calling the active interface elements *agents* but it was too confusing as the term has very strong connotations in computer science, especially within the field of artificial intelligence.

[9] Another fact that divides those types of software is that the painting software, the video software or the 3D package are not packages that are used in live performance.

[10] This is of course what people are working with in the research field often known as NIME (New Interfaces for Musical Expression www.nime.org) where building physical interfaces to control sound on the computer allows for multi-parameter mapping to one sound-engine.

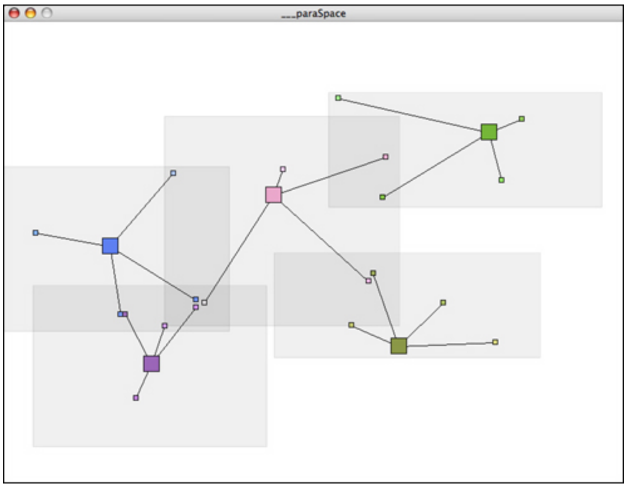


Figure 5: ParaSpace. This application interfaces with audio effects written in SuperCollider (but can talk to any software that supports OSC). Each audio effect has variable number of parameters and they are represented as small boxes in the control interface of ParaSpace. The point here is that the parameters interact on the interface level with automation, artificial life and artificial intelligence.

To clarify this idea of actors being all the elements that affect the interaction in an instrument, let us have a look at the software *Connector*. Here actors move in a system of connectors (a plumbing-like system) and trigger sound samples or MIDI notes that are properties of the connectors. The connectors are actors themselves as they are the receivers of an action and contain the information that yields the sound. It is through the interaction of all the actors and their properties that interaction takes place – interaction between elements *within* the instrument and also with the musician using the instrument – and this interaction is simply the automation that controls the various parts of the music set into motion. In *StockSynth* (Figure 1) the microphone is one such actor (with its properties of trajectory and scope) that interacts with the sound objects that contain the information about the sound and its properties.

4.3.4 Semiotic elements and mapping

The actors and the contexts in which they function are all elements in a semiotic language. This language has dialects or rather idiolects (each application is unique) where the meaning of an element can change as in Wittgenstein's concept of the usage as the word's meaning [15][22] or as in the Saussurian conception of the lack of natural connection between a signifier and the signified.[11] [19] We provide a semiotics or suggest language games where the behaviour of an actor maps onto some parameters in a sound engine. For example, vertical location of an actor could signify the pitch of a tone or playback rate of a sample. Size could mean amplitude, rotation triggering, and direction could mean a tendency for some action. But, it could also signify something entirely different as the controllers are open and it is up to the musician to map the actor's behaviour onto a parameter in the sound engine.

5. CONCLUSION

This paper has tried to show how the materials we work with when we design instruments (digital or acoustic) are the foundation for what can be expressed with the instrument. Whereas the expressive possibilities of an acoustic instrument are highly dependent upon the physical material it is built out of (wood, iron, strings, etc.), the situation is very different when we create digital instruments, especially screen-based. We have shown some examples of the semiotic system we are working towards in our work with ixi software and suggested a terminology of actors, context and network to better understand and modularise the interaction and interface design of virtual instruments. We have also illustrated how an interface can have its own meaning system independent of its relationship to the sound-engine, where the interactive patterns of an instrument can be mapped in many different ways onto the parameters of the sound-engine.

6. FUTURE WORK

Future plans involve exploring the dimensional spaces of the screen-based actors as the interface for musical interaction. The computer is becoming quite good at imitating the properties of acoustic instruments but it excels as an interesting instrument on its own where interaction is designed from the premise of the qualities of the computer and not by imitation of real world objects.

Our work involves experimenting in creating semiotic systems that can be taken further and extended into new dialects and systems of meaning. This system is not exclusive to one type of applications, but can rather be seen as a semiotic toolbox from which elements can be taken and reused in new contexts. Computer music software is a highly interesting area in the field of HCI as it is used in live performances and should contain depth that can be explored and practiced, thus allowing for musical virtuosity. In semiotic interfaces such as the ixi software there is always the filament of concurrent mappings or parallel streams of musical events happening at any one time. The temporal aspect of computer music software makes it also quite unique in relation to other types of software. Facing these incredible demands and challenges of music software we feel that we are just starting our journey into the possibilities of new meaning systems, metaphors, pattern generators and control of synthesis techniques through the creation of semiotic machines in the form of interfaces.

7. ACKNOWLEDGEMENTS

The *ixi software* project (www.ixi-software.net) is a collaboration between Enrike Hurtado Mendieta, Thor Magnusson and various musicians and artists that are involved with our work. I would like to thank Chris Thornton of University of Sussex, Marcelo Wanderley and the people at the IDMIL Lab at McGill University, Montreal and Julian Rohrhuber at the University of Cologne for good discussions and important feedback for this paper.

8. REFERENCES

[1] Andersen, Peter, B. & May, Michael. "Instrument Semiotics" in *Information, organisation and technology. Studies in organisational semiotics*. (eds. Liu, Kecheng; Clarke, Rodney J.; Andersen, Peter B.; Stamper, Ronald K.). Kluwer: Boston/Dordrecht/London. 2001: 271-298.

[2] Andersen, Peter, B. "What semiotics can and cannot do for HCI" in *Knowledge Based Systems*. Elsevier: 2001.

[3] Andersen, Peter, B. "Computer semiotics" in *Scandinavian Journal of Information systems*. Vol. 4: 3-30. Copenhagen, 1992.

[11] For Saussure, the meaning of signs is relational rather than referential, i.e. the meaning lies in their systematic relation to other signs in the semiotic system and not by direct reference to material things, for example.

[4] Apple Computer Inc. *Human Interface Guidelines: The Apple Desktop Interface*. Boston: Addison-Wesley, 1987.

[5] Barthes, Roland. *Mythologies*. London: Paladin, 1972.

[6] Beaudouin-Lafon, Michel. "Designing Interaction, not Interfaces" in *AVI Proceedings*, ACM: Gallipoli, 2004

[7] Beaudouin-Lafon, Michel. "Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces" in *Proceedings of ACM Human Factors in Computing systems (CHI 2000)*, The Hague: ACM Press, 2000.

[8] Beaudouin-Lafon, Michel. "Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces" in in *AVI Proceedings*, ACM: Palermo, 2000

[9] Eco, Umberto. *A Theory of Semiotics*. London: Macmillan, 1976.

[10] Edens, Aden. *Sound Ideas: Music, Machines, and Experience*. Minneapolis: University of Minnesoda Press, 2005.

[11] Gohlke, Gerrit. (ed.) *Software Art - A Reportage about Source Code*. Berlin: The Media Arts Lab, 2003.

[12] McCartney, James. "A Few Quick Notes on Opportunities and Pitfalls of the Application of Computers in Art and Music" in *Ars Electronica*, 2003.

[13] McCartney, James. "Rethinking the Computer Music Language: SuperCollider" in *Computer Music Journal*, 26:4, pp. 61–68, Winter, 2002.

[14] Magnusson, Thor. "ixi software: The Interface as Instrument" in *Proceedings of NIME 2005*. Vancouver: University of British Columbia, 2005.

[15] Magnusson, Thor. "ixi software: Open Controllers for Open Source Software" in *Proceedings of ICMC 2005*. Barcelona: University of Pompeu Fabra, 2005.

[16] Magnusson, Thor. "Processor Art: Currents in the Process Orientated Works of Generative and Software Art", Copenhagen: 2002. www.ixi-software.net/thor

[17] Nadin, Mihai. "Interface Design: A Semiotic Paradigm" in *Semiotica* 69-3/4. Amsterdam: Mouton De Groyer, 1988.

[18] Puckette, Miller. "Pure Data." in *Proceedings of International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 269-272, 1996.

[19] Saussure, Ferdinand de. *Course in General Linguistics*. London: Duckworth, 1983. p. 121.

[20] Shneiderman, Ben. "Direct manipulation: a step beyond programming languages," *IEEE Computer* 16(8) (August 1983), 57-69

[21] Wittgenstein, Ludwig. *The Blue And Brown Books*, Oxford: Blackwell, 1993. p. 6.

[22] Wittgenstein, Ludwig. *Philosophical Investigations*, Oxford: Blackwell, 1994.

[23] Wright, M; Freed, A; Lee, A; Madden, A. and Momeni, A. "Open Sound Control: State of the Art 2003" in *Proceedings of NIME 2003*. Montreal, Canada. 2003.

Different Strokes: a Prototype Software System for Laptop Performance and Improvisation

Mark Zadel and Gary Scavone
Music Technology Area
Schulich School of Music, McGill University
Montreal, Quebec, Canada
{zadel,gary}@music.mcgill.ca

ABSTRACT

This paper presents progress in the design of a new software interface for laptop performance and improvisation. These performances can lack a sense of active creation, as well as a visual connection between the performer's actions and the audio output. This stems partly from certain patterns in laptop performance in which musicians resort to heavy automation to cope with performing complex compositions. The software presented here attempts to address this by requiring that the user create all of the control sequences on-stage. The user defines graphical control patterns that are mapped to sample playback. The current prototype resembles a freehand drawing interface where the strokes create looping and cascading animations that generate corresponding audio, ultimately creating music. This style of interface minimizes the use of prepared material and takes advantage of the computer's unique capabilities.

Keywords

Software control of computer music, laptop performance, graphical interfaces, freehand input, dynamic simulation

1. INTRODUCTION

Laptop performance—musical performance on a standard computer system without novel controllers, usually by a solo artist—is an increasingly common mode of live computer music. The novelty of laptop performance is starting to fade, however, and it is sometimes criticized by audiences as being uninteresting. Laptop performances often foster little sense of the effort, difficulty, or activity audiences typically expect. A disconnect exists between the ostensible producer of the music and the music itself: there is no visible causal link apparent between the performer's gestures and the resulting audio[14]. Cascone notes that the same issues facing live acousmatic music exist in contemporary laptop performance[2]; these are essentially transferred from live tape music.

Though the live acrobatics of laptop performers can be intricate and complex, it is more often the case that musicians exercise limited control over a piece. In these instances, performance is reduced to triggering events, playing prepared control sequences, calling up presets, and perturbing scalar parameters. The performance of complex, layered pieces is difficult for a single performer, and the overwhelming number of variables are cognitively impossible to manipulate at once. Performers often have to resort to preparing automated control sequences and structures before going on-stage to cope with the complexity of a piece. This allows performance via a manageable fraction of the system parameters, but comes at the cost of driving fixed processes with a small set of controls[3]. Further, performance software interfaces are typically organized as dense on-screen control panels, featuring independent scalar widgets that are manipulated individually via the mouse[8]. This arrangement constricts the control flow between the musician and the computer system, additionally compromising live control.

"Preparation" as used here has a different connotation than it might in acoustic music. Jazz musicians, for example, practice various patterns and scales intensively for use in improvisation. A control sequence programmed into a computer can be played back exactly, without effort. Well-prepared human performance is interesting due to its inherent difficulty and natural variability, while prepared computer playback is not because of its ease and mechanical consistency.

This research attempts to address some of these issues in laptop performance by offering an alternative to typical performance software designs. The software prohibits the use of predefined control sequences, and aims to allow expressive performance and improvisatory use. It features an interface paradigm that focuses on a visual, spatiotemporal representation. This representation helps establish a link between performer action and musical output, hoping to address laptop performance's lack of visible, causal gesture. Freehand input via a graphic tablet or mouse is used to drive the software. Ultimately, the software aims to allow for an interesting live experience by returning a sense of active creation to laptop performance.

This paper presents this software system in comparison with various other applications for computer music performance. The design of the system is presented and discussed.

2. RELATED WORK

The most widely used examples of laptop performance software can encourage largely automated live use. These typically feature control-panel interface designs, which can constrict live control. The most popular example of contemporary performance software is *Ableton Live*[1], which allows a musician to layer and process audio clips in real-time. Music is performed by starting and stopping groups of loops, triggering clips, and modifying effects parameters using the on-screen controls. Though some clip rearrangement can be done in real-time, it is expected that the clips and effects setups are arranged before the performance. *Pure Data* and *Max/MSP* are the canonical examples of dataflow music software[12], and are also very popular for performance. Again, the vast majority of the work is usually done before the performance in preparing the patch. The patch is typically controlled live by using the on-screen widgets.

Less mainstream applications exist that de-emphasize the use of prepared material and that make use of dynamic graphics, visual feedback and creative interface design to enhance interaction and live use. Examples include the works of *ixi software*[9] and the *FMOL*[7] application.

Interfaces for performance which use freehand input also exist. These make use of a more spatial interface metaphor. Golan Levin's work on painterly interfaces[8] features inspiring examples of instruments made possible by digital technology that encourage improvisatory use. The two examples most relevant here are Levin's *Yellowtail* and *Loom* applications, as they make use of freehand drawing gestures in a manner similar to our research. These applications allow the user to draw strokes on the screen using a pen or a mouse. The drawing gesture is recorded and replayed repeatedly to animate the on-screen strokes, and the graphical output is sonified to generate audio output. Amit Pitaru's *Sonic Wire Sculptor*[11] creates rotating, three-dimensional "sculptures" from freehand input trajectories. The shapes are mapped to sound, allowing a performer to interactively create looping, multi-voiced music.

Other musical interfaces also employ a spatial metaphor. Toshio Iwai's *Electroplankton* game for NintendoDS features autonomous agents that create music[6]. These are controlled by the user, and interact with each other and their environment. Each has individual properties and sounds. This example is similar to this research in that the user sets active, spatial structures in motion to generate musical patterns.

Live coding[4][15] is an innovative laptop performance technique where artists play pieces through live computer programming. Our software prototype shares two characteristics with live coding techniques: the desire to build pieces from minimal starting material in performance, and the creation of music via concurrent, generative patterns. Textual programming allows great flexibility and fine granularity, but takes a substantial amount of effort and technical proficiency. Our research also aims to allow the live creation and modification of control patterns, but at a much higher level using graphical tools. The hope is that our approach will be a good compromise between control and efficiency.

3. PROTOTYPE SYSTEM

A prototype system was designed and implemented which attempts to address some of the above issues and encourage active, engaging laptop performance. Instead of preparing note sequences and control mechanisms before going on-stage, the system requires that the musician create these mechanisms *as* performance. The goal was to foster a more interesting experience for the user, as well as the audience, and to allow greater performance flexibility.

The system was intended to permit independent control structures to be assembled quickly and efficiently. The system targets dance and minimal "glitch" styles, which are typically structured as layers of looping audio patterns. A musician is able to generate the same effect by creating multiple simultaneous control patterns in performance.

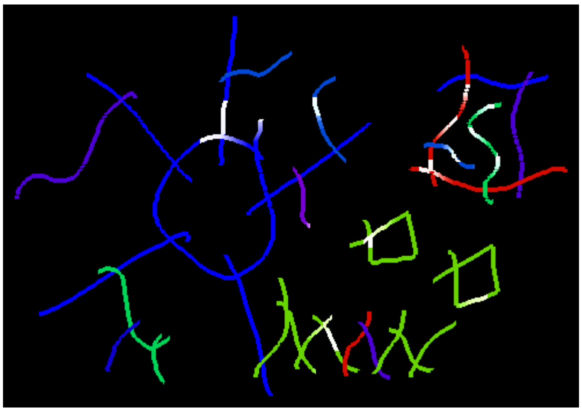


Figure 1: The interface in action

Figure 2: Particle behaviour at points of intersection. The colours have been altered to improve print reproduction.

The prototype system resembles a freehand drawing application, depicted in Figure 1. The program window represents a two-dimensional space in which the user can draw strokes. The strokes define paths along which small, white "particles" may travel. These particles are the active elements of the system; their movements and positions drive the sound playback.

Particles have a special behaviour at points where two strokes intersect. When a moving particle arrives at an intersection point, it is copied to the other stroke, and two particles travel outward from the intersection. This is illustrated in Figure 2.

These simple ideas—particles travelling along strokes that replicate at the intersections—give rise to the system's behaviour. Sets of overlapping strokes thus create figures through which the particles travel and cycle. In turn, these patterns of motion drive audio synthesis, ultimately generating music.

Each drawing gesture made by the user is recorded by the system. Travelling particles mimic these motions as they propagate along the strokes, preserving the gestures' speed and variances. This means that particles always travel along a stroke in the same direction as its original drawing gesture.

Any number of particles may travel along a given stroke at the same time. A particle always follows the user's cursor when drawing, causing new particles to be spawned immediately as intersections are first created.

The motions of these particles govern the sound synthesis via a simple mapping. A stroke can be associated with

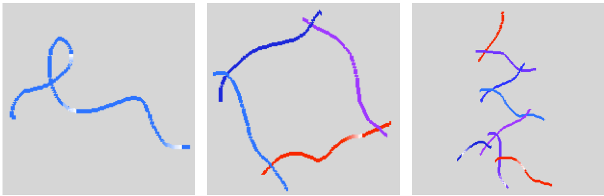


Figure 3: Some typical figures used in performance

a wavetable via keyboard shortcuts. The sound sample is imagined to be "stretched" along the length of the stroke, and a given particle moving along the stroke drives the motion of a corresponding playhead that reads sample data from the wavetable. Thus, faster drawing motions result in higher pitches, and slower ones result in lower pitches. Each stroke is allocated one synthesis voice, and the particle most recently added to the stroke is the one used to drive the mapping. Since there is no limit to the number of strokes, there is no limit to the number of simultaneous synthesis voices.

As mentioned above, wavetables are associated with strokes via keyboard shortcuts. The wavetable is selected from a small, predefined sample set, and subsequently drawn strokes will be associated with that sample until another one is chosen. Strokes are coloured according to their associated wavetable. Silent strokes, with no associated wavetable, can also be selected for control purposes.

Strokes remain stationary after being drawn. The only editing operation currently supported is deletion. A voice may be silenced either by deleting an upstream part of the figure to stop the flow of particles onto the sounding stroke, or by deleting the stroke itself. A stroke's location on the canvas has no effect on the animation or the sound. A figure will behave exactly the same regardless of where it is drawn.

The system can be used with either a graphic tablet or a mouse. One of the key advantages to this system is that it uses freehand drawing gestures, which are known to be very natural[16] and which exhibit interesting, human variability.

There are a number of typical figures that result from this system. The simplest is a stroke with a single loop. A particle orbits around the loop, and a new particle is emitted on each revolution. Loops can also be made using multiple strokes. Not all figures need be loops, and cascading sequences of strokes may also be drawn. These are illustrated in Figure 3.

The system is a self-contained, real-time, graphical application implemented in C++. It uses GLUT[5] for graphics, STK[13] for synthesis, and RtAudio[13] for managing the audio hardware. The code is object-oriented, features design patterns, and uses the C++ standard template library. Further implementation details can be found in [17].

4. DISCUSSION

The prototype system is a first step toward an alternative performance interface that attempts to bring an increased sense of active creation to laptop performance. It aims to address some of the challenges we detail above.

The system avoids a control-panel-style layout in favour of a graphical, spatio-temporal representation. This representation could help visualize and manage the large amount of activity present in complex, multi-layered works. Further, it helps build a connection between the performer's actions, the state of the software system, and the audio output. If projected, the screen display could help to demystify the performance mechanics for the audience.

With this system, we are returning to the acoustic musician's notion of preparation: practicing physical action on an instrument for performance without automatic aids. This is reflected in the fact that there is no mechanism for saving the screen layout or storing predefined control structures. This forces musicians to approach software performance as they might approach acoustic performance: through practice, but not programming. This philosophy also reinforces the interface's usefulness in improvisation.

The use of freehand gestures is very significant to the interface design. Freehand input with a graphic tablet gives the performer a natural, high-bandwidth way to interact with the system that infuses the control data with human variability. This variability can give the music an organic quality that is often absent from computer music. A given piece will sound slightly different each time it is performed, and it is possible to make mistakes. The use of freehand input helps to recapture some of these natural, interesting characteristics of human performance.

The system focuses on the live construction of generative structures for creating musical patterns. This is similar to the live coding approach. This aspect of our research targets styles of computer music where pieces are composed as layers of looping audio patterns. The system hopes to allow these structures to be created quickly and efficiently through graphical means, serving as an intuitive, geometrical way of specifying musical material.

At root, our project aims to be a "spatial language" for quickly specifying audio patterns. The intention was to devise a graphical system with atoms, rules and grammar that could be used for creating music. These visual components would be assembled on-stage by performers to achieve specific effects and patterns. The system's grammar is currently limited, but it will be further developed.

An important point is that the system is intended primarily as an interface for audio performance. The visual component is not necessarily meant to be expressive. In projecting the graphics, we make visible the performer's actions in the same way that one can watch a guitarist's hands as he or she plays. The system is more visually self-explanatory than typical live computer music software, which could help elucidate the performance for the audience. If the graphics are not projected, however, the system's advantages for the performer are still valuable.

There are some outstanding issues present in this early prototype. The current stroke behaviours make it difficult to synchronize between two running patterns. Synchronization could be a useful musical tool to have available. The editing capabilities of the system are currently quite limited, and only support stroke deletion. Some users have expressed that it would be useful to include more subtle editing operations. There is also a problem with the current behaviours for certain stroke topologies. A topology with one loop works well, but topologies with two or more loops constitute a feedback condition. This situation can be encountered unexpectedly in the course of a performance. These challenges will be addressed in future versions of the software.

This project bears similarity to Golan Levin's work in its use of freehand drawing gestures and animated stroke trajectories. While his research provided inspiration for our work, the overall goals of the two projects differ. Levin's work focused on audiovisual performance, the simultaneous creation of abstract visuals and sound. As mentioned above, our project is designed to be a "spatial language" for *audio* performance, allowing one to quickly specify temporal patterns. Levin's focus was on the aesthetic of the interaction, while this research focuses on the interface design issues and the effectiveness of the tool for computer music performance.

Another work to which this research bears some resemblance is Iannis Xenakis's UPIC system[10]. In UPIC, a composer uses a graphic tablet to specify two-dimensional curves that drive various sequencing and synthesis functionality. Our work differs in that we focus on real-time use, whereas UPIC is strictly an offline system. We also have a much narrower scope than UPIC, allowing a particular kind of performance within particular constraints. Xenakis's system is a broadly applicable, general-purpose environment for computer music composition.

An final point to make is that the interface design decisions are idiosyncratic and reflect the opinions of the designers. They were deemed interesting for this iteration of the project, but will be developed further. The current version of the software articulates the flavour of this style of interface and illustrates some of the possibilities offered by our approach. Other stroke behaviours and mappings to audio will be experimented with within the simulation framework, and the system will certainly evolve in the future.

5. CONCLUSION AND FUTURE WORK

This paper has presented research into a software system for computer music performance. Laptop performance is becoming increasingly popular, but audiences can be unsatisfied with the apparent lack of activity and lack of visual cues it sometimes offers. The prototype software system takes the first steps toward a new performance interface design that aims to address these issues. It is a direct-manipulation, graphical interface resembling a freehand drawing program that forces control mechanisms to be assembled live as performance. The system eliminates the use of predefined control sequences and allows improvisation. The variability inherent in the freehand input helps to recapture some of the "humanness" of acoustic performance. This scheme highlights one way in which musicians can exploit the unique affordances of software systems in a performance context.

The prototype system will be further developed. Usability details will be addressed: the system's editing functionality will be extended, menus and other interface items will be added, and the overall graphical feedback will be improved. MIDI or OSC support could also be added for controlling external synthesis software instead of using the application's internal synthesizer.

Most important, the overall system design will be developed. While preserving the core principles and feel of the interface, extended stroke behaviour designs will be found that enhance the system's performance capabilities. The ultimate goal is to have a conceptually minimal system that allows rich, robust and subtle performance.

6. ACKNOWLEDGMENTS

Thanks to Elliot Sinyor for suggesting "Different Strokes" as an entertaining working title for this project.

7. REFERENCES

[1] Ableton homepage. `http://www.ableton.com/`.

[2] K. Cascone. Grain, sequence, system: Three levels of reception in the performance of laptop music. *Contemporary Music Review*, 22(4):101–104, 2003.

[3] N. Collins. Generative music and laptop performance. *Contemporary Music Review*, 22(4):67–79, 2003.

[4] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*, 8(3):321–330, 2003.

[5] GLUT. `http://www.opengl.org/resources/libraries/glut/`, 2006.

[6] T. Iwai. Electroplankton. `http://electroplankton.nintendods.com/`, 2005.

[7] S. Jordà. FMOL: Toward user-friendly, sophisticated new musical instruments. *Computer Music Journal*, 26(3):23–39, 2002.

[8] G. Levin. Painterly interfaces for audiovisual performance. Master's thesis, Massachusetts Institute of Technology, 2000.

[9] T. Magnusson. ixi software: The interface as instrument. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pp. 212–215, 2005.

[10] G. Marino, M.-H. Serra, and J.-M. Raczinski. The UPIC system: Origins and innovations. *Perspectives of New Music*, 31(1):258–269, 1993.

[11] A. Pitaru. Sonic wire sculptor. `http://www.pitaru.com/sws/`, 2003.

[12] M. Puckette. Max at seventeen. *Computer Music Journal*, 26(4):31–43, 2002.

[13] G. Scavone and P. Cook. RtMidi, RtAudio, and a synthesis toolkit (STK) update. *Proceedings of the International Computer Music Conference*, pp. 327–330, 2005.

[14] A. Tanaka. Music performance practice on sensor-based instruments. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pp. 389–405. IRCAM – Centre Pompidou, 2000.

[15] G. Wang and P. R. Cook. On-the-fly programming: Using code as an expressive musical instrument. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pp. 138–143, 2004.

[16] M. Wright, D. Wessel, and A. Freed. New musical control structures from standard gestural controllers. In *Proceedings of the International Computer Music Conference*, pp. 387–389, 1997.

[17] M. Zadel. A software system for laptop performance and improvisation. Master's thesis, McGill University, 2006.

TENORI-ON

Yu Nishibori
YAMAHA Center for Advanced Sound Technology
Music & Human Interface Group
203 Matsunokijima Iwata Shizuoka Pref.
YAMAHA #12/4 CAST MHG
TEL : +81-539-62-6448
E-mail : nishibori@beat.yamaha.co.jp

Toshio Iwai
Media Artist
E-mail : iwai@gol.com

ABSTRACT
Development of a musical interface which allows people to play music intuitively and create music visibly.

Keywords
Human Interface, Software

1. INTRODUCTION
Media artist Toshio Iwai and YAMAHA have collaborated to develop a new digital musical instrument for the 21st century, TENORI-ON (See Figure 1). A 16x16 matrix of LED switches allows everyone to play music intuitively, creating a “visible music” interface [1].

Figure 1. TENORI-ON

In the past, there were some musical interfaces with a new point of view to music, which allow everyone to play music intuitively and visibly like Audio Pad [2] and Block Jam [3], but the new point of TENORI-ON is the interface with the inevitable design that makes you understand the musical structure visibly and the high quality feeling made by YAMAHA that has been manufacturing various kinds of traditional musical instruments for 120 years [4].

The interface of TENORI-N is completely unique and consists of a 16x16 matrix of light emitting switches. These switches, however, are not mere input switches, like the keys on a keyboard, but function as individual displays that emit light that emulates intuitively the related sound. When you push a switch for a short time, a ripple of light spreads out from switch that corresponds with sound you have chosen. If a switch is held down slightly longer, a dot of light remains on the panel indicating that the light and the sound will be played repeatedly. One remarkable effect of this sound and light synergy is that people seem to quickly understand the relationship between the sounds and switches such that even non-musicians can enjoy improvising and even composing almost immediately.

2. EXPOSITION

2.1 System
This interface is an embedded system operated by the real-time OS called ITRON that mainly controls the sound output, LEDs and the 256 switches. The core technology making this interface unique is the algorithm we developed to make music visibly with the 256 switches including LEDs inside.

Below are the main technical functions of TENORI-ON.

1. Analyze how long a switch among 256 switches has been touched
2. Analyze the direction of the finger touching the switches continuously
3. Analyze the angle of the interface with the acceleration sensor embedded in the interface

Those elaborate and highly developed technologies make it possible to give us a lot of information simultaneously, which we can't get with our eyes, and we can expand the possibility of our expression by controlling the sound and the light with those information.

2.2 Interface

2.2.1 Features Of TENORI-ON
Below are the main features of TENORI-ON.

1. Interface that anybody can play easily

   When you push the switch, the light is emitted and you’ll have the points to generate the sound.

2. Create Music with the various kinds of loops

   ① : When the scan bar moves one by one from left to right, it generates the sound at the point where the switches were pushed and loops.

② : Making some figures like triangle and rectangular by pushing switches, the light runs on the line of the figure and when it gets to the vertex, it generates the sound.

And TENORI-ON has more loop modes besides 2 examples mentioned above.

3. Session with another machine

When you connect your machine to another machine, it synchronizes with another one and it enables you to play the session with another person. It will be another interest that you can never experience when you play by yourself.

2.2.2 Construction Of The Interface

2.2.2.1 Surface

The upper frame has 2 speakers and 1 switch as the clear function between the speakers.

The left frame and the right frame have 5 function switches on each side.

The lower frame has a dial, a LCD and 2 function switches. And the base side has 2 terminals for a headphone and a volume.

The inside of the square frame has a 16x16 matrix of light emitting switches. (See Figure 2)

2.2.2.2 Backside

The backside of TENORI-ON controller has also a 16x16 matrix of light emitting button that is a dummy button not to be pushed.

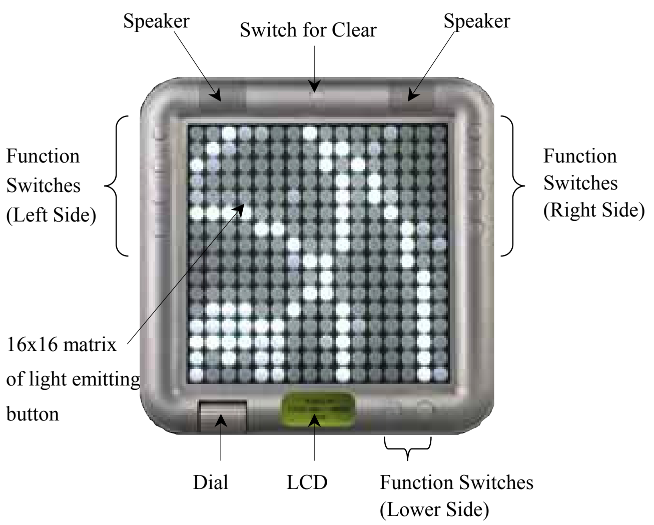


Figure 2. Interface of TENORI-ON (Surface)

2.3 How To Play

2.3.1 Several Modes To Play

TENORI-ON has several modes to play like the following 3 modes. Each mode should be selected from the menu written on the LCD.

- Play Mode : A mode to play TENORI-ON
- Remote Mode : A mode to control TENORI-ON by another MIDI instrument or PC
- Save / Load Mode : A mode to save and load files (examples : a data file of the performance, a file of the state (pattern) of the layer, the function parameter set file, the firmware file (load only) and so on)

2.3.2 Panel Operation

The performance of TENORI-ON should be done by pushing a 16x16 matrix of light emitting switch on the TENORI-ON controller. Depending on how long you push the switch, the reaction of the panel button should be different. The following description is about the basic actions after pushing the button.

- Pushing a switch for a short time : Single tone is generated （LED also emits light at a time）
- Pushing a switch for a long time: You can set the points to generate the sound in each loop mode（LED also continues to emit light with some varieties of animation）
- While you're pushing a switch is pushed : Just the same as Pushing a switch for a long time （If the sound is the single tone (not the continuous tone), the tone should be just generated repeatedly. ）

2.3.3 Master Function Set

The following descriptions are about the data that effect to all layers.

- Master tempo
- Master volume
- Layer selection
- Page selection
- Effect selection

2.3.4 Layer Function Set

The following descriptions are about the data that you can control on each layer.

- Positions of the points where the switches are pushed on each layer
- Loop Mode *
- Tone *
- Layer tempo *
- Volume *
- Sound length *
- Octave *
- Pan *

The data with * cannot be set plurally in the same layer.

2.3.5 Using As A MIDI Controller

You can use TENORI-ON is to control other MIDI instruments or PC by sending MIDI data as the operation data generated from the TENORI-ON controller through the MIDI OUT interface of TENORI-ON.

2.4 Software

2.4.1 Several Loop Modes

TENORI-ON has several loop modes. The following descriptions are about the main outlines of the loop modes in TENORI-ON.

By multiplying these different loop modes, TENORI-ON generates more complex and interesting music.

2.4.1.1 Score Mode

The horizontal axis means the time and there're 16 steps. And the vertical axis means the pitch and the 16 different pitches are assigned.

When the scan bar moves one by one from the left side to the right side and hits the point where the switches were pushed, it generates the sound at the point. Simultaneously the point where the switch was pushed emits the light strongly and the animation should be played from the point. When the scan bar reaches the right side on the panel (16th bar), it gets back to the left side and moves toward the right side again. In Score Mode, this action should be continued. (See Figure 3)

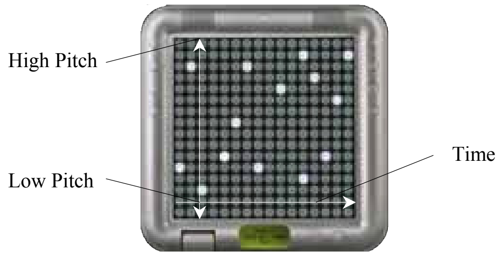


Figure 3. Basic Mode

2.4.1.2 Random Loop Mode

When you push the panel switch one by one, the point where the switch was pushed should be memorized as the order it was pushed.

The light moves between the points where the switches were pushed and the sound should be generated when the light hits the point. Then it moves back to the first point that you pushed at first after the light hits the last point that you pushed. Therefore you can make a loop by making various kinds of figures in this loop mode. (See Figure 4)

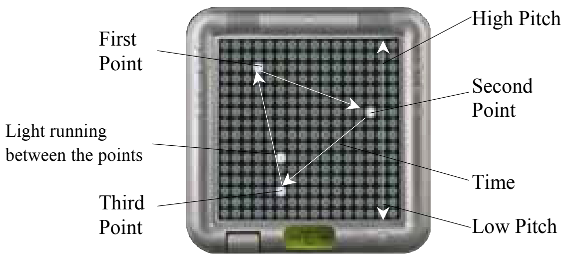


Figure 4. Random Loop Mode

2.4.1.3 Real Time Rec Mode

The data of both the time and the position of the switch that you pushed are memorized during a loop time of the Score Mode, and your last performance should be reappeared and looped as it's memorized. The memorized data of your performance should be repeated as the order that you operated 1 loop time before. (See Figure 5)

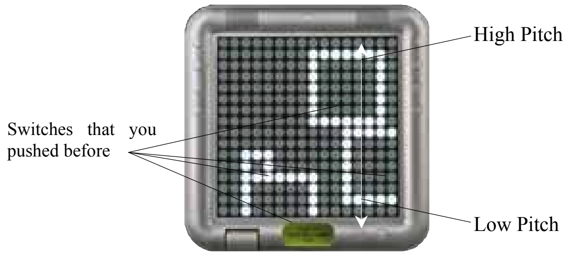


Figure 5. Real Time Rec Mode

2.4.1.4 Bounce Mode

When you push the panel switch, a dot of the light moves down to the bottom line of the panel switch. It generates the sound when the dot hits the bottom line, after that, the dot moves up to the point which you pushed at the first time. And when the dot gets to the point which you pushed at the first time, it moves down again. In this loop mode, it repeats this up and down movement.

In this mode, the horizontal axis means the pitch (the left side : low pitch, the right side : high pitch) and the vertical axis means the time. (See Figure 6)

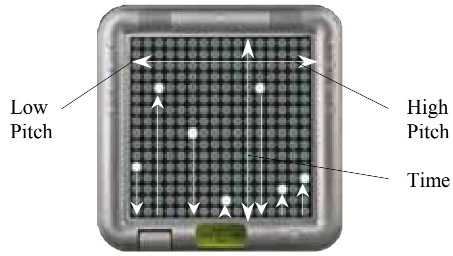


Figure 6. Bounce Mode

2.4.1.5 Push Mode

While the panel switch is pushed, the sound and the light are changed gradually.

After pushing the panel switch for a short time, it stops to generating the sound and emitting the light. And after pushing the panel switch for a long time, it continues to generate the sound and emit the light until the switch for the clear function is pushed. (See Figure 7)

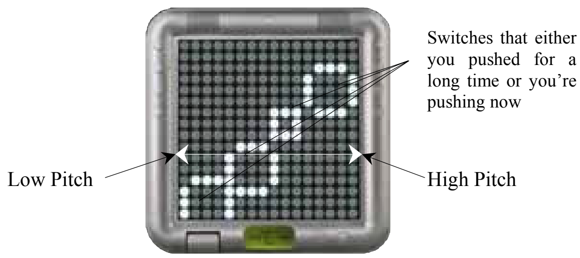


Figure 7. Push Mode

2.4.1.6 Solo Mode

The sound is generated continuously while the panel switch is pushed, and after holding the switch that you pushed, it stops generating the sound and to emitting the light. (See Figure 8)

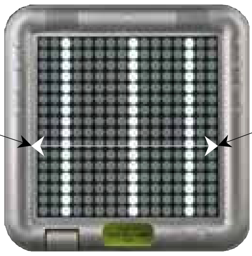


Figure 8. Solo Mode

2.4.2 Other Functions

2.4.2.1 Session With Another Machine

TENORI-ON can be connected to other TENORI-ONs with MIDI and play the session with other users.

In the session, a TENORI-ON as a master role sends the system real-time message among the other TENORI-ONs. The Other TENORI-ONs should get the message from the master TENORI-ON and synchronize the master's timing.

2.4.2.2 Save And Load The Data

2.4.2.2.1 Save and Load the data of the performance

TENORI-ON memorizes when and which panel switches were pushed from the start of the recording to the stop of the recording. The decision of the start and the stop of the recording should be done by the user's operation. The recorded file will be saved in the memory card like SD memory card.

Below are the main data to be saved.

- Positions of the points where the switches were pushed on each layer
- Time stamps of the points where the switches were pushed on each layer (The very first time stamps should be the beginning of the recording)
- Loop modes on each layer
- Master function set
- Layer function set
- Other sets on the menu

2.4.2.2.2 Save the state of a layer

You can save the data of the layer selected as a top layer.

Below are the main data to be saved.

- Positions of the points where the switches were pushed on the layer that's selected as a top layer
- Layer function set

CONCLUSION

2.5 Message From Toshio Iwai -- Musical Instrument For A New Age

I want to handle both light and sound simultaneously and pleasantly, as we play music or draw pictures. This is the theme I have been working on for a long time. Pursuing this idea further I have, in collaboration with YAMAHA, been developing TENORI-ON with particular attention being given to, the beauty of the light and sounds, the ease of performance, and as a musical instrument for the future, the design and the quality of the product as a whole. In days gone by, a musical instrument had to have a beauty, of shape as well as, of sound, and had to fit the player almost organically.(Instrument like the violin spring to mind.) All of these elements were once considered indispensable. Modern electronic instruments don't have this inevitable relationship between the shape, the sound, and the player. What I have done is to try to bring back these, once indispensable, elements and build them in to a true musical instrument for the digital age. TENORI-ON.

2.6 Our Goal

We developed this work with a new point of view to music. And we'd like to keep on thinking the musical structure and music with a new point of view. When we develop some work with the idea and the image gotten from this method and make the idea move, then you can catch a new image of music, which you've never seen before. And if you control the motion of the new image, the interface will be a brand new thing inevitably. Sometimes, you might have to develop some new basic technology at first to make it. In that case, you could develop a work that is exactly mixed art with science, and the possibility being a brand new work could be enhanced. We hope that people could get a new point of view to music and it could connect to the development and the expansion of the possibility of music by people using the interfaces like our work.

Finally we hope this work would make it possible for many people (including people who don't usually play any musical instruments) to experience the pleasure of music more than before. And when our work that has a new point of view to musical structure helps to expand the possibility of music and to develop music, we would like to think that it would be our goal.

3. ACKNOWLEDGMENTS

Our thanks to our friends and the musicians who gave us a great deal of valuable advice.

4. REFERENCES

[1] Toshio Iwai, Yu Nishibori. TENORI-ON. *SIGGRAPH 2005* (Aug. 2005)

[2] James Patten, Ben Recht, Hiroshi Ishii. Audio Pad. *Sonar 2003* (May. 2003)

[3] Henry Newton-Dunn, Hiroaki Nakano, James Gibson. Block Jam. *NIME03 (May. 2003)*

[4] The chronology of YAMAHA

http://www.yamaha.co.jp/pdf/cor/inf/his/cp01j.pdf

Towards a Gesture Description Interchange Format

Alexander R. Jensenius
Department of Musicology
University of Oslo
Pb 1017 Blindern
NO-0315 Oslo, Norway
a.r.jensenius@imv.uio.no

Tellef Kvifte
Department of Musicology
University of Oslo
Pb 1017 Blindern
NO-0315 Oslo, Norway
tellef.kvifte@imv.uio.no

Rolf Inge Godøy
Department of Musicology
University of Oslo
Pb 1017 Blindern
NO-0315 Oslo, Norway
r.i.godoy@imv.uio.no

ABSTRACT

This paper presents our need for a Gesture Description Interchange Format (GDIF) for storing, retrieving and sharing information about music-related gestures. Ideally, it should be possible to store all sorts of data from various commercial and custom made controllers, motion capture and computer vision systems, as well as results from different types of gesture analysis, in a coherent and consistent way. This would make it possible to use the information with different software, platforms and devices, and also allow for sharing data between research institutions. We present some of the data types that should be included, and discuss issues which need to be resolved.

Keywords

Gesture description, gesture analysis, standards

1. INTRODUCTION

Our current research evolves around relationships between music-related gestures (i.e. bodily movement) and musical sound. One of the approaches we have taken is to observe how people move in response to musical stimuli, what we call *sound-tracing*, *sound-sketching* or *sound-mimicking* gestures (e.g. air instrument playing [2]). As presented in more detail in [5], we see the need for a consistent way of describing music related gestures and gesture–sound relationships. In this paper we present our needs for a data format to store such information.

We typically use a number of different tools and methods in our observation studies, everything from motion capture systems, video recordings and sensor data to manual annotation. Up until now, all of these have been recorded in various programs on different platforms, and in all sorts of data formats. This is inefficient and also problematic when it comes to analyzing the material. It also leads to some challenges when it comes to synchronization, since the data is recorded with different time coding. The result is that we need to synchronize data manually, which is a very time consuming affair. Not only are these issues problematic when working within our own research group, but it also makes sharing data with other institutions difficult.

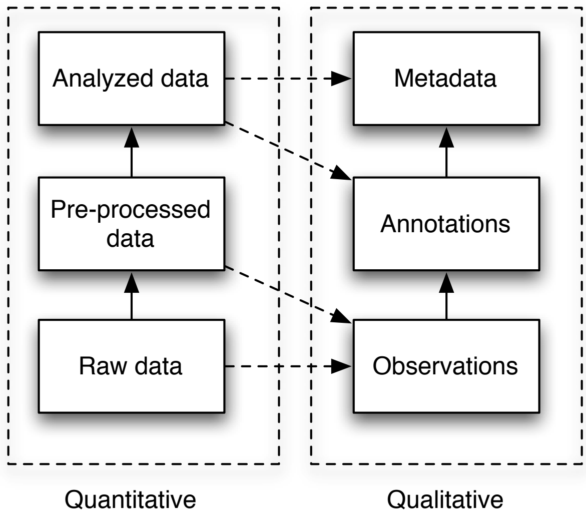


Figure 1: Different levels of data

From our experience, and after talking to colleagues from other institutions, there seems to be a need for an open standard for storing gesture-related data and analysis in a structured way. Inspired by the Sound Description Interchange Format (SDIF), developed by IRCAM and CNMAT in the late 1990s [8], and currently available in a number of audio programs and programming environments [7], we call for development of a Gesture Description Interchange Format (GDIF) to solve some of the above-mentioned issues. We believe it is important to try and use current standards as far as possible, so GDIF could possibly be implemented as an Open Sound Control (OSC) address space, which could allow for both realtime and nonrealtime applications.

This paper presents what we currently find important to include in a GDIF specification, some challenges that need to be addressed, and suggestions for how to proceed developing the standard.

2. STORING OBSERVATION DATA

An example of a typical workflow in one of our observation studies is sketched in Figure 1. The first stage is to record raw data from various sensors and devices, together with our manual observations and descriptions. This information is later pre-processed and analyzed, and forms the basis for annotations and higher level analysis. The following sections will present these different levels, and how it might be possible to store them in a consistent way.

2.1 Controller data

We often use different types of commercially available controllers for our studies. Some of these already have well-defined formats, such as MIDI-instruments, as opposed to for example HI devices[1] and Wacom tablets which typically differ for each device and model. Common for all such devices is that they have moderate sampling rates (around 100 Hz), and resolutions varying from 7-bit for MIDI-devices, 8-bit for most HI controllers and up to 16-bit for Wacom tablets.

As suggested in [9], data from a joystick could be represented with an OSC address space like:

```
/joystick/b xtilt ytilt rotation ...
```

when button 'b' is pressed. A problem with this approach, however, is that the end user will not necessarily know what the different values mean. Even though it is less efficient, we believe it could be wise to split the information into separate components. This would probably make them easier to read and parse for the end user. An example could be:

```
/joystick/button/b [1/0]
/joystick/x/position [0.-1.]
/joystick/x/rotation [0-360]
```

An important issue would be to define the units used. In general, we prefer to normalize values to a 0.-1. range, but for some data it could be more relevant to use other units. This could for example be defined after the value, like:

```
/joystick/x/rotation 270 degrees
```

Similarly, a MIDI device could be described as:

```
/keyboard/midi/[note, velocity, channel] [...]
```

And a Wacom tablet as:

```
/wacom/graphire/[x, y, pressure, xtilt, ...] [...]
```

In the last example, the manufacturer's name and device name is included. For a Wacom tablet this might be relevant information for the end user, while in other cases it might not. Rather than using such names, it could be more relevant to create names describing the quality of the device (e.g. tablet) rather than the brand name.

2.2 Motion capture data

Storing information from commercial motion capture systems should also be quite straight forward. One type of such systems (e.g. Vicon, Eagle 4) use infrared cameras to record the position of reflective markers in a space. The end result is a massive amount of x, y, z data (recorded at up to 4000 Hz) from up to 40 markers. Electromagnetic tracking systems (e.g. Polhemus Liberty/Patriot) typically have lower sampling rates (up to 240 Hz), but can also record orientation (azimuth, elevation, roll) of the markers.

Although most manufacturers of motion capture systems have their own custom built software, there seems to be some agreement on the C3D format[2] as a standard way of coding motion capture data. Since this format is so tied to motion capture systems, and only focuses on absolute positioning, it is not suitable for our needs. It would be better to also code such information into GDIF, for example like:

```
/polhemus/patriot/[x, y, z, azimuth,
                   elevation, roll] [...]
```

2.3 Sensor data

Commercially available controllers and motion capture systems usually have some standardized values that can easily be converted to GDIF messages, but it is more difficult when it comes to data from custom built controllers. They are typically built with all sorts of sensors and the output is also highly dependent on the sensor interface used. Popular interfaces such as iCubeX, Phidgets, Teabox and Kroonde work at all sorts of sampling rates (10 Hz to 4000 Hz), resolutions (7-bit to 16-bit) and ranges (e.g. 0.-1., 0-127, 0-65536), which make it difficult to come up with a general system for how to store the data. For such devices it might be more interesting to store preprocessed information directly, scaled and normalized to for example a 0.-1. range, and grouped according to sensor type, for example:

```
/sensor/accelerometer/[x y z] [...]
```

For custom built controllers it would probably be wise to store information in the header of the GDIF file about how it was built, which sensors and sensor interfaces were used, etc., so it can be possible reproduce and verify the results. The structure of such descriptions, and the level of detail needed, could probably be left open to the end user, but it would be good to provide some examples of good practice.

2.4 Other information

As well as the numerical information mentioned above, we typically also need to store general descriptions about the recording sessions, with information about date, researchers involved in the recording, the subjects, etc. Such information can be written to the header in a file, so that it is easy to get an overview of the data in the file by investigating the header.

Another important issue is to be able to store information about sound and video recordings made in parallel to controller and sensor data. We usually record video and audio in observation sessions, both as raw material for analysis, and as a reference. Working with DV-cameras, this has posed some serious challenges in terms of synchronizing the recordings with sensor data.

To overcome some of these issues, we have developed a set of tools[3] for recording audio, video and gesture data in one program. The tools are based on the Jamoma[4] modular standard for Max/MSP/Jitter, using OSC for all messaging. This allows for easily sending and receiving data from networked computers and devices, and makes it possible to synchronize all the data. In our current setup, all data is stored on one computer, which is a bottleneck when we need to record video from multiple cameras at the same time as recording high density data coming from a motion capture system. We are currently developing a system which can store synchronized information on several computers.

Since our studies often involve analysing movements in relation to prerecorded music, we need a way to store and synchronize original sound files, next to sound recorded during observation sessions. It would also be interesting to have a way of storing higher-level musical descriptors, such

[1] Human interface devices, e.g. game controllers.
[2] www.c3d.org
[3] Available from http://musicalgestures.uio.no
[4] http://www.jamoma.org

that it could be possible to retrieve structural elements in performances.

Finally, during observation sessions we typically also want to store some descriptions about new sections, interesting or unusual things happening, etc. Such invaluable information could come in any form, anything from time markers to text comments, and should also be time-stamped and stored alongside the other data.

3. STORING GESTURE ANALYSIS

Besides storing data, information and descriptions from the recording sessions, we are also interested in storing results from analysis of the data. Here the challenge is not so much on the data processing side, but more on the analytical, as we are still developing methods and tools for analysing musical gestures.

As discussed in more detail in [5], we find it important to differentiate between different *analytical perspectives* (Figure 2). For example, we find that it is a big difference whether the movement is observed with respect to the performer, the listener, or the controller/instrument. Analysing the movements of a pianist, there is a big difference whether the gestures are studied from the pianist's or from the audience's perspective. A performer usually starts to think about, and carry out, a gesture long before it is actually seen from an audience. For the performer, the *goal point* (e.g. key on the piano) of the movement is crucial, as well as the internal feeling of the movement, while for the average listener the larger shape of the gesture is probably more important, and probably also the only thing that is actually seen at a distance from the stage.

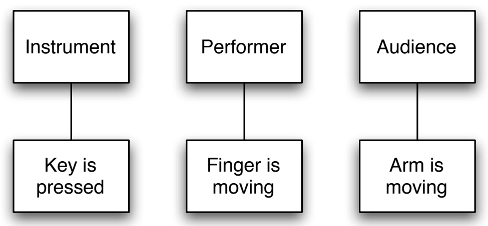


Figure 2: Three different analytical perspectives.

Another important thing to consider in the analysis, is which *attention level* or *analytical resolution* to use. As humans we tend to process information at several different levels simultaneously, and this is also important for how we process information. In terms of a piano performance, we may choose to focus on both small and large gestures at the same time. We also typically operate at several different time scales, looking at rapid attacks of hands and fingers, but also how the upper body is moving over time. The same should be the case in our analysis, so it is important that we have a tool where it is possible to have multiple analytical streams, and different analytical levels next to each other (Figure 3).

In the following sections we will present some of the different analytical methods we employ and how it could be possible to store the results in a GDIF standard.

3.1 Biomechanical analysis

Biomechanical analysis typically focuses on quantitative aspects of human motion, such as velocity and acceleration curves from various joints and limbs. For such analysis, the trick is to get good recordings, and then the analysis is a matter of calculating the relevant information. Since the information is numeric and easily labeled, such information is also easily stored.

We still need to figure out how to better represent the information, for example whether it should focus on body parts:

```
/arm/right/[velocity, acceleration, direction] [...]
```

or be grouped according to kinematic quality:

```
/velocity/arm/[right, left] [...]
```

What to choose often depends on the main focus of the analysis, and what is more practical when the information should be parsed and used in other systems.

3.2 Laban movement analysis

On a more qualitative side, we are interested in storing Laban movement analysis (LMA) focusing on observations of the inner qualities of movement [3]. LMA consists of the descriptors *body*, *effort*, *shape* and *space*, of which all can be broken down to separate elements. For example, the four effort elements are *weight*, *space*, *time* and *flow*, and each of these elements are defined in terms of the following axes:

Weight light — strong

Space direct — indirect

Time sustained — sudden

Flow free — bound

These axes may be a good starting point for creating a numerical way of storing Laban information, since each of the pairs could be defined by a 0.-1. range. Thus it would be possible to store subjective Laban data numerically by for example adjusting a slider between the two extremes when carrying out the analysis.

3.3 Other types of analysis

What we find particularly important in our analyses of gesture–sound relationships, is that of separating and comparing gestures to *musical objects* [6]. Segmentation of movements and sounds can be derived quantitatively, but we typically do segmentation manually, since it is also often several ways of doing this [2]. Again it is important to be able to store multiple layers of information, since segmentation can typically occur at different levels (as sketched in Figure 3).

We are also interested in trying to formalize a way of describing gesture primitives such as *trajectory*, *force* and *pattern*, described in [1], as well as organological parameters related to instrument control and performance, such as presented in [4].

4. TOWARDS GDIF

These are some of the most important criteria when it comes to development of GDIF:

Open so that everyone can use it for any gesture-related data storage.

Human-readable so that it is possible to easily understand what the data means.

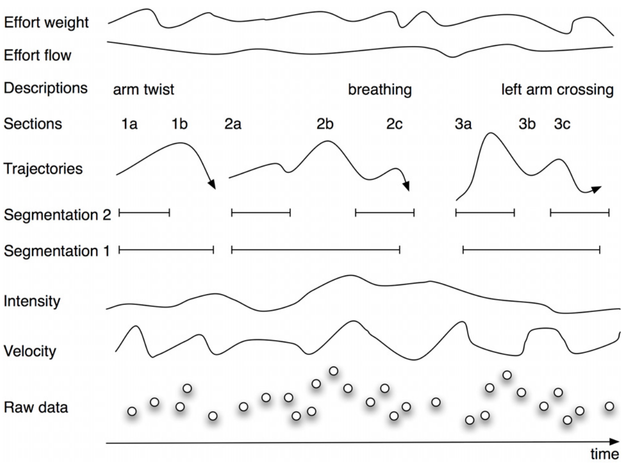


Figure 3: Sketch of gesture content that could be stored in a GDIF file.

Multiplatform so that it will work on any computer platform and software.

Flexible so that it can allow for multiple layers of analysis, dependent on the analytical level or different researcher's opinions.

Extendable so that it is possible to add more descriptors and content when it is necessary.

Simple so that it is easy to get started. This means that there should only be a very limited basic requirement for the data format, probably only a header with some descriptors, while everything else could be decided by the user. This will also make it easier to implement in different programs.

Efficiency is not the most important, as we believe well documented and easy to read codings are more valuable for our research. That said, it is always good trying to be as efficient as possible.

We have currently started to develop GDIF as an OSC address space, and storing the data streams with time tags, but it might be possible that an XML approach might be the way to go.

5. CONCLUSION

The paper has presented some of our current needs when it comes to storing and sharing information about music-related gestures. We suggest the development of a Gesture Description Interchange Format (GDIF), as an open and flexible format for storing gesture-related data and analysis. However, it should not be so open that it serves no purpose, so a number of data types and a standard way of describing certain devices and analytical methods should be required. Ideally, such a standard would allow for sharing information between different software, platforms and research institutions.

In addition to the challenges presented in the paper, we see a number of unresolved issues which will need to be addressed in the future development:

- What type of synchronization is better?
- What type of time-base should be used?
- What precision and resolution is necessary?
- Which audio and video format(s) and compression standard(s) should be used?
- How can chunks of information be represented at various levels?
- Should the information be stored progressively so it would be possible to change the attention level when analysing the material?

Currently, researchers at the University of Oslo and McGill University are involved in development of GDIF, and others are more than welcome to join. The next step is to create a draft specification, along with a number of more detailed example files and software, which can be presented at a relevant conference in the near future.

6. REFERENCES

[1] I. Choi. Gestural primitives and the context for computational processing in an interactive performance system. In M. M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 139–172. Paris: IRCAM - Centre Pompidou, 2000.

[2] R. I. Godøy, E. Haga, and A. R. Jensenius. Playing 'air instruments': Mimicry of sound-producing gestures by novices and experts. In S. Gibet, N. Courty, and J.-F. Kamp, editors, *Gesture in Human-Computer Interaction and Simulation: 6th International Gesture Workshop, GW 2005, Berder Island, France, May 18-20, 2005, Revised Selected Papers*, volume 3881/2006, pages 256–267. Springer-Verlag GmbH, 2006.

[3] P. Hackney. *Making Connections: Total Body Integration Through Bartenieff Fundamentals.* New York: Routledge, 2000.

[4] T. Kvifte. *Instruments and the Electronic Age. Towards a Terminology for a Unified Description of Playing Techniques.* Oslo: Solum Forlag, 1989.

[5] T. Kvifte and A. R. Jensenius. Towards a coherent terminology and model of instrument description and design. *NIME06, Paris*, Forthcoming.

[6] P. Schaeffer. *Traité des objets musicaux.* Editions du Seuil, 1966.

[7] D. Schwarz and M. Wright. Extensions and applications of the sdif sound description interchange format. In *Proceedings of the International Computer Music Conference, Berlin, Germany*, pages 481–484, 2000.

[8] M. Wright, A. Chaudhary, A. Freed, D. Wessel, X. Rodet, D. Virolle, R. Woehrmann, and X. Serra. New applications of the sound description interchange format. In *Proceedings of the International Computer Music Conference, Ann Arbor, Michigan*, pages 276–279, 1998.

[9] M. Wright, A. Freed, A. Lee, T. Madden, and A. Momeni. Managing complexity with explicit mapping of gestures to sound control with osc. In *roceedings of the 2001 International Computer Music Conference, Habana, Cuba*, pages 314–317, 2001.

SensorWiki.org: A Collaborative Resource for Researchers and Interface Designers

M. M. Wanderley, D. Birnbaum, J. Malloch, E. Sinyor, J. Boissinot
Input Devices and Music Interaction Laboratory
Schulich School of Music, McGill University
Montreal, Canada
http://www.music.mcgill.ca/musictech/idmil/

ABSTRACT

This paper describes an online *Wiki*, a collaborative Web site designed to allow users to edit and add content. It was created at the Input Devices and Music Interaction Laboratory with the aim of promoting and supporting the construction of new musical interfaces. Although many individual universities and research centres offer sources of relevant information online, this project allows for easy sharing and dissemination of information across institutional and international boundaries. In this paper, the internal framework and categorization scheme for the Wiki is profiled, and each section is introduced. The benefits of joining this effort are clearly demonstrated, and the possible future directions of the project are detailed.

Keywords

sensors, Wiki, collaborative website, open content

1. INTRODUCTION

A *Wiki*, a term derived from the Hawaiian for "quick", is a Web site configured to collect and distribute free information, by allowing site viewers to edit its content [18]. It is made up of two systems working together — a template layer which converts a simple markup language written by users to HTML documents, and a version control log that records the time and page on which each individual edit occurs. These two parallel subsystems facilitate non-destructive editing and help safeguard against vandalism [10]. By its nature, a Wiki makes possible many-to-many communication amongst contributers and users [4].

There are several Wiki software engines available, both proprietary and open source [13] [9] [12] [16] [6]. WikiMatrix.org [17], a Web forum for discussion of Wiki engines, lists 46 separate systems. The size of a Wiki is usually measured by article count, but several other options exist [7], which include:

- Total size of Wiki in bytes
- Total equivalent pages in A4 paper
- Total number and frequency table of words
- Number of articles in different byte size ranges, such as 50, 250, or 1000 bytes
- Size of articles N/L, 2N/L, ... LN/L where N is the total number of articles and L is the number of languages in which articles are written

The largest Wiki by all measurements is the *Wikipedia*, an online collaborative encyclopedia project [4]. It utilizes WikiMedia Foundation's GPL–licensed engine, Mediawiki [15], as does SensorWiki.org. While some Wikis require a short registration process for editors, as of writing the Sensor Wiki does not. [1] The advantage of Wikis as compared to traditional websites is that information can be quickly shared amongst all the interested members in a given field or community. Structured Wikis like SensorWiki.org attempt to combine this open nature with the format consistency and flexibility of a database application [14].

While some of the information compiled and developed by research labs is proprietary, there is an abundance of material that would be made public if the proper forum for its release were made available. The Sensor Wiki project is just such a forum.

2. EXISTING RESOURCES

There is much published research about sensors [11] [3]. Papers (for instance [1]) and more recently a book [8], address musical applications of sensors. Although these resources are useful because they have undergone a rigorous editing and review process, most of what is available stops short of providing specific data, such as information on sensor purchasing (where, how much), as this information changes often. This is also true for publications in journals and conference proceedings.

To find practical information, one must usually conduct online research to determine what is available for a given task, compare specifications and prices, and finally make contact with a company and place an order. But entering the word "sensor" into a search engine such as Google yields millions of superfluous results. The process is not only extremely time-consuming, it is often repeated unnecessarily because the information gathered each time is not organized and preserved.

The creation of a resource tool about sensors in music is an ideal application for a Wiki system for several reasons: It serves as a single place to gather resources and

[1] As there has not yet been excessive amounts of vandalism, a registration process has not proved necessary. If spam and editing by "bots" becomes an issue, this point will need to be reconsidered.

information, it allows and encourages members of different institutions to share their findings and discoveries, and finally it can be updated quickly and easily as new information becomes available. It is then complementary to other sources of information, such as articles and books.

3. SENSORWIKI.ORG

The Sensor Wiki is located at www.sensorwiki.org, and is currently organized into three sections:

- A comprehensive list of sensors, each with their own sensor description page.
- A database of references on interfaces and interaction.
- A section containing detailed tutorials related to sensor interface design.

3.1 Sensor list

Sensors in the list are organized in categories according to the physical phenomenon they sense, for instance, rotary position, linear position, or force/pressure/strain.

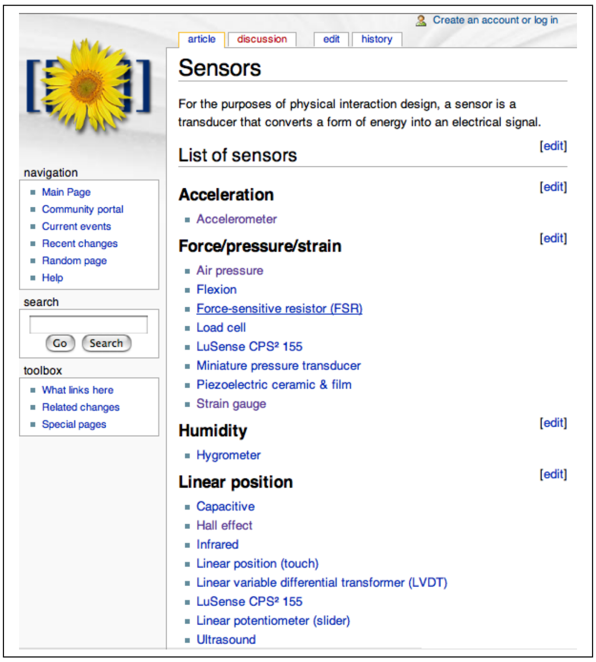


Figure 1: A screen shot of part of the sensor list in the Sensor Wiki.

If a sensor can be used to sense more than one phenomenon, it is included under each category for completeness. However, this is only a repetition of the link; each leads to the same sensor description page.

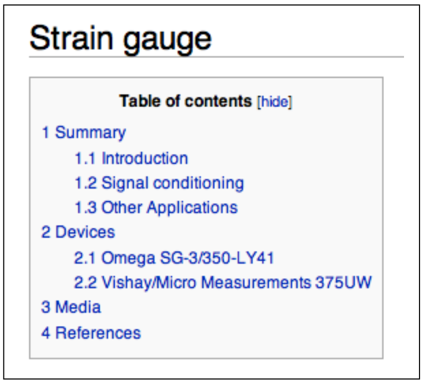


Figure 2: A typical list of topics covered for a particular sensor.

Each sensor description page includes:

- An introductory paragraph where background is provided and general issues about the sensor are discussed.
- A section describing the practical use of the sensor, including ways of constructing conditioning circuits, mounting techniques, and type of signal output.
- A list of companies that offer this sensor, including a data sheet, price list, and link to the company's site.
- Media featuring the sensor, such as images, video, circuit diagrams, and CircuitMaker [2] simulation files.
- External links and a reference list of resources used in the writing of the article.

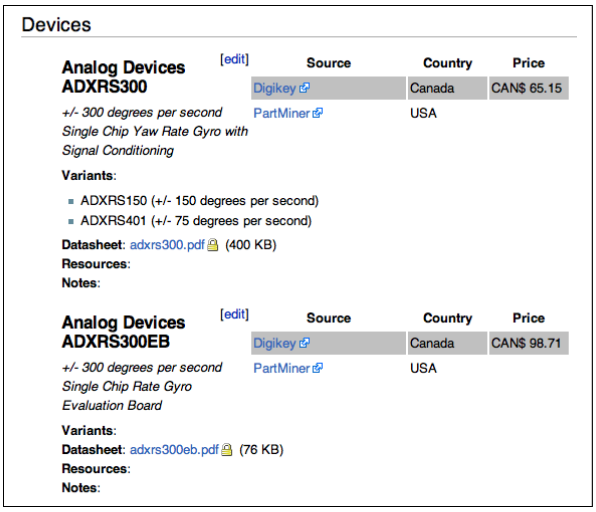


Figure 3: Example of a list of devices commercially available, where to find them and prices – gyroscope.

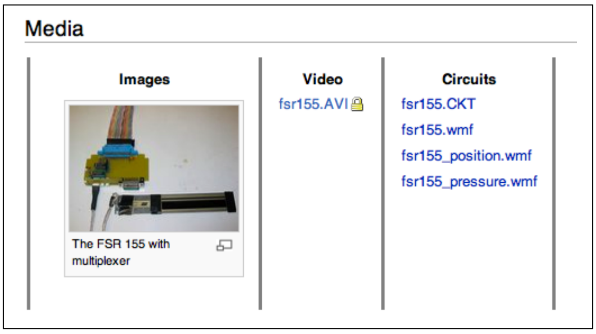


Figure 4: Example of media information available – LuSense CPS2 155.

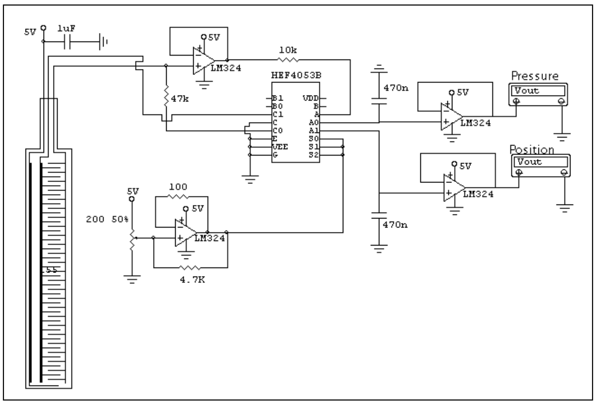


Figure 5: Example of a conditioning circuit - CPS2 155 by LuSense. Circuit design by Patrice Pierrot.

3.2 Reference List

The list of references started as a duplicate of the online resource *Interactive Systems and Instrument Design in Music Working Group* (ISIDM), meant to provide a knowledge-base for researchers and workers in the field. Although the information included in the original knowledge-base hosted at ICMA (International Computer Music Association) and later at McGill is invaluable, it is incomplete and difficult to update (it is a standard HTML webpage and not editable by the public).

It is here proposed that Sensor Wiki provides a much better forum for this knowledge-base, as references are easily added and edited, and the discussion pages allow public communication on changes or direction.

Like the original ISIDM webpage, the Sensor Wiki knowledge-base provides links and references for the following topics:

- Evolution of interactive electronic systems
- Interaction & performance
- Sensors & actuators
- Interface design
- Mapping
- Software tools
- Dance technology

Each topic consists of three sections: *Introductory References*, which introduces the topic to beginners, clarifies some of the vocabulary used, and provides references and links to published work that outlines the topic, an exhaustive *Bibliography* in Computer Music Journal format, and an *Internet Directory*, linking to useful resources available online. We hope that by moving the ISIDM to the Sensor Wiki we will achieve the level of collaboration originally intended by the working group [5].

3.3 Tutorials

SensorWiki.org also includes tutorial pages oriented toward guiding a reader through specific projects from beginning to end. These are usually prepared by individuals, and subsequently edited in minor ways. Initial examples include an overview of basic sensor interfacing techniques, and a lucid and complete tutorial on integrating the USB with microcontroller projects.

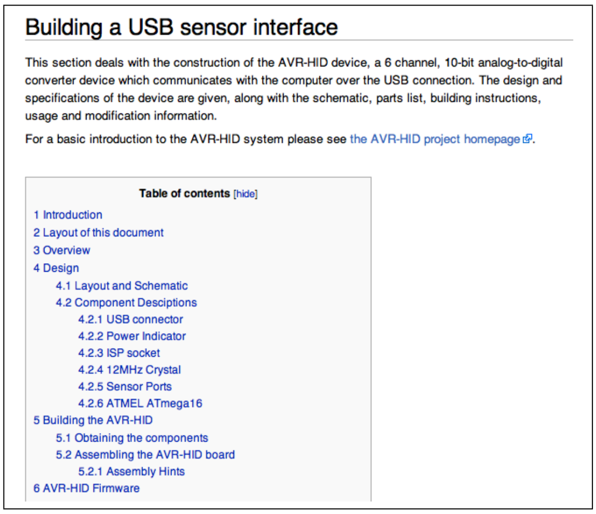


Figure 6: Excerpt - Tutorial on Building a USB sensor interface, by Mark Marshall.

3.4 Applications

It is hoped that this flexible and comprehensive resource will prove useful for researchers who wish to use sensors in their projects. Since it is hosted by a music technology research lab, the Wiki's content tends to be music-oriented, however the information it provides is also useful for robotics, installations, interactive dance systems, and research in a host of other fields.

Reasons for joining the Sensor Wiki project should be clear — the project will allow everyone in our community to benefit from the knowledge and experience of their colleagues. In the same spirit that the NIME and ICMC conferences foster research, new innovation and collaborations, the Sensor Wiki will allow individuals and schools working in the field to grow and learn faster together than they could apart.

4. FUTURE EXPANSION

The future direction and expansion of the Sensor Wiki project will depend heavily on groups and individuals not associated with the IDMIL or the Music Technology Area of McGill University. Although the initial contributors guided the design and formatting of the Wiki in a two-year development process, the content provided thus far is intended to merely initiate a dialogue and sharing of information that will benefit all of us equally, and the basic design and layout may change according to the suggestions of new users.

More accounts of individual experiences with the interface design process are much needed. Backgrounds on musical instruments, commercial controllers, and experimental designs could also be included. Plans for expansion over the coming year include a comprehensive list of actuator technologies to match the sensor list, as well as overviews and tutorials on haptic feedback systems.

5. CONCLUSIONS

Already, SensorWiki.org is a valuable repository of sensor information; as of writing there are approximately 49 "legitimate content pages", with over 33,543 page views. Other institutions have begun linking to the site, such as the University of Oslo, the University of Washington's Center for Digital Arts and Experimental Media, Stanford University's Center for Computer Research in Music and Acoustics, and the Department of Music at Columbia University. We invite researchers and all individuals who wish to share their expertise to participate in the development of the Sensor Wiki. With broad participation, it could serve as a central place for open music technology information, a summary for students new to the field, and a valuable resource for students, hobbyists, and researchers.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of many students and colleagues since the inception of this project in 2004, including Paul Kosek, Mark Zadel, Avrum Hollinger, Stephen Sinclair, Simon de Leon, Mark Marshall, and Darryl Cameron. This project is partially funded by a Natural Sciences and Engineering Research Council of Canada (NSERC Discovery Grant) to the first author and by research funds from the Centre for Interdisciplinary Research in Media Music and Technology (CIRMMT) to the third author.

Pre-emtive thanks are also due to all those who have yet to contribute!

7. REFERENCES

[1] B. Bongers. chapter "Physical Interfaces in the Electronic Arts. Interaction Theory and Interfacing Techniques for Real-time Performance". In M. M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music*. Ircam - Centre Pompidou, 2000.

[2] CircuitMaker. http://www.microcode.com/. 2006.

[3] J. Fraden. *Handbook of Modern Sensors. Physics, Design and Applications*. Springer Verlag, 3rd edition, 2004.

[4] A. Lih. Wikipedia as participatory journalism: Reliable sources? Metrics for evaluating collaborative media as a news resource. In *Proceedings of the International Symposium on Online Journalism*, 2004.

[5] Marcelo M. Wanderley. Report on the First 18 Months of the ICMA/EMF Working Group on Interactive Systems and Instrument Design in Music - ISIDM. In *ICMA Array*, 2002.

[6] MediaWiki. http://www.mediawiki.org/wiki/mediawiki. 2006.

[7] Meta-Wiki. Wiki stats other than the article count. In *http://meta.wikimedia.org/wiki/*, 2006.

[8] E. R. Miranda and M. M. Wanderley. *New Digital Musical Instruments: Control and Interaction beyond the Keyboard*. A/R Editions, 2006.

[9] MoinMoin. http://moinmoin.wikiwikiweb.de/. 2006.

[10] I. O'Murchu, J. G. Breslin, and S. Decker. Online social and business networking communities. In *Proceedings of ECAI 2004 Workshop on Application of Semantic Web Technologies to Web Communities*, 2004.

[11] R. Pallas-Areny and J. Webster. *Sensors and Signal Conditioning*. Wiley Interscience, 2nd edition, 2001.

[12] PhpWiki. http://phpwiki.sourceforge.net/. 2006.

[13] TWiki. http://www.atlassian.com/software/confluence/. 2006.

[14] TWiki:StructuredWiki. http://twiki.org/cgi-bin/view/codev/structuredwiki. 2006.

[15] J. Voss. Measuring wikipedia. In *Proceedings of the 10th International Conference of the International Society for Scientometrics and Infometrics*, 2005.

[16] C. T. E. Wiki. http://www.atlassian.com/software/confluence/. 2006.

[17] WikiMatrix. http://www.wikimatrix.org/. 2006.

[18] Wikipedia:Wiki. http://en.wikipedia.org/wiki/wiki. 2006.

A simple practical approach to a wireless data acquisition board

Smilen Dimitrov
Medialogy
Aalborg University Copenhagen
Lautrupvang 15
2750, Ballerup, DK
smilen@smilen.net

Stefania Serafin
Medialogy
Aalborg University Copenhagen
Lautrupvang 15
2750, Ballerup, DK
sts@media.aau.dk

ABSTRACT
In this paper, we describe the design of a novel wireless acquisition board, its advantages and characteristics, and its applications to NIME.

Keywords
Interface, data acquisition, wireless, interaction.

1. INTRODUCTION

The recent development in human-computer interaction research shows a demand for customized design of electronic sensor systems, used in ubiquitous computing, wearable interfaces, music and dance performances etc. This implies that the interaction designer should consider applicable sensors, in addition to programming the interpretation of their signals in a high-level software platform. Often, the problem occurs beyond this scope of interaction design – being about which hardware/software interface platform to use, to make the sensor signals available as altering variable values in the desired software platform. Market acquisition or custom building of such interface platforms requires acquaintance with data-acquisition systems from an engineering perspective, which is not covered in the academic scope of traditional human computer interaction educations. Such trends are becoming more and more addressed academically (see, for example, [1] [2]), however they require introduction to a broad theoretical base. In this paper, we describe the concept of a simple prototype implementation of a wireless data acquisition board, which physically illustrates theoretical concepts and their systematic binding, which might assist a student of HCI in gaining practical insight.

Custom development of a data acquisition system for interaction applications is made easier (if not possible) by the continuous market development, and availability of inexpensive ICs that are functional units with relatively simple electronic interfaces. In this case, the central interest is in microcontrollers, which both contain on board memory (and can be easily reprogrammed), and different built-in functional units: multiple input A/D converter, an universal asynchronous receiver transmitter (UART) for serial communication, radio transmitter etc. In addition, the market offers a myriad of different devices aimed at establishing a digital wireless radio link, with a different range of possibilities – from simple modulation of a radio wave, up to full implementation of network communication protocols.

In this paper, we focus on implementing the (almost) simplest possible board allowing a multichannel, wireless data acquisition, aimed at interaction processing usage. A data acquisition board commonly accepts a conditioned analog voltage signal, hence it can be used with a variety of sensor systems – as long as the sensed values are mapped to voltage that fits the defined input range of the board. The important point is that any microcontroller that features a multiple channel A/D converter, can be programmed to act as a data acquisition hardware in a relatively simple manner (as long as there is a high-level software IDE with a serial code routine, as is mostly the case).

Our intention is to illustrate a simple data acquisition system involving a single microcontroller circuit that communicates the acquisition data to a PC via serial. The traditional PC asynchronous serial interface allows the possibility to establish a simplex communication line using only one transmission line; that is, a single digital voltage signal. This then allows that the simplest modes of radio wave modulation (AM/FM or the binary versions ASK/FSK) can be implemented to establish a wireless link, using tuned/coupled modulator+transmitter and receiver+demodulator IC devices. Finally, we obtain a prototype system of two hardware devices, a transmitter and a receiver, and a corresponding software driver.

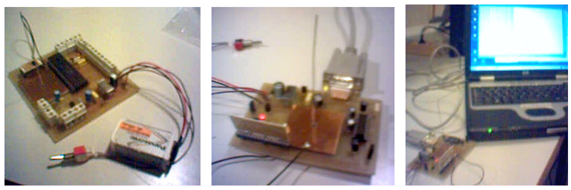


Figure 1. Transmitter device (left), receiver device (center), and receiver connected to a PC (right).

The academic value is first and foremost in the simplicity: all circuits involve only a single side printed circuit board design and basic etching and soldering skills. The design revolves around usage of a microcontroller (in this case a PIC16F74) as a data acquisition hardware that creates an asynchronous serial code, transmitted through a coupled pair of radio devices - digital FM transmitter and receiver ICs (in this case RTFQ1 and RRFQ1[1]), and a MAX-232 chip for conversion of TTL into RS-232 levels

[1] www.rfsolutions.co.uk

used by the PC. A serial driver/decoder is provided in Max/MSP[2], coded fully through its native objects. The simplicity then allows direct relationship to engineering concepts: the particular microcontroller architecture relates to the basic structure of a multichannel data acquisition system. The serial code generated, in both TTL and RS-232 levels, is observable on an oscilloscope, and directly relates to the theoretical basics of asynchronous serial. Furthermore, the system can be seen as a simplex communication line, and we ourselves define the communication protocol, and code and decode it, illustrating basic telecommunication concepts – without however going further into electrically more complex concepts of handshaking. Involving a wireless radio link illustrates basic introduction to radio waves, antennas and propagation – however, most of the complicated RF issues can be avoided thanks to the functionality of the IC devices; only a ground plane and stable power source need to be considered in this particular design. Finally, this requires but four ICs and several basic electronic components (resistors, transistors, capacitors) to implement, all easily accessible in common educational facilities for electronic hardware – and usually regularly covered in a budget of an educational institution, thus minimizing additional expenses to running the implementation of the system as a student exercise (even for private purchase, the total current price of components required should not be over 70-100 Euro).

The simple design and quick implementation, the direct illustration of diverse basic theoretical concepts in a single system, and the relatively low expenses related to building process, are the general points that make the implementation of such a prototype level wireless data acquisition system applicable as an course exercise, executed say in a one week workshop format, and aimed at students of interaction design.

2. RELATED WORK AND INSPIRATION

We consider such device to be like Teleo Intro board by Making Things[3] or the Kroonde wireless data acquisition system by La Kitchen[4], which are marketed for use with Max/MSP by Cycling 74, a software development platform commonly used for customized interaction processing. Such devices are characterized by digitizing of multiple input channels, with sampling rates with an upper bound of around 1 KHz per channel, sampling resolutions spanning from 1 to 16 bit per channel sample, some form of physical connection to a PC (USB, Ethernet) and a driver software, which eventually is exposed as interface as a Max/MSP object – allowing further interaction programming in this software platform. Most commonly, each channel to which a conditioned analog sensor voltage is applied (commonly in the range of 0V-5V) is exposed as an integer numeric value, provided through a corresponding outlet of the driver interface object in Max/MSP; and refreshed at the effective sampling period per channel.

Similar functionality is also found in other systems. Besides these commercial market products, HCI research also discusses custom-built data acquisition interfaces, such as Atomic Pro [4] or WiSe Box [5], which too have a similar functional interface (although aimed at other interaction uses, such as MIDI or OSC from CNMAT [7] for music instrument hardware).

The Teleo Intro board implements this functionality through a PIC microcontroller, which samples the four analog inputs on the Teleo, and communicates the results to a PC via USB connection. In hardware, the analog inputs on the Teleo Intro are connectors with screws, and in addition, +5V and ground connectors are offered. This allows that simple sensors (such as a potentiometer) can be attached directly to the board, mechanically instead of through soldering, which makes it quite accessible for education uses - and inspired the same solution for our prototype system as well.

3. THE PROPOSED DESIGN

The system design revolves around usage of a PIC16F74 microcontroller, coupled digital transmitter/receiver IC's RTFQ1 and RRFQ1 (operating at 868 MHz) and a MAX-232 line signal converter ICs. This results with two circuits – a transmitter and receiver circuit, where the central issue is to program the PIC to behave as a data acquisition hardware. The system is completed with a Max/MSP decoder/driver.

Power sections are implemented through voltage regulator ICs, and power connectors are 9V battery connectors, allowing both battery and standard adapter power-up.

3.1 PIC data acquisition program

The PIC program is an extremely simple endless loop, which simply activates the A/D conversion on an analog input, and stores the results on a corresponding index in an array; the process is repeated for all available analog inputs. Based on this array, a 35 bytes string is produced, which represents a single update word – and sent via serial, which involves generating an TTL voltage signal on one of the microcontroller pins. A pseudo-code looks like the following:

```
Initialize PIC

loop forever

  erase out array
  read ADC 1
  store ADC 1 value in out array
  read ADC 2
  store ADC 2 value in out array
  ...
  read ADC 8
  store ADC 8 value in out array

  format string based on out array
  send string based on out array

end loop
```

This program involves knowledge of only a few registers in the PIC and can be very easily written in a high-level software. We used a BASIC interpreter, which almost follows the pseudocode structure directly. The serial uses a software bit-banging routine, instead of using the hardware UART, thus it can use any I/O pin of the microcontroller for serial code output.

---

[2] www.cycling74.com

[3] www.makingthings.com

[4] www.la-kitchen.fr

Using a Microchip PIC microcontroller allows for usage of a free integrated development environment for writing and compiling a given PIC program. However, a PIC also allows for usage of the classic David Tait [6] hardware serial programmer, as well as variety of freeware software programmers, for programming the microcontroller. The hardware programmer used here is also custom-built version of the Tait programmer, and in this case can also be integrated in the educational exercise concept, as it not only discusses basic microcontroller programming, but also usage of the parallel port of a PC as a programmable power supply, and synchronous serial communication.

3.2 Transmitter circuit

The transmitter circuit hosts the power supply section, the PIC and the transmitter IC RTFQ1, and is powered by a 9V battery. The antenna is a 9 cm long wire, and a ground plane is implemented by a separate non-etched PCB. The dimensions of the unit are 88 x 94 mm, the antenna can be folded and the unit can be worn in a pouch or a pocket.

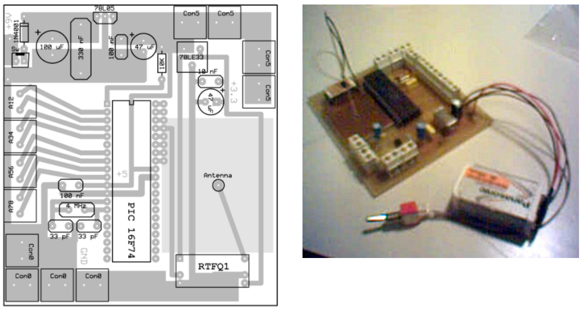


Figure 2. PCB Layout and implementation of the transmitter circuit

The PCB layout on Figure 2 shows that only 15 components are used for the circuit, the rest of it is connectors.

3.3 Receiver circuit

The receiver circuit hosts the power supply section, the receiver IC RRFQ1, and the line level converter IC MAX-232; as well as a DB-9 connector for serial connection to a PC. The dimensions of the unit are 94 x 64 mm, and is powered by a standard AC/DC adapter set at 9V.

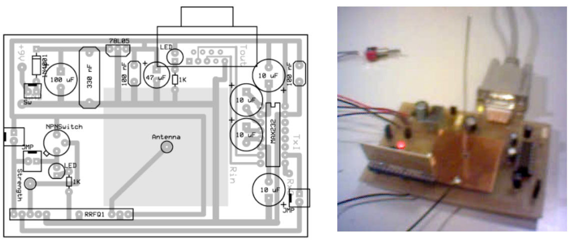


Figure 3. PCB Layout and implementation of the receiver circuit

The device also features two LED diodes – one to indicate power up, and one turns on whenever the transmitter is turned on (which is made possible by a RF signal strength measurement pin on the RRFQ1 IC).

The electric design is again simple, however, it doesn't take into account that when the transmitter is not operating, the receiver IC generates in essence a random digital voltage. Such a signal usually causes errors in a PC, hence there has to be a procedure where the transmitter is turned on first, and then the receiver.

3.1 Max/MSP decoder/driver

The usage of the synchronous serial signal in Max/MSP involves

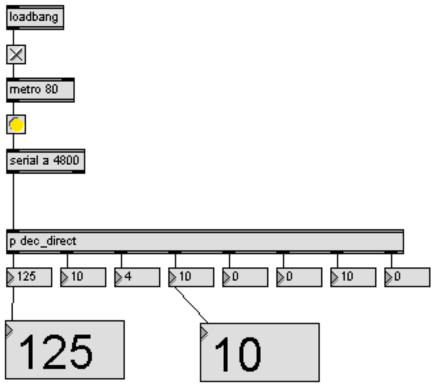


Figure 4. A sample Max/MSP patch demonstrating use of the decoder

the native Max object *serial*, which provides the characters received through the serial port of the PC as a consecutive sequence of integers (ASCII codes). From the original 35 bytes word, we need to extract the 8 packed values that represent the latest sensor values acquired by the transmitter, which is performed by a custom Max decoder.

As Figure 4 demonstrates, the decoding functionality is encapsulated in a subpatcher object call *p dec_direct*, which accepts the incoming character stream as an input, and provides the extracted sensor channel values as integers through the corresponding outlets, which is again a principle that the Teleo uses as well for its interface object.

The decoder itself is implemented by native Max objects, which allow basic understanding of a decoding process through a graphic programming environment. Although this is useful from an educational perspective, where such representation might be closer to the student than a serial driver in C, it may be so doing so at expense of quality – as Max patches using this particular driver in its current version, crash quite often.

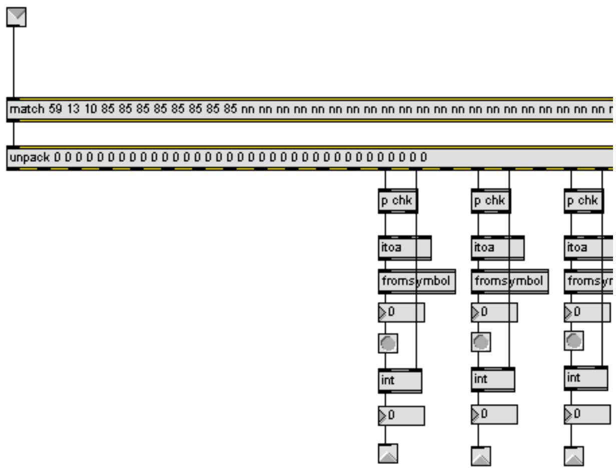


Figure 5. The Max/MSP serial decoder implemented through native Max objects

4. FINAL REMARKS

Concerning the first version of the prototype, lot of things can still be improved. Advantages of the prototype are the fact that it provides 8-channel, 8 bit per channel wireless data-acquisition, and a single 9V battery provides up to 20 hours of transmitter operation.

However, even at this level certain measures can be undertaken that allow a more reliable operation of the system – usage of onboard UART on the PIC; compacting the redundant 35 byte codeword; implementing a gating circuit at the receiver that opens only when transmitter is sending, improving possible RF construction issues, and increasing the communication speed to the maximum offered by RRFQ1/RTFQ1 which is 9600 bps. A reliable and slightly faster circuit can then already be applied in any custom interaction design that could also implement a Teleo or a Kroonde as interaction data acquisition boards.

In the sense of actual usage, the exact same concept can be further engineered, by using other microcontroller devices (possibly with built in wireless transmitters) and/or other wireless ICs (possibly ones that use a network communication protocol), which of course increases the design and implementation complexity.

However, as the system currently is, its simplicity is an excellent opportunity to physically illustrate the most basic engineering issues in data acquisition for interaction design to students of HCI.

This device has already been successfully used in several interaction design and music projects, where students appreciated mostly the wireless feature which enhanced motion possibilities in a virtual reality context [3].

Having the opportunity to build such a system and use it, gives the students practical insight in electronics, telecommunications and software issues, an insight that would assist the HCI student both in evaluating different data acquisition systems on the market for a given interaction task, and give a starting point if a building of a custom data acquisition system is required.

Detailed circuit diagrams and instructions on how to assemble this board are available at www.imi.aau.dk/~sts/board.html.

5. REFERENCES

[1] Putnam, W., Knapp, R. B., "Input/Data Acquisition System Design for Human Computer Interfacing", online course notes, Stanford University, 1996 http://ccrma.stanford.edu/CCRMA/Courses/252/sensors/sensors.html

[2] Cook, P. R., "Real-Time Computing For Human Computer Interfacing", online lecture notes, Princeton University, http://soundlab.cs.princeton.edu/learning/tutorials/RealTime/realtime.html

[3] Nordahl, Rolf. Auditory rendering of self-induced motion in virtual reality. M.Sc. project report, Dept. of Medialogy, Aalborg University Copenhagen, 2005.

[4] Fléty, E. AtoMIC Pro : a Multiple Sensor Acquisition Device. Proc. NIME 2002.

[5] Fléty, E. The WiSe Box : a Multi-performer Wireless Sensor Interface using WiFi and OSC. Proc. NIME 2005.

[6] David Tait homepage: http://www.people.man.ac.uk/~mbhstdj/piclinks.html

[7] Freed, A. and Wright, M. Open SoundControl: A New Protocol for Communicating with Sound Synthesizers. In *Proc. of ICMC 1997*.

Mapping strategies in DJ scratching

Kjetil Falkenberg Hansen
Dep of Speech, Music and Hearing, KTH
Stockholm, Sweden
kjetil@kth.se

Roberto Bresin
Dep of Speech, Music and Hearing, KTH
Stockholm, Sweden
roberto@kth.se

ABSTRACT

For 30 years Disc Jockeys have been expressing their musical ideas with scratching. Unlike many other popular instruments, the equipment used for scratching is not built as one single unit, and it was not intended to be a musical instrument. This paper gives an overview of how DJs use their turntable, vinyl record and audio mixer in junction to produce scratch music. Their gestural input to the instrument is explained by looking at the mapping principles between the controller parameters and the audio output parameters. Implications are discussed for the design of new interfaces with examples of recent innovations and experiments in the field.

Keywords

Instrument mapping, DJ, scratching, controllers, virtual instruments

1. BACKGROUND

The success of a musical instrument will have a multitude of reasons. Some immediate suggestions are that it is cool to look at while somebody plays it; it sounds really good; one can express feelings and communicate emotions with it; it is cheap; you can feel its vibrations; it has blinking lights; it has a shoulder strap; you can hit it... Scratching is one such successful "instrument" that takes some effort to get acquainted with.

Scratching is, firstly, not an instrument, it is what Disc Jockeys (DJs) sometimes do with turntables and mixers. The popularity and history of this musical niche has been explained in several aspects in previous works (see for instance [18, 20, **?**, 3, 9]). The combination of using both a turntable and a mixer laid the ground for scratching about three decades ago, but despite this, DJs only refer to their instrument as "turntable", inadvertently including the mixer unit. Indeed, analysis of common DJ scratching techniques have shown that the mixer is almost as important for the musician as the turntable [10]. Imperative to the control and interface part is the sound *source*, the vinyl record. Sound generation is therefore controlled by two different devices, but dependent on what is currently playing on the record. In this sense, scratching can be compared to new computer based instruments that has separated the interface and controllers from the sound generator.

Recent alternative interfaces for scratching follow to a great deal the separation of a sound level unit and a sound 'source' unit. Why is this so? Should not the effort of the new interface developers be focussed on the task of making *one* instrument for scratching? Evidently, there is a mismatch between development of the turntable-part and the mixer/crossfader-part: this is because the struggle to make an acceptable substitute to the turntable is of highest priority. For instance, all leading manufacturers have each marketed a number of devices for scratching digitally stored sound, but only one of them has produced a mixer with additional control possibilities [19].

2. CONTROLLERS AND MAPPINGS

Mappings in interfaces for scratching are quite intuitive, but not simple one-to-one mappings as it may seem at a first glance. One hand controls the playback speed of the record by pushing and dragging it with fast, short movements. The other hand controls the sound level out from the system, in general only the extremes of total silence and full sound[1].

2.1 Common interfaces

2.1.1 The turntable and vinyl record

Playback speed modulations directly affects the pitch we hear, but they do more also. Sound samples[2] used in scratching are often words, short phrases, synthesized sounds, instrumental sounds, and sound effects from movies or daily life. However, they are seldom played as withheld and straight tones of constant pitch. Therefore the perception of the sound changes dramatically when played at different speeds and backwards. This is true not only in the time domain when sounds are compressed or stretched out in time, but also in frequency domain when the whole spectrum is shifted up or down. To disguise the origin of the sound being scratched further, there is normally a fast acceleration and deceleration with each hand gesture that transforms the sound.

[1]The DJ mixers also have tone controllers and linear amplitude sliders, but the most utilized control is the logarithmic crossfader which works as an on-off switch for sound level.

[2]In this text, *samples* are recorded sounds on vinyl records. Samples can be short (from a tenth of a second) or long (to more than a minute), but typically they are a bit less than one second.

Another aspect to consider related to the mapping between the record-moving gesture and the sound we hear, is clearly the sound source on the record itself. As stated, pitch changes and artifacts in the time domain will occur in the interaction with the turntable. But these are not at all consistent for all sounds. For instance, short sounds of low frequency are hard to stretch out in order to manipulate them with the crossfader, while longer sounds easily allows virtuous use of crossfader. Some sounds are unfit for playing with pitch, such as the sound of a snare drum hit, while others such as a sample of James Brown shouting 'aaayeah', are perfect for that purpose.

Not only the sounds on the record are important for identifying the mapping. Even the silent parts before and after each sound sample must be taken into account. If the DJ starts to move the record *before* the needle reaches the spot where the sound starts, the sound will start with a high pitch. If she starts the move when the needle is located *within* the sound, a sudden glissando effect can be heard. Likewise, if the DJ pulls the record back and forth repeatedly over an entire short sound sample, we will hear a succession of sharp tone onsets, while the same movements "inside" for instance the long James Brown sample will generate a very different, more siren-like sound gliding fast up and down in pitch.

Consequently, if the record on the turntable consisted of a long, unchanging sound, such as a synthesized tone of constant pitch, the musician's gesture in moving the record would be more audible and apparent.

The three most common gestures for moving the record are to place the hand flat down with fingertips on the record and use the elbow to move; to hold the hand at an angle and use the wrist; and to hold the hand steady and only make small movements by using the finger joints. In the first two cases, the hand and fingers are flexible and can supplement the main gesture.

2.1.2 The audio mixer

The audio mixer is placed between the turntable and the loudspeaker, so the signal from the record is always going through the mixer. It has a number of controllers, most important are the crossfader, a line/phono switch, a volume fader and one or more knobs for equalization (tone control). DJs use the crossfader much more than the other mixer controllers. During recent years, the crossfader has evolved from fading linearly between the two turntables in the DJ set-up to being a logarithmic fader with very steep fading curves that in practice is only an on-off switch for sound. Traditionally, the line/phono switch was used for turning the sound quickly and sharply on and off, but today it is superseded by the crossfader, which is much better suited to fast manipulations than the switch.

The crossfader knob can be moved using different hand gestures, but two methods are dominant. The knob is either pinched between the thumb and a finger and pushed the short distance on the fader that runs from silent to sound, normally adjusted to 1-2 millimeters, or it can be lightly hit with one or more fingers against the thumb which acts as a spring, bouncing the fader back. Either way, the starting position can be both in the silent and the full sound area. The crossfader can be reversed so that the silent part can be on either the extreme right or the left side of the fader run.

2.2 Current mapping in scratching

By following principles for describing instrument mapping strategies suggested by Hunt et al. (in [12, 13, 14]) an overview of the existing mapping between gestures and sound in scratching is presented here. The controller parameters considered are record speed, sound source (on the record), the needle's placement in the sound groove, and on the mixer the use of crossfader, volume slider and tone control knobs. The audio output parameters considered are pitch (not necessarily in the meaning of discrete semitone steps), tone onsets and durations, and tone or sound characteristic such as timbre and dynamics. Figure 1 shows the relations between controller and audio output parameters.

2.2.1 Timbre and dynamics

Timbre is determined by the sound source used, and the speed at which it is played back. In addition, the DJ can control timbre to a certain degree using the tone control knobs, which is done for certain scratch techniques [9].

Dynamics, or sound level variation, is a very important factor for expressive playing in most musical instruments. However, it is rarely found in scratching. Sound level is determined by the volume slider on the mixer, the sound source and the playback speed. The crossfader can not be regarded as a controller for dynamics, as its fading curve is far too steep. The needle's placement in the sound groove can have a great effect on dynamics, as many sounds decrease in amplitude over time.

Timbral and dynamical variations can also be achieved by choosing locations with different amount of wearing on the record. In records where one sound is repeated (e.g. for several minutes, as in [7]) the sound can be more detoriated in some grooves than others, making the same sample sound different depending on where the needle is located [11].

2.2.2 Pitch

Pitch is determined by the sound source and is adjusted straightforwardly with the playback speed. Experienced DJs sometimes use the pitch adjustment slider on the turntable to "tune" the sound sample to the tonality of the piece to which they are scratching. A large part of all (scratched) tones have the same speed as, or close to, the revolving turntable [10]. For many of the favored sounds on the record, it is hard to hear a fundamental frequency. Several samples, for instance the popular "Fresh" (a long *shhh* sound, [8]), have a noise band with a broad maximum, inducing the perception of some pitch.

2.2.3 Tone onsets and durations

By tone onsets it is meant both the start of a tone and the characteristics of its attack. Attack properties are controlled by record speed, sound placement and sound source. A slow push produces a slow rise in pitch and a "soft" attack, while a fast push produces a harder attack. A movement starting with the needle positioned inside a sample produces a softer attack than the same movement starting before the sample starts. The sample of a snare drum hit as sound source will have a very different attack from a sample of a moaning soul singer.

The rapid stream of tone attacks is a very noticeable and distinguishing aspect of scratching. In a typical technique with a duration a couple of tenths of a second, there can be as many as 6-9 tone attacks [10]. The most important controller for generating such rapid onsets is the crossfader, especially when the fingers bounce it against the thumb. Also quick shifts in playback tempo from go-

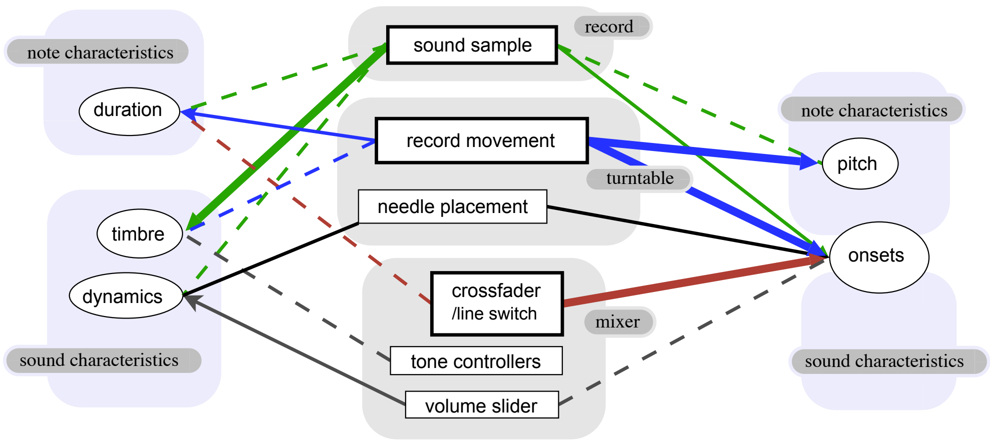


Figure 1: Mapping sound output parameters (left and right) to control parameters (middle) in a normal instrument setup for scratching. Onset and pitch are undeniably the most crucial sound attributes, and are placed to the right in the figure. Line thickness (from dashed to bold lines) represent the importance of the connection. The bold boxes are considered to be the most essential control parameters: record movement, sound sample properties and crossfader movement.

ing forward to going backward generates onsets. Certain techniques can produce more than 20 onsets a second only using the record speed. A third control possibility is to use sound sources where short sounds are placed close to each other, for instance a slow drum roll or hand-clapping. One single gesture of moving the record over such samples produces a multitude of tone attacks.

The length of the sound sample influences the durations of the tones which can be controlled by hand gestures on the turntable. The crossfader can at any time shorten the duration by stopping the tone. It is often hard to sustain short samples for long, which is also still the weakest point of all the interfaces for scratching digital sound files. Even long samples can be hard to sustain with a constant speed, as the turntable's tone arm obstructs and makes circular gestures problematic.

2.3 New interfaces for scratching

New interfaces introduce new possibilities. Generally, the common first feature is to give control over an additional parameter, either in the sound itself or in processing the sound [16]. With scratching, the first step has been to reproduce the existing hardware (the turntable part) in software models and with alternative controllers, often realized as some type of a revolving wheel resembling the turntable. This strategy is obviously in conflict with the design principles lined up by Cook [6]. The pursuit to produce an acceptable virtual turntable has led to commercially available interfaces for controlling sound files that are quite good. Solutions for allowing control over additional parameters, however, is still a rather unexplored field, and the big companies have not yet acknowledged the potential.

In previous NIMEs and in related journals, and even in commercially available products, several new interfaces for DJs have been envisaged. This section of the paper gives a short review of possible mapping strategies of a selection of these interfaces with regards to the previous section.

Of the many interfaces manipulating digitally stored sounds, *Final Scratch* by Stanton has probably been most accepted, using a time-coded vinyl record to control sound files on a computer. Other such interfaces include devices for scratching with CDs, marketed in various designs by all leading manufacturers (e.g. Pioneer, Technics, Stanton, Vestax and Denon). Common for this group of interfaces is that they do not offer new ideas for controllers nor new mapping possibilities.

DVD scratching is a similar concept to above mentioned scratch tools for digital media. Although few companies have marketed instruments for this, Pioneer being one exception [17], it is a promising performance concept for VJs (video DJs). Current version can treat the video image equivalently to sound by modulating the playback speed. The visuals introduce new output parameters for the interface, which parameters can be mapped to other controllers than just the 'turntable' gestures.

Mixxx by Andersen [1] is developed both as a set of performance tools and as a means of studying DJ performances. Both mixer and turntable interfaces are replicas of the standard interfaces, but they also send MIDI messages. For performance, it enables to have a mapping layer between the controller parameters and the audio output, so that they can be used for controlling new or existing sound properties. *Mixxx* also includes a new visual feedback technique in an augmented turntable, the *Fisheye* [2], a method for projecting a waveform image on the record. In the presented implementation, *Mixxx* is not addressing DJ scratching specifically.

Mixxx also includes a new visual feedback technique in an augmented turntable, the *Fisheye*[2], a method for projecting a sound image on the record.

The *D'Groove* by Beamish et al. (in [5] and [4] introduces haptics, induced by a force feedback motor, as a feedback method in the turntable interface. This allows for haptic 'visualization' of the sounds on the record, which assists the DJ in for instance finding the sound sample 'borders'. In tests, experienced scratch DJs quickly start to experiment with new techniques and performance ideas, and learned to use the novel features that haptic feedback provided. Some new controllers were also implemented, but did not suggest new mapping layers or strategies, as was the case with *Mixxx*.

Skipproof [11] is a software tool for evaluating synthesized scratch techniques, and also a scratch performance instrument. The controller and audio output parameters found in common DJ set-ups can be mapped freely to various devices for computer input, such as MIDI interfaces or gesture sensors. *Skipproof* does not have the general approach to DJing as the above mentioned interfaces, but is focussing on the skip proof feature found in specialized records for DJs where a one revolution long sample (1.8 seconds) is looped. This allows for designing specific scratch interfaces that contrast the established ones.

16padjoystickcontroller by Lippit [15] is a realtime sampling system for scratch DJs. Lippit acknowledges that DJs are fully occupied with their hands during performance and suggest supplementary controllers, a foot switch, a joystick and a drum pad, that enable the DJ to interact with their own performances in real-time. With these controllers, DJs can record and manipulate portions of their scratch performance. In such context, the added controllers are separated from the existing ones and represent isolated instrument systems without possibilities for interaction.

Samurai mixers by Vestax [19] are the only commercially available tools from a major manufacturer that have added possibilities for sound control with the crossfader. The crossfader is programmable so that the DJ can handle amplitude patterns instead of the normal fading curve, so that one gesture on the crossfader can generate several onsets, not just one. Vestax has also marketed a turntable with integrated crossfader, the *QFO*, but this interface does not introduce new control possibilities or mappings. It illustrates however the need for a dedicated scratch instrument.

3. CONCLUSIONS

The existing mappings between gestures and sound with a turntable and a mixer should be considered when designing and building new interfaces for scratching. Up to now, gestures used for scratching have originated from the idea of reading a sound file at varying speeds and direction, with a one-to-one mapping between the extent of a (hand) movement and the position in a sound. Regardless of the kind of gesture, it is always a measure of distance. Gestures for controlling crossfader, or sound on/off, are not as intuitive as the turntable-controlling gesture, and also much less studied.

In general, commercial manufacturers of equipment and instruments focus on existing control paradigms instead of exploring new possibilities at hand. Immediate responses from DJs participating in the described interface developments and experiments, indicate an openness to new control concepts and mappings for scratch performances. New interfaces will probably contribute to develop the playing styles and DJ techniques in other directions.

DJs have with great consensus developed a common language of playing techniques that require complex, synchronized gestures to be performed correctly [9]. Despite the rather unusual and hard-to-master handling of the instrument, its principles can be applied to other interfaces, both for music performance and for general human-computer interaction.

4. ACKNOWLEDGEMENTS

This work is supported by S2S^2 (acronym for Sound to Sense, Sense to Sound, pronounce s two - s square), a FET-Open Coordination Action (number of contract: IST-2004-03773).

5. REFERENCES

[1] T. H. Andersen. Mixxx: Towards Novel DJ Interfaces. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, pages 30–35, Montreal, Canada, 2003.

[2] T. H. Andersen and K. Erleben. Sound interaction by use of comparative visual displays. In *Proceedings of the Second Danish HCI Symposium*. HC Tryk, November 2002.

[3] T. Beamish. A Taxonomy of DJs. 2001.

[4] T. Beamish, K. MacLean, and S. Fels. Manipulating Music: Multimodal Interaction for DJs. In *Proceedings of the Conference for Human-Computer Interaction 2004*, Vienna, Austria, April 2004.

[5] T. Beamish, K. van den Doel, K. MacLean, and S. Fels. DGroove: A Haptic Turntable For Digital Audio Control. In *Proceedings of the 2003 International Conference on Auditory Display*, Boston, 2003.

[6] P. Cook. Principles for Designing Computer Music Controllers. In *Proceedings of the CHI 00*, NIME Workshop, 2000.

[7] DJ 1210 Jazz. Book of Five Scratches. Book 2. Snickars Rec., SR1206, 2001.

[8] Fab Five Freddie. Change The Beat. 12" Celluloid Records CEL 156, 1982.

[9] K. F. Hansen. The Basics of Scrathing. *Journal of New Music Research*, 31(4):357–365, 2002.

[10] K. F. Hansen and R. Bresin. Analysis of a Genuine Scratch Performance. In *Proceedings of the Gesture Workshop*, pages 519–528, 2003.

[11] K. F. Hansen and R. Bresin. *The Sounding Object*, chapter Complex Gestural Audio Control: The Case of Scratching, pages 221–269. Mondo Estremo, Florence, Italy, 2003.

[12] A. Hunt and R. Kirk. Multiparametric control of real-time systems. In J. R. . M. W. Marc Battier, editor, *Trends in Gestural Control of Music*. FSU - Florida, April 2000.

[13] A. Hunt and R. Kirk. *Trends in Gestural Control of Music*, chapter Mapping Strategies for Musical Performance, pages 231–258. Ircam - Centre Pompidou, 2000.

[14] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32(4):429–440, 2003.

[15] R. Khazam. Electroacoustic alchemist. *The Wire Magazine*, 160:36–, 1997.

[16] T. M. Lippit. Realtime Sampling System for the Turntablist, Version 2: 16padjoystickcontroller. In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, 2004.

[17] J. A. Paradiso and S. O'Modhrain. Current Trends in Electronic Music Interfaces. *Journal of New Music Research*, 32(4):345–349, 2003.

[18] Pioneer. Pioneer DVJ-X1, 2003.

[19] D. Toop. *Rap Attack*. South End Press, Boston, USA, 1984.

[20] Vestax. Vestax Samurai Mixer Series, 2006.

[21] M. White. The phonographic turntable and performance practice in hip hop music. Technical report, 1999.

'GXtar', an interface using guitar techniques

Loic Kessous
Tel Aviv University
Speech, Language and Hearing Laboratory
Chaim Sheba Center, Israel,
+972 54 204 52 98
Kessous@post.tau.ac.il

Julien Castet
Institut National Polytechnique de Grenoble
46 av. Felix Viallet
Grenoble, France
+336 08 67 28 08
Julien.castet@gmail.com

Daniel Arfib
CNRS-LMA
31 Chemin Joseph Aiguer
13402,
Marseille, France
+33 491 16 4210
arfib@lma.cnrs-mrs.fr

ABSTRACT

In this paper we describe a new guitar-like musical controller. The 'GXtar' is an instrument which takes as a starting point a guitar but his role is to bring different and new musical possibilities while preserving the spirit and techniques of guitar. Therefore, it was conceived and carried out starting from the body of an electric guitar. The fingerboard of this guitar was equipped with two lines of sensors: linear position sensors, and tactile pressure sensors. These two lines of sensors are used as two virtual strings. Their two ends are the bridge and the nut of the guitar. The design of the instrument is made in a way that the position of a finger, on one of these virtual strings, corresponds to the note, which would have been played on a real and vibrating string. On the soundboard of the guitar, a controller, with 3 degrees of freedom, allows to drive other synthesis parameters. We then describe how this interface is integrated in a musical audio system and serves as a musical instrument.

Keywords

Guitar, alternate controller, sensors, synthesizer, multidimensional control.

1. INTRODUCTION

When the design of an electronic instrument is inspired by the one of an acoustic instrument, then the performer can use instrumental techniques and motor behaviors, which he has developed by a long practice time.

Thus, the performer will be able to transpose his skills to other applications. Indeed, he will have access to new sonorities, new possibilities of gestural control of the sound materials without having to learn completely new instrumental techniques.

2. RELATED PREVIOUS WORKS

2.1 Guitar-like controller

2.1.1 MIDI systems for guitar

Some systems use electromagnetic microphone to measure the vibration of the strings (independently for each strings) [6]. Then, by analysis of the signal (usually pitch tracking and energy envelope following) and interpretation of these features, MIDI message are generated. The commercial systems available generally don't consider the spectrum characteristics of the string vibration. Recently, C. Traube and P. Depalle have proposed other useful features extractions to qualify the plucking style [2]. This is certainly a good approach to capture standard guitar playing. Our approach is different, in the way that we want to build a new controller inspired by a guitar but which has his own identity.

2.1.2 The Ztar series

The Ztar series [4] are guitar-like MIDI controllers that are deeply related to the controller described here. But some characteristics of them, for example the discrete fingerboard, in our point of view, limit their expressive possibilities. The GXtar follows the philosophy of the Ztar series, but we tried to bring new ideas, such as continuous pitch control and aftertouch, and technological and technologic solutions to create a new more expressive instrument.

2.2 Other string-like controllers

Several efforts have been made to design violin-like controllers [1] [4]. Obviously, both from the technologic point of view and from the gesture approach, there are many common points between new violin-like new instruments and guitare-like new instruments. Probably, the 'Superpolm' [4] is the more related to the GXtar. But, in our approach we have tried to put more effort on some aspects in order to design a controller with more similarities with the acoustic instrument concerning position of the non-preferred hand fingers and pitch control.

2.3 Touch sensors and fingerboard

To measure pressure and displacement of fingers on a fingerboard, FSR (Force Sensing Resistors) sensors offer interesting possibilities to measure both pressure and position on a fingerboard. Other technologies as, for example, optic fiber networks can be good solutions too [5]. They have been used successfully in related applications, but are difficult to be adapted to the geometry of the guitar neck and are much more expensive. 'Old school' magnetic strips can also be useful but can only measure position and not pressure. It's generally the same for electric current voltage division. FSR are generally not expensive and can provide both position and pressure measurements. Even if some sensor manufactures offer to build 'made to measure' sensors it's generally hard to find ready to use sensors with an adapted size. That's the reason why some concessions have been made in this project compared to the initial project, i.e., a 6 strings instrument.

3. DESIGN OF THE GXTAR

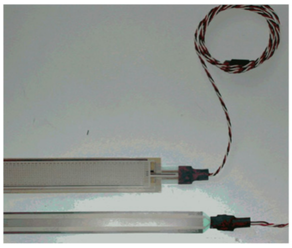
Usually, in the acoustic guitar play, the non-preferred hand acts as selecting notes and the preferred hand acts as playing them by plucking the string. Other techniques like Hammer-on, Pulling-off and glissando are generally not considered in the design of electronic guitar-like controllers. The invention of the electric guitar associated to the evolution of amplification system and audio effects has allowed guitarist to invent and/or develop new playing techniques. One of this techniques is the 'tapping', where both the preferred hand and the non-preferred hand are involved in the excitation of the string, selecting the note and producing the vibration at the same time by tapping on the fingerboard. The concept of the GXtar uses this idea by allowing the non-preferred hand to select and play notes when the other hand acts on spectral continuous parameters. Glissando and other techniques previously cited are also considered in the Xtar.

3.1 Body of the controller

Figure 1 Global view of the board

The physical 'support' used is the body of a real electric guitar. It is a 'solid body' (i.e. without resonating chamber) guitar (imitation of a model Gibson SG) being previously equipped with brass strings. Before modification, the sound was generated by an electronic analog circuit. The frets were removed in order to be able to stick sensors on the fingerboard. Indeed, the new instrument is a fretless string instrument (fig 1). The sensors being of a width larger than usual spacing between strings in standard guitars, it was not possible to provide 6 virtual strings but only 2. Although it would be interesting to have 6 strings available the instrument with 2 strings remains interesting. In fact, some musicians need just two strings to produce fantastic music (as for example the bass player of the legendary rock band ‘Morphine’). And this observation has been the origin of our hypothesis, which gives a musical interest to a couple of strings. The sensors having a dimension slightly longer than their zone of effective measurement, the nut was removed, so that measurement can be effective from the first millimeters of the fingerboard. We then add a new nut to support strings used as guide as it will be explain in a next part of this paper.

3.2 FSR for the non preferred hand



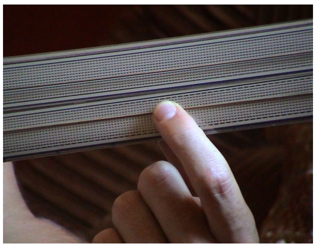


Figure 2. Position and tactile pressure FSR sensors / Playing on the fingerboard

For the non-preferred hand, tactile position and pressure sensors equip the fingerboard. The sensors used represent two virtual strings side by side (fig 2). The position of the fingers on these strings, and the pressure exerted by the fingers on the neck, are measured by superimposed different sensors. These sensors are FSR sensors. We decided to preserve the relation present on the acoustic or electric string instruments between the position of a finger on a string and the pitch of the played note. As on the acoustic instrument, the relationship between lengths of vibrating string fundamental frequencies of vibration corresponding is preserved. To obtain the played frequency, it is thus necessary to know the length of string between the finger position and the bridge. On our support, a string from the nut to the bridge measures 62 cm, but our sensor makes it possible to measure only one 39 cm length zone. Nevertheless 39 centimeters are sufficient to practically cover the totality of the neck, i.e. the equivalent of an octave on a single string.

To play a note using the fingerboard and to define a velocity for this note we use the principle of double threshold applied on the pressure. This principle is generally used to position to derivate velocity, as for example, in the legendary 'Radio Baton'. Previous experiences with other sensors using only the Max/MSP external library or other third party externals to implement this were not satisfying. A new external, called ‘whack’ (as the double threshold implementation was called in the documentation of the 'Radio Baton') has been implemented using the Max/MSP external SDK.

3.3 3D joystick for the preferred hand

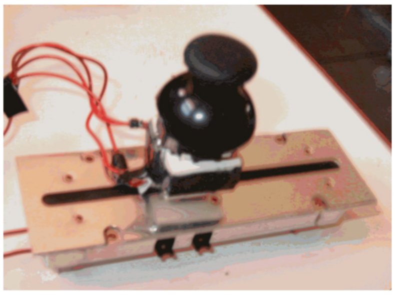


Figure 3. The 3 degree of freedom controller composed of a mini joystick and a slider

The control mainly assigned with the preferred hand is a control with 3 degrees of freedom. Two degrees come from a mini-joystick, the third comes from a rectilinear potentiometer which is used as rail of displacement for the joystick. The choice of a mini-joystick rather than of a joystick of higher size is justified by the idea to use only the grip made of the thumb and the index to handle it, whereas the palm of the hand and the other fingers handle and move the support of this mini-joystick. We chose to use a rectilinear potentiometer usually called a slider, and to superimpose a joystick on top of it. Thereby, the joystick was fixed on the moving part of a slider. The two devices associated in this way offer the control of 3 parameters simultaneously (fig 3).

Displacement along the rectilinear potentiometer reminds the displacement of the plucking position on the strings. The effect on an acoustic or electric guitar is strongly correlated to a comb filter effect. Finally, after different experiments, we selected a slider with an adequate length in order to reinforce this metaphor and the relation with the standard guitar playing techniques. The distance from the bridge is then correlated to the sound characteristics as it used to be for plucking in an acoustic or electric guitar. One can then use the possibility to get a different sound by choosing a different distance from the bridge for plucking.

3.4 Special plectrum

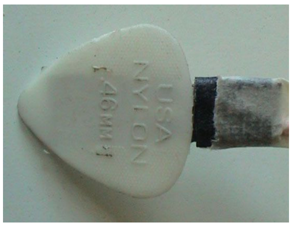


Figure 4. The plectrum-like controller: the sensor is between two glued thin plectra (0.46 mm)

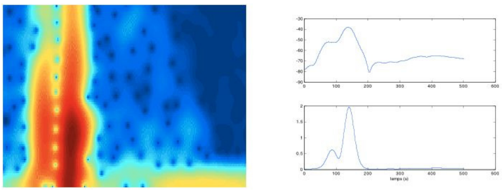


Figure 5. The left: a time-frequency analysis of a plectrum signal; right: two spectral lines

We also studied the design and use of a new control tool: a special plectrum. It is composed of a piezo film sensor fixed between two thin plectra with glue (fig 4). The idea is to extract instrumental gesture from the analysis of the 'plucking' signal coming from the sensor. Indeed, the signal coming from this new plectrum contains several cues about the performer intention and expression. Both the plectrum grasping and the impact with the string can provide interesting sources of control. This work has been carried out for specific initial condition: the case where plectrum and string are in contact before the acquisition and when the string is not already in vibration. First we made a real time asynchronous analysis; we detected time increase, maximum range and number of oscillations. In a second part, we tried to improve the result with the RMS (Root Mean Square) energy. A mapping combination of the features from oscillation and RMS measurement appears interesting to obtain an efficient control parameter. However this method was not fully satisfactory by looking only in the time domain. A time-frequency analysis has been done with this objective. The time-frequency analysis reveals an interesting element: we observe the appearance of a second spectral line (frequency of oscillation) with an intensification of playing (fig 5). A strong grasping of the plectrum can be detected with this direct consequence. This result is an example of relation between the intention and the signal measured from the plectrum. An advanced and systematic (all cases) analysis with a series of experimentation made in goods conditions should made this augmented plectrum a critic interface for the extraction some guitarist gestures.

3.5 Gesture guidance

Figure 6. The head of the Xtar. One can see the two lines of sensors and the new nut supporting the guiding strings

The guitar offers to the guitarist many reference marks helping him to play on it. One finds of course visual marks but also tactile marks. Apart from its acoustic role, the string is a physical interface between the sound and the musician. It presents:

- A role of guide for the fingers of the non-preferred hand

- A dynamic reaction due to its tension (a force feedback)

One can emit the hypothesis that the friction of the finger on the string during a glissando is quantifiable by the musician and brings additional information to him. In the first version of our prototype, the contact of the fingers with the fingerboard was done without intermediary mean. The fingers were thus in direct contact with sensors stuck on the plane surface of the fingerboard. The instrument was playable, functioned correctly, but was rather not very stimulant to play. Indeed the epistemic role [8] of the string seems essential in order to take advantage of the guitarist knowledge. It appears like the minimum to exploit the motor behaviors. That's the reason why strings, without acoustic role, were thus installed like simple guides. Not only the instrument became more pleasant to play, but moreover control became more precise. It seems that a major explanation is that the tension of the string also provides a force feedback perpendicularly to the fingerboard, which is the source of a control loop (action/perception) more efficient in this case. Technically another explanation of this fact can be that the size of the contact point on the sensor is smaller, and so it improves the precision of the gesture measurement.

3.6 Sound synthesis

For this realization, a MIDI software synthesizer was used to generate the sound (fig 7). MIDI codes were generated starting from the mapping, were made in Max/MSP, between the data measured by the sensors and the control parameters of the MIDI synthesizer. That made it possible to focus only on the design and the part of the mapping dedicated to the interpretation of the gestures. Obviously certain gaps of the MIDI protocol have been problematic. We solved these problems in different ways. One of them was to use in parallel two instance of the software synthesizer. The major problem was to be able to send a continuous control of pitch for the two strings. Indeed, with MIDI it means that each string has to be on a different MIDI channel if possible or connected to different synthesizers.

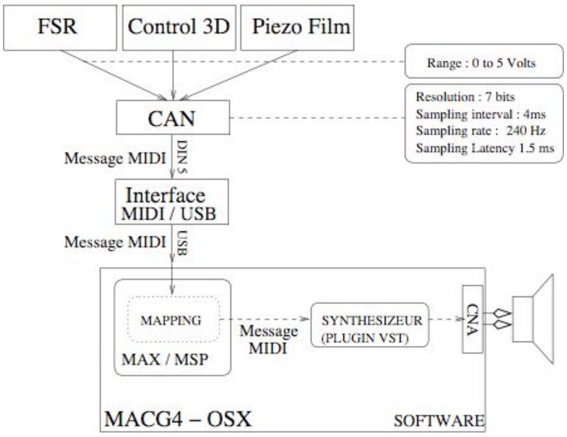


Figure7. Global Organization of the GXtar

4. MUSICAL USE OF THE GXTAR

4.1 Playing techniques

The GXtar allows 3 major playing configurations. First, one can exploit the 'tapping' technique with the 2 hands. Using only the fingerboard one can play in a percussive or rhythmic way, fast articulated phrases as in the tradition of the 'tapping'. The second configuration is to use only one hand for tapping and to use the other for spectral modulation with the 3 DOF controller (filtering, navigation in timbre space, other modulation). The performer can in this case sculpt the sound material. The third major configuration is to use a playing technique similar to the conventional guitar playing: selecting the pitch on the fingerboard and exciting the string with the plectrum. But several other hybrid configurations can be possible. It is, in fact an interesting field of exploration.

4.2 Comparison with related instruments

The first obvious improvement concerns the fact that guitar playing techniques related to modulation of pitch has been conserved in the design of this instrument (Hammer-on, pulling-off, glissando). At the same time the use of the tapping technique associated to the one of the 3 degrees of freedom controller allows new musical possibilities. Even if it's not so usual in guitar instrument (a little more or bass guitar) the use of fretless style is also an interesting possibility, in particular for non-tempered music.

4.3 Other sensors possibilities

A new generation of sensors, or a new strategy to use existing sensors could improve considerably the design of such instrument, especially concerning the fingerboard. The issue could be the same as other multi-point pressure sensitive control surfaces, or one could have a new approach specific to guitar or other related instrument. In this case FSR technology could just be used with a different geometry.

4.4 Learning, skills and training

Nevertheless, the efforts of training are undoubtedly still necessary to fully reach the new possibilities that the instrument offers. These efforts required, which are less important than in the case of the training of a completely new instrument, can be justified partly by a new organization of the motor tasks.

5. Conclusion

The new instrument presented in this paper is not an imitative instrument, e.g., it's not supposed to imitate a conventional instrument. It's even not exactly an equipped instrument, e.g. a conventional instrument equipped with sensors to bring new possibilities while preserving the usual playing and sound generation (acoustic or electric). It's more an instrument with a design inspired by a conventional instrument, which can, in a way be considered as a subgroup of alternate instrument. The role of this new instrument is to give new possibilities in electronic music to performers who have already developed skills in guitar playing. The FSR technology gives satisfying results for this kind of application and could probably bring more possibilities with more adapted size. The new organization of motor tasks seems also pleasant and adequate, but of course learning and practice certainly improve the musical use.

REFERENCES

[1] Diana Young, 'The Hyperbow Controller: Real-Time Dynamics Measurement for Violin Performance', Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002

[2] Caroline Traube, Philippe Depalle, Marcelo M. Wanderley, 'Indirect Acquisition of Instrumental Gesture Based on Signal, Physical and Perceptual Information'. NIME 2003: 42-48

[3] F. R. Moore. The dysfunction of midi. Computer Music Journal, 12(1): 19–28, 1988.

[4] StarrLabs. Starr labs midi guitars and custom midi controllers. http ://www.starrlabs.com.

[5] Robert Huott 'An Interface for Precise Musical Control', Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002.

[6] See Divided Pickup Roland GK-(http://www.rolandus.com) or Yamaha G1D (http://www.yamaha.com)

[7] Suguru Goto and Takahiko Suzuki. 'The case study of application of advanced gesture interface and mapping interface, virtual musical instrument "superpolm" and gesture controller "bodysuit" '. In Proceedings of NIME-04, Hamamatsu, Japan, June 3-5 2004.

[8] Claude Cadoz, Le geste canal de communication homme / machine, la communication instrumentale, *L'interface des mondes reels & virtuels*, pp. 1-31, Montpellier France, September 8 199

Visual Methods for the Retrieval of Guitarist Fingering

Anne-Marie Burns
Input Devices and Music Interaction Lab
Schulich School of Music, McGill University
555 Sherbrooke Street West, Montréal, Québec, Canada, H3A 1E3
anne-marie.burns@mail.mcgill.ca

Marcelo M. Wanderley
Input Devices and Music Interaction Lab
Schulich School of Music, McGill University
555 Sherbrooke Street West, Montréal, Québec, Canada, H3A 1E3
marcelo.wanderley@mcgill.ca

ABSTRACT

This article presents a method to visually detect and recognize fingering gestures of the left hand of a guitarist. This method has been developed following preliminary manual and automated analysis of video recordings. These first analyses led to some important findings about the design methodology of a vision system for guitarist fingering, namely the focus on the effective gesture, the consideration of the action of each individual finger, and a recognition system not relying on comparison against a knowledge base of previously learned fingering positions. Motivated by these results, studies on three aspects of a complete fingering system were conducted: the first on finger tracking; the second on strings and frets detection; and the last one on movement segmentation. Finally, these concepts were integrated into a prototype and a system for left hand fingering detection was developed.

Keywords

gesture, guitar fingering, finger-tracking, Hough transform, line detection, gesture segmentation

1. INTRODUCTION

Fingering is an especially important aspect of guitar playing, as it is a fretted instrument where many combinations of string, fret, and finger can produce the same pitch. Fingering retrieval is an important topic in music theory, music education, automatic music generation and physical modeling. Unfortunately, as Gilardino noted [6] [7], specific fingering information is rarely indicated in scores.

Fingering information can be deduced at several points in the music production process. Three main strategies are:

- Pre-processing using score analysis;
- Real-time using Midi guitars;
- Post-processing using sound analysis;

Radicioni, Anselma, and Lombardo [10] retrieve fingering information through score analysis. The score is fragmented in phrases, and the optimum fingering for each phrase is determined by finding the shortest path in an acyclic graph of all possible fingering positions. Weights are assigned to each position based on a set of rules. The problem with this approach is that it cannot account for all the factors influencing the choice of a specific fingering, namely philological analysis (interpretation of a sequence of notes), physical constraints due to the musical instrument, and biomechanical constraints in the musician-instrument interaction. Outputs of these systems are similar to human solutions in many cases, but hardly deal with situations where the musical intention is more important then the biomechanical optimum fingering.

Other systems retrieve the fingering during or after a human plays the piece. One of these approaches uses a Midi guitar. Theoretically, using a Midi guitar with a separate Midi channel assigned to each string, it is possible to know in real-time what pitch is played on which string, thus determining fret position. In practice however, Midi guitar users report several problems, including a variation in the recognition time from one string to another and the necessity to adapt their playing technique to avoid glitches or false note triggers [13].

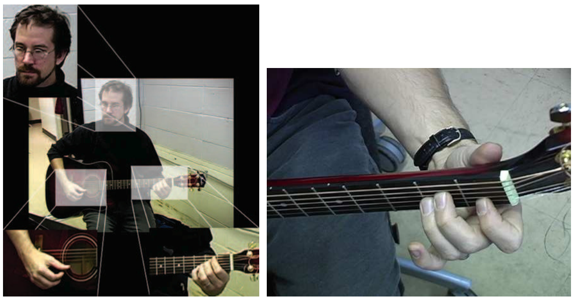
An approach using the third strategy is the study of the guitar timbre. Traube [11] suggested a method relying on the recording of a guitarist. The method consists of analyzing the sound to identify the pitch, find the plucking point and then determine the string length to evaluate the fingering point. Shortcomings of this method are that it cannot be applied in real time, it works only when one note is played at the time, and error of the string length evaluation can be as high as eight centimeters in the case of fretted strings [12].

This paper presents an alternative method for real-time retrieval of the fingering information from a guitarist playing a musical excerpt. It relies on computer analysis of a video recording of the left hand of the guitarist. The first part of this article discusses the preliminary manual and automated analyses of multiple-view video recordings of a guitarist playing a variety of musical excerpts. The subsequent sections present studies on three aspects of visual analysis of a guitarist fingering: finger tracking, string and fret detection, and movement segmentation. Finally a system integrating these three components is presented.

NIME06, June 4-8, 2006, Paris, France

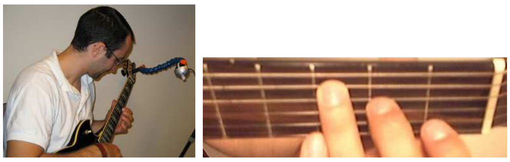
2. PRELIMINARY ANALYSIS

During the preliminary analysis, different camera views were evaluated (global view, front view, and top view). The



(a) Global view with zooms (b) Top view of the left hand

Figure 1: Two different views of a guitarist playing captured from a camera on a tripod placed in front of the musician: (a) Global view with zoom on different zones for gesture analysis: facial expression and front view of right and left hands. (b) Top view of the left hand.



(a) Camera mount (b) Camera view

Figure 2: Depiction of the guitar camera mount that was used to eliminate the ancillary gesture problem: (a) The camera mount installed on an electric guitar. (b) The camera on a classical guitar. In this example, the camera is placed to capture the first five frets.

aim was to find a viewpoint that allows the retrieval of the most information possible with the desired degree of accuracy and precision.

The top view (figure 1(b)) was retained for its interesting characteristics with respect to the problem, namely a detailed view of the fingers, the possibility for string and fret detection, and the ability to observe finger-string proximity. However, slow motion observations of the video recording showed that the neck is subject to many ancillary movements. Preliminary automated tests have shown that this type of movement can influence the computer's capacity to correctly identify fingering. Consequently, the tripod was replaced by a camera mount on the guitar neck (figure 2). The preliminary automated fingering recognition tests were performed by comparing two top view recordings of a musician playing musical excerpts against top view images of previously recorded chords played by the same performer stored in the form of Hu moments vectors [8]. These tests allowed to identify three main issues:

1. Using an appearance base method limits the system to previously learned material.
2. Using the global shape of the hand limits the system to the recognition of chords.
3. Using a knowledge base makes the recognition time grow with the knowledge base size.

From the above issues, the main specifications for a fingering recognition system are:

1. Focus on effective gestures by further reducing the presence of ancillary movements and background elements.
2. Use of a representation that considers the action of individual fingers.
3. Use of a recognition mechanism that eliminates the burden of a knowledge base and that is therefore not limited to previously learned material.

The first specification can be achieved using the guitar mount as presented in figure 2. In order to fulfill the other specifications, three studies were conducted. In a first study, the circular Hough transform was chosen to perform finger-tracking. The second study examined the use of the linear Hough transform for string and fret detection, and a third one explored movement segmentation.

3. FINGER-TRACKING

The circular Hough transform algorithm used in this paper was developed and implemented in EyesWeb [2]. It presents the following interesting characteristics:

1. It demonstrated to have a high degree of precision and accuracy;
2. It can be applied in complex environments and with partial view of the hand;
3. It can work on edge versions of the images.

3.1 Circular Hough Transform

As illustrated in figure 3, the circular Hough transform [3] is applied on the binary silhouette image of the hand. The edge-image is obtained by applying the Canny edge detection algorithm [4] on the silhouette images. The circular Hough transform algorithm makes use of the fact that finger ends have a quasi-circular shape while the rest of the hand is more linearly shaped. In this algorithm, circles of a given radius are traced on the edge-images and regions with the highest match (many circles intersecting) are assumed to correspond to the center of fingertips.

4. STRING AND FRET DETECTION

By tracking the fingertips it is possible to know where each finger is in space. In the case of guitar fingering, this space can be defined in terms of string and fret coordinates. Prior to the detection stage, the region of interest (in that case the guitar neck) must be located in the image. Once the neck has been located, the strings and frets are segmented from the grayscale neck image by applying a threshold. A vertical and a horizontal Sobel filter are applied on the threshold image in order to accentuate the vertical and horizontal gradients. A Linear Hough Transform [3] is then computed on the two Sobel images. The linear Hough transform allows detection of linearity in a group of pixels, creating lines. These lines are then grouped by proximity in order to determine the position of the six strings and of the frets. Once this is done, it is possible to create a grid of coordinates to which fingertip positions will be matched.

5. MOVEMENT SEGMENTATION

Movement segmentation is essential in order to detect fingering positions and not simply track fingertips during the

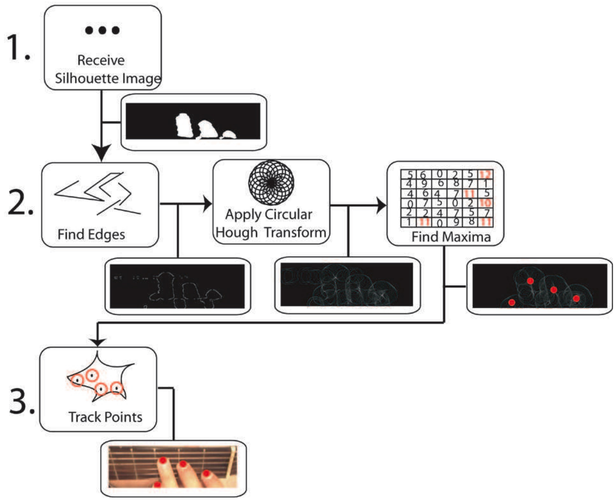


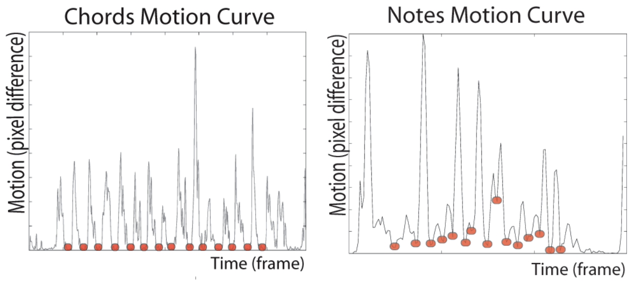
Figure 3: Fingertip detection using the circular Hough transform algorithm

playing sequence. Furthermore, in order to save computer resources, this segmentation must be done early in the global process so that subsequent analysis steps are not performed unnecessarily. Movement segmentation is used to separate the nucleus phase of the gesture from the preparation and retraction phase [9].

In the preliminary analysis, movement segmentation was done by applying a threshold on the motion curve (figure 4 a) generated by the computation of the pixel difference between each frame. The characteristic lower velocity phase of the nucleus was easily detected between each chord. However, in other playing situations, such as when playing a series of notes, the separation between the movement transitory phases and the nucleus is not that clear (figure 4 b). This is due to a phenomenon called *anticipatory placements of action-fingers* that has been studied in violin [1] and piano [5]. In these cases, the preparation phase of other fingers occur during the nucleus of the action-finger. Thus the motion is not serial and consequently, the global motion curve does not exhibit clear global minima like it is the case for chords. However, local minima can still be observed and detected as they can be assumed to correspond to the moment the note is trigged by the right hand. Local minima are found by computing the second derivative of the motion curve. As the prototypes work in real-time, this is done by subtracting the signal with its delayed version twice.

6. PROTOTYPE

The prototype was designed to fulfill the requirements for a fingering recognition system highlighted by the preliminary analysis. The focus on effective gestures is partially realized at the hardware level by affixing the camera to the guitar neck, thereby eliminating the motion of the neck caused by the ancillary gesture. Elimination of background elements is done by selecting a strict ROI (Region of Interest) around the neck and by applying a background subtraction algorithm on the image. Movement segmentation is performed by finding minima in the motion curve, obtained by computing the difference of pixel between each



(a) Chords motion curve (b) Notes motion curve

Figure 4: (a) Motion curve of a guitarist playing chords (b) Motion curve of a guitarist playing a series of notes

frame. The action of each individual finger is considered using the finger-tracking algorithm described above. The details of the algorithm are shown in figure 5.

During preliminary tests, the prototype was able to correctly recognize all fret positions. Due to the chosen camera view, the space between the strings is smaller for the high strings (E, B, G) than for the low strings (D, A, E), therefore affecting the accuracy of the recognition system. As demonstrated in [2], the circular Hough transform has an accuracy of 5 +/- 2 pixels with respect to the color marker references. The resolution of the camera used in this prototype is 640x480 pixels, therefore giving a 610x170 pixels neck region. The distance in pixels between the first and second string is of 12 pixels at the first fret and 17 at the fifth fret. Between the fifth and sixth strings, the distance in pixels is 16 and 20 pixels for the first and fifth fret, respectively.

Since the chosen algorithm attributes the string position to the finger by proximity, in the worst case the finger-tracking algorithm error exceeds half the space between the higher strings, therefore confusion happens. However, since this problem does not happen with lower strings were the distance between two strings is greater, the problem could be solved with an higher resolution camera. Another limitation is that in the current system only the first 5 frets are evaluated, but this could be solved with a wide angle camera. One problem that cannot be easily solved by changing the hardware is finger self occlusion. This problem only rarely happens, but exists in the case of fingerings were two fingers play at the same fret, for example in the case of C7 and Dm7. In future developments, this problem could potentially be solved by estimating the fingertip position using the finger angle.

7. CONCLUSIONS

This article discussed new strategies to capture fingering of guitarists in real-time using low-cost video cameras. A prototype was developed to identify chords and series of notes based on finger-tracking and fret and string detection. It recognizes fingerings by matching fingertip positions to the strings and frets' grid of coordinates, therefore not relying on any knowledge base. Results of the prototype are encouraging and open possibilities of studies on many aspects of a guitarist instrumental gesture, namely gesture segmentation, anticipatory movements, and bimanual synchronization. Applications of this research include automatic chord transcription, music education, automatic music generation

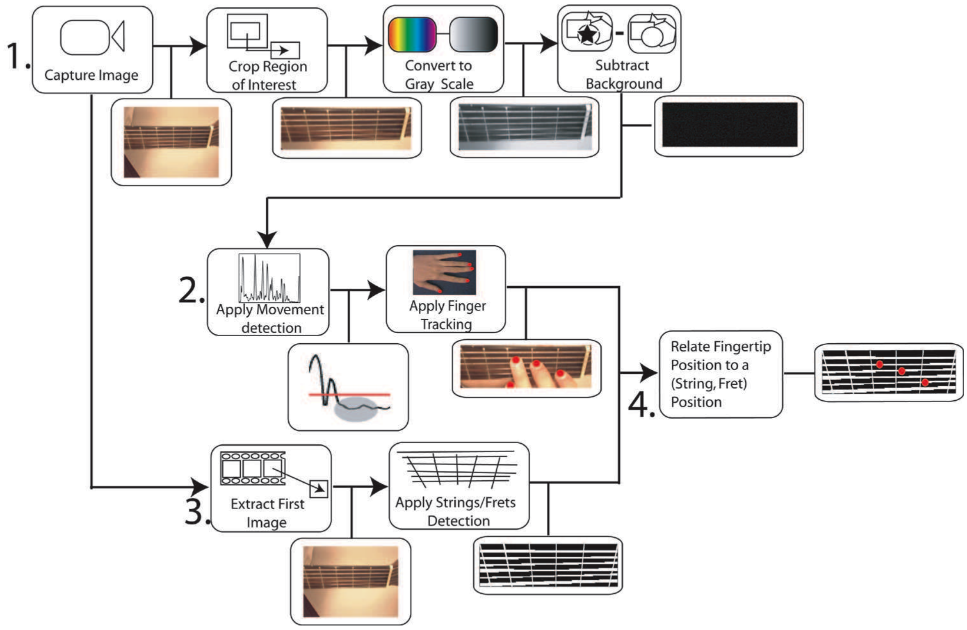


Figure 5: Prototype - algorithm

and physical modeling.

8. ACKNOWLEDGMENTS

The study on finger-tracking was realized at InfoMus laboratory, D.I.S.T., Universitá degli studi di Genova and has been partially supported by funding from the Québec Government (PBCSE), the Italian Ministry of Foreign Affairs, and by the EU 6 FP IST ENACTIVE Network of Excellence. The first author would like to thank all students and employees of InfoMus lab who "lent a hand" for the realization of the tests. Special thanks go to Barbara Mazzarino, Ginevra Castellano for her help in compiling the results, Gualtiero Volpe for his contribution to the development of the EyesWeb blocks. Anne-Marie Burns would also like to thank Antonio Camurri for welcoming her as an internship student researcher.

The authors would like to thank the guitarists that participated in the tests: Jean-Marc Juneau, Riccardo Casazza, and Bertrand Scherrer. Thanks also to Mr. Donald Pavlasek, Electrical and Computer Engineering, McGill University, for the conception and implementation of the camera guitar mount.

9. REFERENCES

[1] A. P. Baader, O. Kazennikov, and M. Wiesendanger. Coordination of bowing and fingering in violin playing. *Cognitive Brain Research*, 23:436–443, 2005.

[2] A.-M. Burns and B. Mazzarino. Finger tracking methods using eyesweb. In *S. Gibet, N. Courty and J.-F. Kamp (Eds)., Gesture Workshop 2005 Proceedings, LNAI 3881*, pages 156–167, 2006.

[3] A.-M. Burns and M. M. Wanderley. Computer vision method for guitarist fingering retrieval. In *Proceedings of the Sound and Music Computing Conference 2006.* Centre National de Création Musicale, Marseille, France, May 2006.

[4] J. A. Canny. Computational approach to edge detection. *, IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[5] K. C. Engel, M. Flanders, and J. F. Soechting. Anitcipatory and sequential motor control in piano playing. *experimental Brain Research*, 113:189–199, 1997.

[6] A. Gilardino. Il problema della diteggiatura nelle musiche per chitarra. *Il "Fronimo"*, 10:5–12, 1975.

[7] A. Gilardino. Il problema della diteggiatura nelle musiche per chitarra. *Il "Fronimo"*, 13:11–14, 1975.

[8] S. Paschalakis and P. Lee. Pattern recognition in grey level images using moment based invariant features image processing and its applications. *IEE Conference Publication*, 465:245–249, 1999.

[9] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[10] D. Radicioni, L. Anselma, and V. Lombardo. A segmentation-based prototype to compute string instruments fingering. In *Proceedings of the Conference on Interdisciplinary Musicology*, Graz, 2004.

[11] C. Traube. *An Interdisciplinary Study of the Timbre of the Classical Guitar*. PhD thesis, McGill University, 2004.

[12] C. Traube and J. O. Smith III. Estimating the plucking point on a guitar string. In *Proceedings of the COST G-6 Conference on Digital Audio Effects*, Verona, Italy, 2000.

[13] J. A. Verner. Midi guitar synthesis yesterday, today and tomorrow. an overview of the whole fingerpicking thing. *Recording Magazine*, 8(9):52–57, 1995.

Combining accelerometer and video camera: Reconstruction of bow velocity profiles

Erwin Schoonderwaldt
CSC
KTH
Lindstedtsv. 24
S-100 44 Stockholm
(+46) 8 790 7561
schoondw@speech.kth.se

Nicolas Rasamimanana
IRCAM
CNRS STMS UMR 9912
Real Time Applications Team
1, Place Igor Stravinsky
75004 Paris
(+33) 1 44 78 48 26
Nicolas.Rasamimanana@ircam.fr

Frédéric Bevilacqua
IRCAM
CNRS STMS UMR 9912
Real Time Applications Team
1, Place Igor Stravinsky
75004 Paris
(+33) 1 44 78 48 31
Frederic.Bevilacqua@ircam.fr

ABSTRACT

A cost-effective method was developed for the estimation of the bow velocity in violin playing, using an accelerometer on the bow in combination with point tracking using a standard video camera. The video data are used to detect the moments of bow direction changes. This information is used for piece-wise integration of the accelerometer signal, resulting in a drift-free reconstructed velocity signal with a high temporal resolution. The method was evaluated using a 3D motion capturing system, providing a reliable reference of the actual bow velocity. The method showed good results when the accelerometer and video stream are synchronized. Additional latency and jitter of the camera stream can importantly decrease the performance of the method, depending on the bow stroke type.

Keywords

Bowing gestures, bowed string, violin, bow velocity, accelerometer, video tracking.

1. INTRODUCTION

Accelerometers and standard video camera are two different types of widely used sensors in the design of cost-effective gesture capture systems. In particular, such sensors have been incorporated in several musical interfaces. Each of these types of sensor has different characteristics. First, accelerometers are typically used to build miniature low latency systems. They are for example particularly well suited to capture percussive gestures. Nevertheless, quantitative use of accelerometer might be difficult due to the fact that the signal depends on both the tilt angle and the actual acceleration. Second, video cameras are well suited to localize, and to spatially follow object. Nevertheless standard video rate are relatively slow for musical application and important latencies are difficult to avoid.

This very brief description points out that accelerometers and video camera are actually complementary systems [8]. Moreover, as discussed by Foxlin [7], there has been a growing interest in the field of Augmented Reality to combine both inertial systems (accelerometers and/or gyroscopes) and vision systems to perform efficient tracking. Generally, the inertial component is fixed to the camera.

We report here a simple approach combining accelerometers with a standard camera to capture bowing gestures. Such a combination is interesting in this case since bowing gesture contains fast and slow temporal features, as well as small and large spatial displacements. As a matter of fact, in bowed string instruments such as the violin, bowing gestures form an essential part in the tone production, giving the player a continuous, yet complex, control of the produced sound.

The capture of bowing gestures have shown important interests in various contexts. Bowing measuring systems can be either used in *non realtime* for fundamental studies of violin playing or as gestural interfaces to control in *realtime* various digital sound processes.

Concerning fundamental studies, the first detailed measurements of bowing parameters were performed by Askenfelt [1, 2], using a bow equipped with sensors for measuring bow force, bow position and bow-bridge distance. Knowledge of how players make use of these parameters provide an important key to violin performance, which could for example be useful for controlling physical models of bowed string instruments or applications in music education.

The *Hyperbow* [6], or more recently the *Augmented violin* [3, 5] are two examples, among others, of modified bows used in live performance. In these particular cases, accelerometers are placed on the bow. The acceleration signal can be used to detect bow stroke and in some cases to recognize bowing styles [5].

Bow velocity is an important parameter to characterize bowing, and is one of the most important input parameters for playing a physical model of the violin. Several acoustic studies ([1, 2]) have also clearly shown the relationship between velocity and sound quality. Nevertheless, bowing velocity can be difficult to measure accurately in a playing situation. Velocity can potentially be derived from video tracking or accelerometer signals. Difficulties arise in both cases:

-Computing accurate velocity profile from video tracking system generally requires the reconstruction of the bowing orientation in 3D space. Moreover, expensive camera for high temporal resolution is generally necessary for fast movements such as bowing attacks. The use of systems such as 3D motion capture system is generally limited to the laboratory environment.

- Computing velocity over a longer time span by integration of the accelerometer signal is problematic. For example, the accelerometer used in the augmented bow [3, 5] is sensitive to both inclination and acceleration (generally referred to as static and dynamic acceleration), which means that there is a variable amount of drift present in the integrated signal.

The method described here, combining the use of both an accelerometer and standard video, allow us to overcome such problems. In particular, we present a method compensating for this drift, enabling therefore the reconstruction of bow velocity from the bow acceleration signal. First, we explain the general principles of the method (section 2). In section 3 the method is assessed using data obtained with a 3D motion capture system, to evaluate the potentiality and limitations of the method. The results are discussed in section 4.

2. BOW VELOCITY RECONSTRUCTION

We first explain the general principle of the setup and reconstruction method. Second, we describe a particular pilot study that serves as a proof-of-principle example.

2.1 Setup

The setup is shown in Fig.1. The two main components are a fixed camera and an accelerometer placed on the bow. The accelerometer axis is set to be parallel to the bow axis. Standard video processing techniques are used to track the bow movements. For example, two color markers can be placed on the bow enabling robust tracking. The accelerometers can be part of the wireless modules described in reference [5] and [6]. Note that simpler implementation is also possible using wired connection between the accelerometer and a sensor A/D interface.

Two angles in this setup are important to consider, both varying significantly during the playing. First, α is the angle between the bow and the vertical direction. The variation of α is the main responsible for the drift in the acceleration signal. Second, β is the angle between the axis of the camera and the direction perpendicular to the bow. The relation between the velocity along the length-axis of the bow and the velocity observed by the camera is then characterized by a scaling factor $\cos(\beta)$.

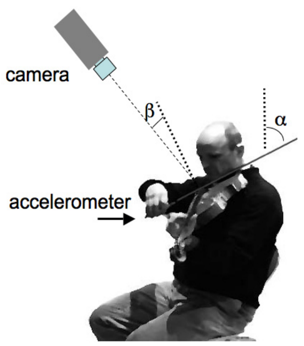


Fig. 1. Setup.

2.2 Computation

As already mentioned, velocity computed from the accelerometer signal typically contains a drift over time. The reconstruction method allows for the compensation of this drift in the integrated signal, by detecting moments when bow direction change. At such moments the bow velocity is equal to zero (referred below as "breakpoints").

Using such information, the accelerometer signal is piece-wise integrated between these "breakpoints". After this step, the velocity curve shows discontinuities at the "breakpoints", which can be removed making an assumption on the form of the acceleration drift. The simplest assumption corresponds to a constant drift in the acceleration signal between two breakpoints, corresponding to a constant α during a bow stroke. This constant drift in the acceleration produces piece-wise constant slopes in the integrated signal. Such linear trends can be simply computed and removed, resulting in a velocity profile that is continuous at the breakpoints.

To obtain the zero-crossings of the bow velocity, a simple video camera can be sufficient. Key points on the bow and the violin are tracked using video processing techniques in order to detect the moments when the bow changes direction. Errors induced by the camera position and the low frame rate of standard video camera are addressed in section 3.

The reconstructed velocity can potentially have a high spatial and temporal resolution (depending on the accelerometer and on A/D conversion system). Therefore, the method offers an easy-to-implement and a cost-effective alternative to expensive commercial motion capture systems to obtain bow velocity signals in violin playing.

The next section describes the implementation of the reconstruction method in a pilot experiment, using a normal video camera and the augmented bow.

2.3 Pilot experiment

A pilot experiment was performed to test the feasibility of the velocity reconstruction method. The bow acceleration was measured with the augmented bow, developed at IRCAM [3, 5]. The bow was equipped with two Analog Device ADXL202 acceleration sensors at the frog, and the acceleration data was sent wirelessly to a RF receiver, connected to a sensor acquisition system, Ethersense [4]. The acceleration data was digitized on 16 bits at the frame rate of 500 Hz.

The video data was obtained with a Sony digital handcam (type DCR-TRV245E). For the measurements reported in Figure 2, two points were marked using differently colored pieces of fabric: one attached to the curl of the violin and the other to the wrist of the player's bowing arm. The camera was positioned in front on the right side of the player, so that the marked points were visible, and the bow motion could be clearly observed (the influence of such setup configuration will be discussed in section 3). The color markers were tracked using Eyesweb software [9], by selecting the pixels with the specified colors and calculating the centre of gravity of the observed pixel regions. The frame rate of the video data was 25 Hz.

The acceleration and video data were synchronized by aligning two synchronization events at the beginning and the end of the recording. The synchronization features were obtained by

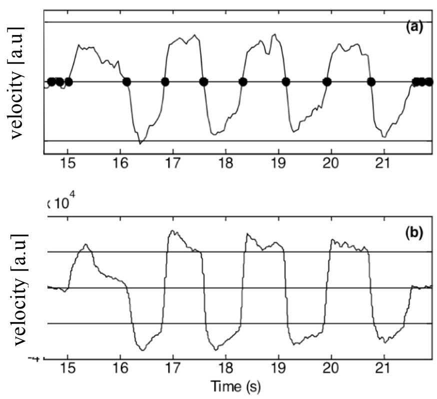


Fig. 2. Détaché bowing on one string. Velocity signal derived from video data (a). The detected zero-crossings are indicated by dots. Reconstructed velocity signal (b) obtained by piece-wise integration of the acceleration of the bow.

tapping with the bow on the curl of the violin, which left a measurable trace in both data series.

From the video data, the position of the wrist marker was obtained in the camera reference. The motion of the violin marker was subtracted to obtain the relative motion of the wrist marker. The derivative was computed and smoothed using Savitzky-Golay filtering (order 5, frame size 11). The zero-crossings of this velocity signal were detected and refined using linear interpolation to obtain a more precise time estimate. Figure 2a shows a series of recorded *détaché* bowing styles: a single tone was repeatedly played to avoid string crossings.

In the next step, the velocity was reconstructed from the acceleration signal of the bow, shown in Figure 2b. The reconstructed velocity profile is coherent as there is an equal repartition between positive and negative velocity, which is in accordance with the performed movement. The reconstructed velocity data has a temporal resolution equal to the accelerometer sampling rate, i.e. 500 Hz. In comparison, the sampling rate of the velocity profile from the video data is only 25 Hz. Moreover, Figure 2 shows that the velocity from the video data is significantly noisier than the velocity profile reconstructed from the accelerometer.

3. METHOD ASSESSMENT

The velocity reconstruction method was further evaluated using 3D motion capture data. During this experiment the bow acceleration was measured simultaneously to the motion of the bow and the violin. The motion capture data provided a reliable reference signal of the actual bow velocity, which was used for quantitative comparison with the reconstructed bow velocity.

The evaluation of the method was performed on three aspects. First, velocity reconstruction is evaluated quantitatively, using an optimal velocity signal. Second, the influence of the video camera position is addressed. Last, the influence of video latency and jitter is quantified.

3.1 Motion capture setup

The motion of the violin and the bow was tracked using a Vicon MX system, with 12 cameras at the frame rate of 150 Hz. The motion capture data was smoothed using Savitzky-Golay filtering (order 3, frame size 9).

Bow acceleration was measured with the same augmented bow described in section 2.3. The sample rate of the acceleration data was 500 Hz. Acceleration data was smoothed using Savitzky-Golay filtering (order 3, frame size 25 – corresponding with the mocap smoothing parameters in the time domain). The data selected for the evaluation was a recording of scales with different bowing styles: *détaché*, *martelé* and *spiccato* played at 60 bpm.

3.2 Velocity reconstruction validation

We first validated the velocity reconstruction method in the case where the velocity zero-crossing were accurately known from the motion capture data, in order to show the achievement of the method under optimal conditions. As the reference signal the velocity along the length-axis of the bow was taken, which corresponds with the actual bow velocity at the string. The reference signal was computed from the motion capture data, compensating for the motion of the violin. During the first validation step the zero-crossings of the reference velocity were used as breakpoints for the reconstruction.

Figure 3 (top) shows the acceleration signal of the bow during a *détaché* scale. It clearly demonstrates the influence of angle α on the acceleration signal, the different offset levels corresponding to playing on different strings.

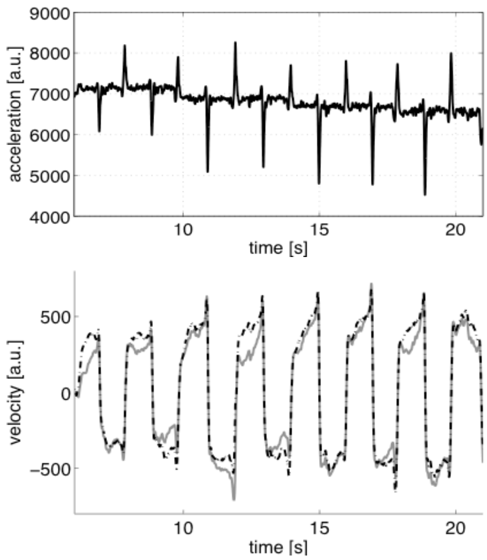


Fig. 3. (top): accelerometer signal during a *détaché* bowing. (bottom): reference velocity (dotted) and reconstructed velocity (plain)

Figure 3 (bottom) shows the reconstructed bow velocity, as well as the bow velocity reference for a series of *détaché* bow strokes using the reference signals zero-crossings as breakpoints. It shows that the reconstructed velocity is in relatively good agreement with the reference velocity. Note that similar results were also found for the other bow stroke types. For example, the correlation coefficients between the reference and the estimated velocities are 0.984 for *détaché*, 0.998 for *martelé* and 0.987 *spiccato* (computed on ascending and descending scale at 60 bpm). These high correlation values demonstrate the validity of the method, when the velocity zeros are accurately determined.

3.3 Influence of the camera viewpoint

The video camera implies a 2D projection of the markers movement. Such a 2D projection is in most cases sufficient since we are not interested in the actual velocity profile from the video data, but only in the determination of the zero-crossing. The most important point is thus to guarantee a sufficiently a high resolution image of the bow movements. Poor resolution can lead to important errors in the determination of the velocity zero-crossing. The optimal viewpoint is therefore the one providing the best overall resolution of the bow displacement.

As mentioned in section 2.1, the projected bow velocity depends on the angle of the camera with the bowing direction with a scaling factor of $cos(\beta)$. For an optimal projection the video camera should therefore be placed perpendicular to the length-axis of the bow, for example above the player (see Fig. 1).

3.4 Influence of video latency and jitter

Latency and jitter between the accelerometer and video streams could occur, especially if no synchronization mechanism is operated. Such phenomena will mainly affect the timing of detected zero-crossings. In this section, we evaluate the effect of altering such timing in the reconstruction velocity.

An increasing latency was added to the reference breakpoints. The latency varied from 0 to 80 ms by steps of 20 ms. An uniformly distributed random number of span 40 ms was added to simulate a jitter effect. The consequence on the correlation coefficient is shown in Figure 4.

Détaché showed a better robustness to jitter than the two others with a smaller variation. Nevertheless, when the latency is kept sufficiently low, the jitter is not a major source of error for the three tested bowing styles.

For *détaché*, the correlation coefficient decreased from 0.984 to 0.87 when latency increases to 40 ms. However, the correlation coefficient stabilized as the latency increases to 80 ms, dropping only to 0.84. For *martelé*, the method is less sensitive to small latencies. The correlation coefficient is higher than 0.85 for latencies less than 60 ms. S*piccato* appeared to be the more sensitive to latency/jitter effects as the correlation coefficient fell almost linearly.

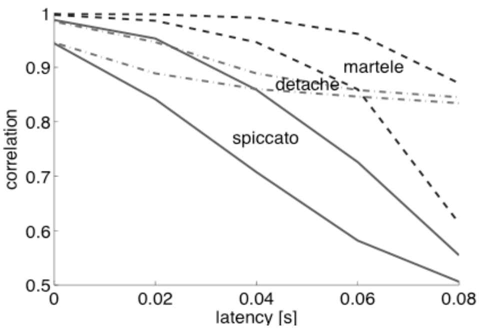


Fig. 4. Correlation factor according to latency and jitter for the three bowing styles. For each bowing style, the upper line is without simulated jitter and lower line with simulated jitter (40 ms).

3.5 Influence of the movement of the player

Except for the errors due to projection and synchronization issues, there are two other possible error sources associated with the reconstruction method. First, as already mentioned it was assumed that the drift between the breakpoints is constant, implying that the angle α in Fig.1 does not change. However, this assumption is not necessary valid in any playing situation, as the player can vary the angle of the bow, especially on the outer G and E strings. Second, the acceleration is measured relative to a fixed reference, rather than the moving violin. This means that movements of the player not directly related to playing are contributing to the reconstructed velocity as well. Thus, the achievement of the method could be dependent on the complexity of the bowing pattern, as well as the amount of additional (expressive) movements by the player.

4. DISCUSSION AND CONCLUSIONS

The overall results show that the method has good potential to reconstruct velocity profile with high temporal resolution. The method is simple to implement and cost-effective compared high-performance motion capture system.

This method could be useful for fundamental studies of bowing gesture, in particular cases where expensive motion capture system systems are not available or too cumbersome to use. The results shown here are promising, but further validation is required to fully characterize the precision and accuracy of this method.

The reconstruction method could also be used in live performance. Nevertheless, the implementation describe here cannot be used as a “strict” realtime system, in the sense that the reconstruction method implies an inherent variable delay. As a matter of fact, the accelerometer drift is not corrected continuously but at discrete time (i.e. velocity zero-crossing). Moreover, as shown in section 3.4, accurate results might require synchronizing the accelerometer and video streams, which would add an additional delay.

Nevertheless, the system can still be useful in performance situation, where detailed information of the bowing gesture is desired and a delayed response is manageable. In such cases, the reconstructed velocity profile can reveal to be very helpful, since the bow velocity is one of the fundamental parameter in the bowing gesture.

The method we described here could also be very valuable for pedagogical applications. For example, accurate information on the playing regularity of specific bow strokes could provide the students with helpful information. In such cases, the information is needed only after the playing.

5. ACKNOWLEDGEMENTS

This work was supported by the Swedish Research Council, project: Bowed String Analysis (621-2001-2537) and the Cost287-ConGas action. We are grateful to Stephane Donikian (IRISA) and Franck Multon (Université Rennes 2) for giving us access to their motion capture system and their precious help during the mocap sessions. We also thank Christophe Desjardins and Emmanuel Fléty for their helpful participations in this project.

6. REFERENCES

[1] Askenfelt, A. *Measurement of bow motion and bow force in violin playing*. J. Acoust. Soc. Am. 80 (4), pp. 1007-1015, 1986.

[2] Askenfelt, A. *Measurement of the bowing parameters in violin playing. II: Bow-bridge distance, dynamic range, and limits of bow force*. J. Acoust. Soc. Am. 86 (2), pp. 503-516, 1989.

[3] Bevilacqua, F., Rasamimanana, N., Fléty, E., Lemouton, S. & Baschet, F. The augmented violin project: case studies, composition and performance report. In *Proceedings of the Conference on New Instruments for Musical Expression*, NIME 2006.

[4] Fléty, E., Leroy, N., Ravarini, J. & Bevilacqua, F. Versatile sensor acquisition system utilizing network technology. In *Proceedings of the Conference on New Instruments for Musical Expression*, NIME, 2004

[5] Rasamimanana, N., Fléty, E., Bevilacqua, F. *Gesture Analysis of Violin Bow Strokes*. GW 2005, Lecture Notes in Artificial Intelligence 3881, S. Gibet, N. Courty, and J.-F. Kamp (Eds.), pp. 145–155, 2006.

[6] Young, D., The hyperbow controller: real-time dynamics measurement of violin performance. In *Proceedings of the 2002 conference on NewInterfaces for Musical Expression*, NIME, Dublin, 2002

[7] Foxlin, E., Naimark, L., "VIS-Tracker: A Wearable Vision-Inertial Self-Tracker". In IEEE Conference on Virtual Reality (VR 2003). Los Angeles, CA.

[8] Kwon, D. Y., Gross, M., Combining Body Sensors and Visual Sensors for Motion Training, Proceedings of ACM SIGCHI ACE 2005, pp. 94-101.

[9] http://www.eyesweb.org

Reflective Optical Pickup For Violin

Nicolas Leroy
Performing Arts Research Team
IRCAM – 1, place Igor Stravinsky
75004 PARIS - FRANCE
+33/1 44 78 42 31
nicolas.leroy@ircam.fr

Emmanuel Fléty
Performing Arts Research Team
IRCAM – 1, place Igor Stravinsky
75004 PARIS - FRANCE
+33/1 44 78 15 49
emmanuel.flety@ircam.fr

Frederic Bevilacqua
Performing Arts Research Team & Real Time Applications Team
IRCAM – 1, place Igor Stravinsky
75004 PARIS - FRANCE
+33/1 44 78 48 31
frederic.bevilacqua@ircam.fr

ABSTRACT

We present here the development of optical pickup for acoustic violin. Unlike other optical pickups, this one works in a reflective mode, which is potentially less intrusive. This pickup, aimed principally at pitch tracking, uses a modulation technique to improve the signal-to-noise ratio, and limit artifacts of the ambient light.

Keywords

Violin, pitch tracking, optics, pickup, optical microphone

1. INTRODUCTION

The research presented here is a part of our on-going projects on "augmented instruments", and in particular a part of the "augmented violin" project [1]. This project concerns the development of various gesture capture systems for the violin (initially inspired, from a technology point of view, by the work at MIT on *hyper-instruments* [2]). Previously, we reported on a capture system for bow dynamics [3][4] that has been used in live performances.

One of the project's goals is to propose solutions compatible with any acoustic violin, avoiding any type of alteration (including on the acoustic sound). Recent developments do not take into consideration this constraint and therefore cannot be used in this context.

Here we report on developments concerning pitch tracking, which is an important feature for controlling various sound processes or performing score following. Pitch determination is generally difficult on violin, especially when polyphonic playing is used. Operating pitch tracking for each string separately can greatly simplify this task. Moreover, there is strong interest in capturing string vibration directly to avoid various acoustic effects due to the violin's resonance and other external sounds.

2. RELATED WORKS

Axon [5], Roland [6] and recently Gibson [7] have developed guitar and bass pickups and electronic systems that enable pitch tracking. A magnetic pickup is mounted under each metal string and each audio signal that comes from the strings can be converted into MIDI or USB messages.

However, in the case of the violin, such pickups cannot be used because the strings are not made from a magnetic alloy (the E-string is often made of a magnetic alloy but there is no rule for the others). Piezo pickups [8], inserted in the bridge, are an alternative solution but this implies either replacing the original bridge or integrating the pickups directly in the violin, as it is the case for Zeta violins [9]. This solution cannot be put in place with our constraints, where musicians use their own instruments that cannot be altered in any way.

Another solution is the optical pickup. In this case, a light beam is sent to the string and is modulated by the string vibration. The resulting light is measured by a photo diode or a phototransistor and can be electronically converted into an audio signal. LightWave Systems [10] and Ron Hoag [11] already sell guitars and basses that use optical pickups. Concerning the violin, Dan Overholt [12] developed high quality pickups for his "Overtone Violin".

Nevertheless, all these cases make use of optical pickups in a transmission mode, i.e. the string is located between the beam emitter and the receiver. This implies a setup that reduces the playing capabilities. Moreover, this configuration is difficult to install without making important modifications to the violin structure (placing them on the bridge for example, as it is the case for the Overtone violin).

An elegant solution makes use of optical pickups in a reflective mode, where the emitter and receiver are placed on the same side of the strings. This allows one to place the sensors under the strings. Therefore, there is nothing that physically limits the bow's movements.

Nevertheless, this solution is technically difficult, as described by Freed [13] in his attempt at such a technique for the guitar. He did not pursue this project due to small and variable levels of reflected light and interference with the ambient light. Other difficulties are connected to possible string bending and possible occlusion of the pickup by the hand.

We propose here a modulation technique solution that overcomes most of these problems, i.e. low light level and ambient light interference.

3. LIGHT MODULATION

To no surprise, our first attempt showed that the direct use of reflected light on the string is particularly disturbed by ambient light. To overcome, we implemented an IR modulation technique. The IR beam emitted is modulated using a 50kHz carrier. The string vibration modulates the amount of light reflected back to the IR photo receptor (a photo transistor), as described by the following equation:

$$s(t) = A(t) A_0 \sin(w_0 t + phi_0 + phi_1) + b(t) \quad (1)$$

where:

- $s(t)$ is the resulting signal,
- $A_0 \sin(w_0 t + phi_0)$ is the modulation signal with $w_0 = 2\pi \cdot 50\ khz$ and phi_0 the initial phase of the signal,
- $A(t)$ is the resulting amplitude modulated by the movement of the string and phi_1 the phase shift due to the system,
- $b(t)$ is the noise introduced by ambient light and electronic.

$A(t)$ is the interesting part that we want to extract and transform in an audio signal. We can use a synchronous demodulation circuit. $s(t)$ is demodulated by multiplying it with the 50kHz carrier $A_1 \sin(w_0 t + phi_0)$, we obtain s_d, the demodulated signal:

$$\begin{aligned} s_d(t) = A(t) A_0 A_1 \sin(w_0 t + phi_0) \sin(w_0 t + phi_0 + ph \\ + b(t) A_1 \sin(w_0 t + phi_0) \end{aligned} \quad (2)$$

$$\begin{aligned} s_d(t) = \tfrac{1}{2} A(t) A_0 A_1 \cos(phi_1) \\ - \tfrac{1}{2} A(t) A_0 A_1 \cos(2 w_0 t + 2 phi_0 + phi_1) \\ + b(t) A_1 \sin(w_0 t + phi_0) \end{aligned} \quad (3)$$

The non-modulated part of $s(t)$ is 50kHz shifted up in the spectrum and the modulated part appears around 0Hz and 100kHz. Therefore we can isolate the low frequency part of s_d by a strong low pass filter at approximatively 10kHz (not too low to preserve harmonics from the violin) to get:

$$\tfrac{1}{2} A(t) A_0 A_1 \cos(phi_1) \quad (4)$$

and transform it into an audio signal.

The phase shift phi_1 has to be as small as possible to maximize its cosine.

4. SYSTEM ARCHITECTURE

The schematic of the system is described in Fig. 1, summarizing the different steps explained in section 3.

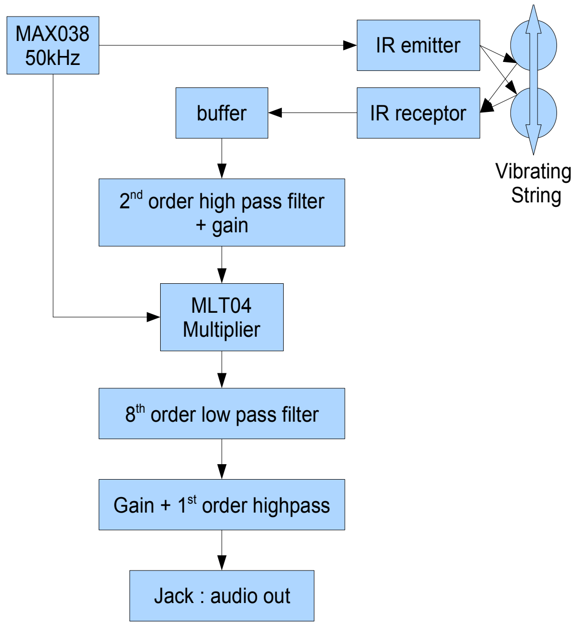


Figure 1.

The whole system is composed by 3 electronic boards :

- the pickup board holding four optical devices and an operational amplifier driving the four signals,
- the main board,
- a power supply board.

4.1 Pickup board

A Kodenshi SG-2BC reflective sensor is mounted under each string. The SG-2BC is a reflective sensor that combines a GaAs IR emitting diode (IRED) and a high-sensitivity phototransistor in a sub miniature ceramic package. Each sensor is mounted on a planar support that can be oriented by three small screws and springs to insure exact positioning under the string (Fig 3). These supports are mounted on a removable mechanical system, designed by Alain Terrier (Fig 2), held under the fingerboard.

As already pointed out, this system should not alter the violin or modify the acoustic sound of the instrument. The system is designed to be easily removable (ideally the musician should be able to use his/her own violin). Three screws are used on each side and two epoxy pads are inserted between the screws and the fingerboard to prevent any scratches on the fingerboard.

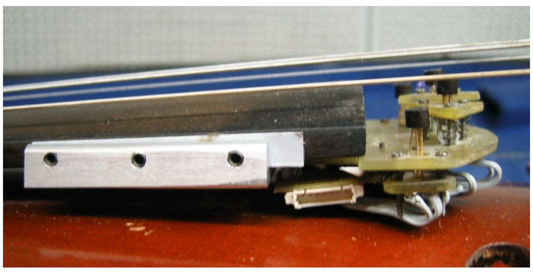


Figure 2. Side view of the pickup board

The position of the sensors is particularly important. We choose to position them as close as possible to the fingerboard in order to take advantage of the greatest amount of string vibration. However other choices are possible and are discussed in the last section of the paper.

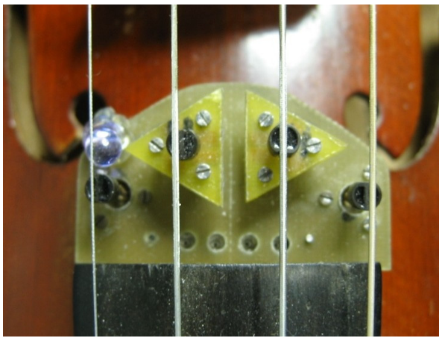


Figure 3. Top view of the pickups

An IR Led was added to the Kodenshi emitter for the E string in order to significantly increase the amount of modulated light emitted, therefore increasing proportionally the amount of reflected light.

The signals coming from the sensors are buffered by an operational amplifier and sent to the main board using a 7 wire shielded cable.

4.2 Main electronic board

The main board has three functions: 1) generating the modulation signal, 2) demodulating the incoming signals from the pickup board, 3) transforming the incoming signals in audio asymmetric signals.

4.2.1 Modulation - demodulation

We use a MAX038 waveform generator from Maxim IC to generate a 50kHz sinus. This signal is routed to two paths:

- the base of a transistor that drives the current through the IR emitters : sine signal from 0 to 100 mA,
- the synchronous demodulation circuit.

The demodulation is carried out using a MLT04 from Analog Devices. The MLT04 is a quad multiplier that performs the following operation:

$$Out = \frac{Input_1 * Input_2}{2.5} \quad \text{(volts)}$$

Each incoming signal passes through a high-pass filter in order to keep only the 50kHz frequency and then through a gain stage to make maximum use of the components' dynamics.

As shown in equation (4), the amplitude of the resulting signal is multiplied by $\cos(phi_1)$ that results of the phase shift due to the electronic system. It appears that phi_1 is very small and so $\cos(phi_1)$ is close to 1. The board includes two stages of all-pass filter aimed at shifting the phase of the reference synchronous demodulation signal.

4.2.2 Filtering and Audio Outputs

The signals coming from the demodulation are rich in high frequencies and need a strong filtering. Each signal passes through a 8th order low pass filter to eliminate the frequencies of 50kHz and above.

The 8th order filters are made using the MAX274 chip from Maxim IC and added resistor to set up the frequency and Q.

Those signals are then buffered, with a gain stage, through an operational amplifier and low pass filtered to eliminate common mode. The outputs are asymmetric audio signals. The gains found at all levels of the component chain have to be set for each string to allow maximum amplitude for the output signals without any saturation through the signal path.

5. TESTS

5.1 System use

We found that the system does not disturb the instrument: the acoustic sound of the violin and the playing techniques are not affected. Therefore, the system has a good potential for broad acceptance by musicians. The pickup board can be easily removed and the entire electronic system fits in a metal box set close to the musician.

5.2 Test patch

To be used with a computer, the system requires a sound card with 4 audio inputs. For the test we used a Motu 828 connected by firewire to a Macintosh G5 running a Max/MSP patch.

For the tests, we recorded the four signals that come from the optical pickups. A fifth audio signal from an acoustic microphone (close microphone specially designed for violin) was also recorded in order to compare the two different types of signals.

We tested the pitch tracking using the Max/MSP yin~ object, implemented at IRCAM by Norbert Schnell and based on an algorithm proposed by de Cheveigné and Kawahara[14]. Other pitch tracking objects such as fiddle~ [15] have been also tested.

Each audio signal is preprocessed to assure the best quality according to the *a priori* knowledge that we have on the signal (compression and band pass filter according to the attainable notes for the considered string) and is then routed into a separate yin~ object.

5.3 Results

The first results show that the audio signals can be retrieved with a certain noise, but this noise does not affect the results of yin~ object outputs.

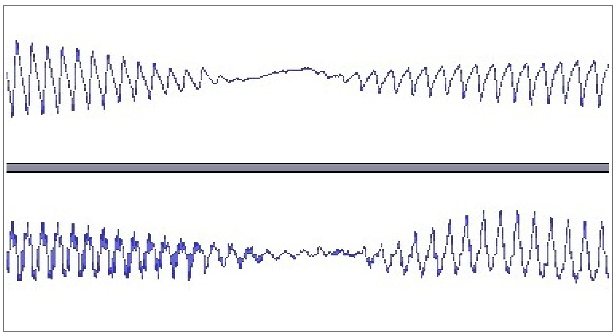


Figure 4. Waveform on the G string (top : G-string optical pickup, bottom : acoustic microphone)

Figure 4 shows the waveforms of the signals that were recorded during a test. This figure displays the transition between two different notes (upbow and downbow).

The waveform of the signal coming from the optical pickup is a good illustration of the typical stick and slip forms of the vibrating string excited by the bow. The second waveform shows the acoustic signal recorded with a microphone. As clearly seen on the transition, the acoustic microphone track contents additional spectral components due to the resonance of the violin body. This highlights that pitch tracking can be easier by capturing the string vibration directly.

Figure 5 shows the outputs of yin~ objects for each optical pickup and for the acoustic microphone. Each note composing the chords is correctly determined from each monophonic optical pickup. On the opposite, the pitch tracking failed with the acoustic signal when the chords are played. This example demonstrates the utility of this optical pickup in the case of polyphonic playing.

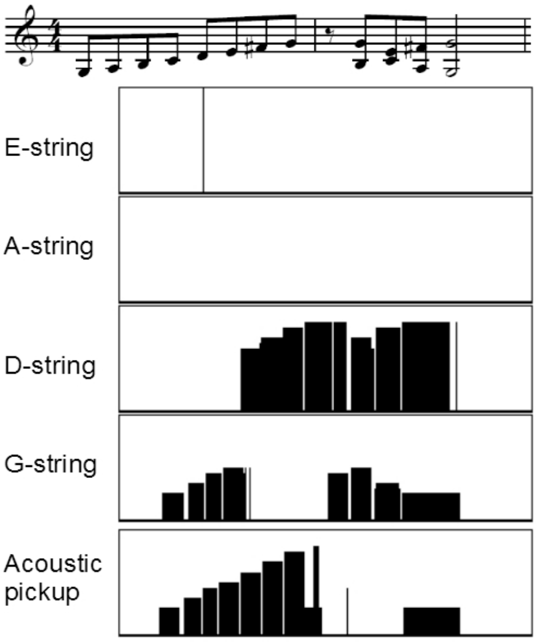


Figure 5. Estimated pitches from yin~ objects for a simple score containing chords

5.4 Issues

We found a higher crosstalk than expected between the pickups, which origin might be explained by either a mechanical crosstalk at the level of board or by an electronic crosstalk. Further investigation need to be performed to clarify such artifacts that can certainly be reduced.

The main issue is, as expected [13], potential perturbation in the optical signal due to the bow. For example, a noise burst appears when attacking the note close to the pickups. Also, there is a change in the optical signal when the bow passes just over the pickups. In this case the amplitude of the modulation signal is reduced causing some noise in the pitch tracking. This could be partially solved by a better positioning. The optical pickup could be placed very close to the bridge. In such a case such disturbance would occur only when the musician plays *ponticello.*

Finally, note that these optical pickups are aimed at performing pitch tracking and not to perform any type of "clean" amplification. The optical pickups "sound" appear to be quite noisy if listen through speakers. This is partly due to noise at the photo transistor level and relatively poor electrical characteristics of the electronic components used for this prototype.

6.CONCLUSION – FUTURE WORK

This development demonstrated that an optical reflective technique can be used to sense string vibration, and in particular in the case of the violin. The use of modulated light solves, to some extend, the perturbation of ambient light. This first prototype was found usable for pitch tracking. As discussed, the main limitation concerns possible perturbation by the bow passing over the pickups that should be improved in the future by a different positioning.

7.ACKNOWLEDGMENTS

Special thanks to Alain Terrier who has designed and build the mechanical system to hold the pickups under the bridge and to Nicolas Rasamimanana for his very useful skills on the violin and for the tests. Thanks as well to Deborah Lopatin who corrected this paper.

8.REFERENCES

[1] Augmented violin project page : http://recherche.ircam.fr/equipes/temps-reel/movement/projects/augmented%20violin.html

[2] D. Young. *The hyperbow controller: Real-time dynamics measurement of violin performance*. NIME 2002.

[3] N. Rasamimanana, F. Bevilacqua, E. Fléty, *Gesture Analysis of Violin Bow Strokes*, GW 2005, LNAI 3881, pp. 145–155, 2006.

[4] F. Bevilacqua, N. Rasamimanana, E. Fléty, S. Lemouton, F. Baschet, *The augmented violin project: case studies and performance report,* NIME 06.

[5] Axon Module : http://www.musicindustries.com/axon/

[6] Roland hexaphonic pickup : http://www.roland.com/products/en/GK-3/index.html

[7] Gibson Digital Guitar : http://www.gibsondigital.com/

[8] M. Mathews, *Pickups for the vibrations of violin and guitar strings using piezoelectric bimorphic bender elements*.
Tech. Rep. STAN-M-54, Stanford University Department of Music.

[9] Zeta Violins : http://www.zetamusic.com/

[10] Lightwave Systems : http://www.lightwave-systems.com/

[11] R. Hoag : http://www.opticalguitars.com/

[12] D. Overholt, *The Overtone Violin : a New Computer Music Instrument*, ICMC 2005.

[13] A. Freed, O. Isvan, CNMAT, *Musical Applications of New, Multi-axis Guitar String Sensors*, ICMC 2000.

[14] A. de Cheveigné and H. Kawahara. *YIN, a fundamental frequency estimator for speech and music*. J. Acoust. Soc. Am., 111:1917–1930, 2002.

[15] M. Puckette, T. Apel, D. Zicarelli, *Real-time audio analysis tools for Pd and MSP*, ICMC 1998.

Mary had a little scoreTable* or the reacTable* goes melodic

Sergi Jordà
Music Technology Group
IUA, Universitat Pompeu Fabra
08003, Barcelona, Spain
sjorda@iua.upf.edu

Marcos Alonso
Music Technology Group
IUA, Universitat Pompeu Fabra
08003, Barcelona, Spain
malonso@iua.upf.edu

ABSTRACT
This paper introduces the *scoreTable**, a tangible interactive music score editor which started as a simple application for demoing "traditional" approaches to music creation, using the *reacTable** technology, and which has evolved into an independent research project on its own. After a brief discussion on the role of pitch in music, we present a brief overview of related tangible music editors, and discuss several paradigms in computer music creation, contrasting synchronous with asynchronous approaches. The final part of the paper describes the current state of the *scoreTable** as well as its future lines of research.

Keywords
Musical instrument, Collaborative Music, Computer Supported Collaborative Work, Tangible User Interface, Music Theory.

1. INTRODUCTION
1.1 To Pitch or Not To Pitch?
At the seminal NIME conference that took place in Seattle in 2001, Perry Cook stated that when building a new controller, one of the first things he would try to do is to play the simplest song such as "Mary had a little lamb" [5]. As a response to his statement, one of this paper's author defended a complementary approach [19], which was well applicable to FMOL, the instrument he was then presenting [9] and which is also clearly extensible to the one that came after, the *reacTable** [11]: instead of trying to replicate properties that can be very well handled by traditional instruments, we prefer to invent new instruments with the potential for creating music impossible to perform using traditional ones. As an example, our instruments allow to play simultaneously with timbre and form [10], controlling dozens of parameters of which pitch is not necessarily the favourite child.

Playing the "correct" notes is one of the least requirements for someone to be considered able to play a [pitched] instrument. When Max Mathews conceived the *Conductor*, the program that would be originally used in conjunction with the *Radio Baton* [14], believing that playing a different pitch from that written in a score would almost always be considered as a mistake, he chose to automatize this step using a predefined score. He therefore prohibited the performer any type of pitch selection. A decade later, Laurie Spiegel's interactive music software *Music Mouse* allowed the performer to indicate a tendency (e.g. higher or lower) leaving to the software the final selection of the correct notes [18]. When in 1997 one of the authors designed FMOL, an instrument that was conceived with the proselytist intention to introduce newcomers, possibly non-musicians, to more experimental and 'noisy' music, in order to avoid a "wrong note syndrome" that would inhibit the performer's experimentation and freedom of expression, he decided instead to minimize the importance of pitch. FMOL does neither fix nor correct pitch; it just dethrones it from its privileged position and turns it into merely another parameter, both at the control and at the perception level.

In any case, the prevalence of pitch as the primary variable in the music of many cultures [16], and the possibility of alternative but coherent and rich approaches, constitutes indeed a non trivial research topic that surpasses the purpose of this paper. This paper introduces the *scoreTable**, a new instrument – or perhaps a variation of an existing one – devoted to playing only pitch! We would like to stress though that this does not reflect a conservative turn into the authors' musical conception!

2. TOWARDS THE *SCORETABLE**
The *reacTable**, an instrument being developed by this team durimg the last three years, is build upon a tabletop tangible user interface [11, 12, 13]. Its performance paradigm (in which several simultaneous performers share the instrument's control by moving physical artefacts on the table surface and constructing different audio topologies) is inspired in the analog voltage controlled modular synthesizers [3]. Since each musical piece has to be constructed from scratch starting from an empty table, playing the *reacTable** is equivalent to building it. This establishes a continuum not only between composition and performance [4], but even between *lutherie*, composition and performance. Since this is combined with an extensive control on the lowest timbral level, the *reacTable** performs quite well at the poles of the musical spectrum: form and timbre. It does not excel however, at the intermediate level: it is hard to perform "Mary had a little lamb" on it. We consider this a feature more than a drawback. However, this topic is frequently raised when demonstrating the *reacTable** in general non-computer music contexts.

The idea of constructing a tangible music editor based on the *reacTable** know-how and technology (which includes the open-source computer vision engine reacTivision [1] and TUIO, a protocol for table-top tangible user interfaces [13]) was initially a humble one-week project aimed at showcasing different possibilities and applications of tabletop interfaces. The project was thus not meant to be very original neither too complicated. Several related instruments already exist and ours was not initially supposed to include outstanding innovation.

2.1 Tangible Sequencers
Enrico Constanza's *Augmented Musical Stave* [6] allows constructing simple melodies by manipulating rectangular blocks. The vertical position of the each object determines the

pitch whereas the duration of each note is predetermined by the objects themselves. The *Music Table* [2] enables the composition of musical patterns by arranging cards on a tabletop. A card's position on the axis running toward or away from the user determines the pitch of the note to be played, while its position from left to right determines its timing in a looping timeline. The *Music Table* allows also to change instruments by means of special instrument card, and to save patterns or phrases in special phrases cards by means of copy cards that behave as copy tools. The *Circular Optical Object Locator* (COOL) [8] is based on a hand-rotating platter on top of which opaque objects can be placed. Rather limited in is features, its radial configuration makes it more similar to the *scoreTable**.

All these implementations provide simple ways of "writing" music interactively by means of tangible user interfaces (TUI). None of them however, complements these physical artefacts with digital visual information, at least not directly; when they provide digital visual feedback they do it on a regular monitor separated from the table, loosing thus one of the key-features of TUI, which is the seamless integration of control and representation [20].

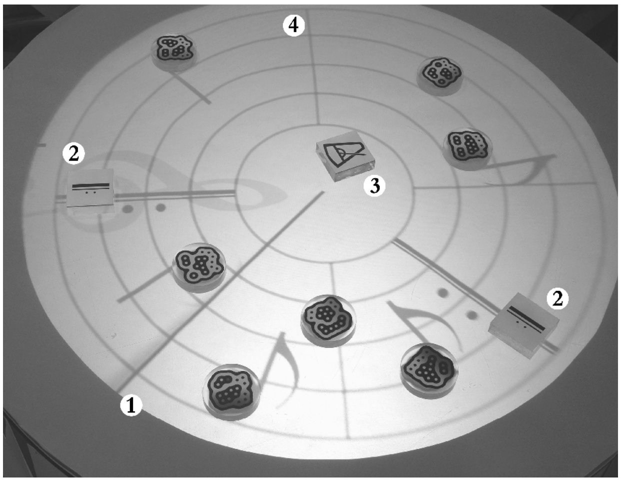


Figure 1. The *scoreTable**

(1) radar sweep, (2) begin-repeat bar lines, (3) metronome, (4) bar separators

2.2 The basic *scoreTable**

The first quick and dirty *scoreTable** had basic and straightforward functionalities (all of which are still fundamental in the current version). It allows to position physical pucks (the same ones from the original *reacTable**) in a circular looping stave. A radar sweep rotates triggering the corresponding note (by means of the computer internal MIDI synthesizer) each time it passes a note puck. The pitch of each puck is controlled by its vertical position on the stave (its distance to the centre of the table) whereas its angular position, determines the onset time of the event. In this first version, by using a MIDI piano sound without sustain, we avoided note duration considerations and problems. Furthermore, rotating a special *Metronome* object changed the angular speed of the moving radius and thus the tempo. As shown in Fig. 1, visual and sonic feedbacks make this basic setup completely self-explainable. The perceived affordance of the first *scoreTable** (using Normans' approach to the term [15]), is clear for anyone who has ever seen a musical score. Playing with this simple model was, however, more exciting than expected; perhaps because writing music is normally a non-real-time activity! These and related considerations persuaded us to further explore the *scoreTable** potential. Its current state is described next.

2.3 Real-time Tangible Music Writing

The *scoreTable** uses the same physical pucks as the *reacTable**, which come in six different shapes. On the *scoreTable**, circular pucks are used for placing notes on the staff; squares are used for all types of non-note objects which have also a place on the staff (such as *begin-repeat* bar lines for the creation of loops) (cfg. 2 in Fig. 1).; cubes are used as 6-sides program changes, and the remaining ones (pentagons, rounded squares and domes) are used as different types of tools. Several pucks come in four different colours (RGBY) allowing four part writing. Colours are currently used in circular pucks (notes) and also in some "voice oriented" tools such as transposition.

Some tools are global, affecting the whole table independently of where they are positioned. They usually control only one parameter, which can be changed by rotating the corresponding puck. Some global tools are: *Metronome* (cfg. 3 in Fig. 1), *Number of Bars*, *Key Signature, Time Signature* or *Temporal quantization*. In this sense, because it was originally conceived as a tool for showing music notation to kids, musically speaking, the current *scoreTable** implementation is quite traditionalist and still intended for tonal music. *Key* and *Time* signatures changes are obtained by rotating their respective pucks and are instantaneously reflected on the table display. *Temporal quantization* quantizes notes onsets and durations according to a selected value. This is summarized in Table 1.

Table 1. A summary of the *scoreTable Objects and Actions**

Local controls (score position dependent)	
Notes	Radial position → note pitch Angular position → onset time Color (RGBY) → instrument or track Rotation → note duration
Repeat bar lines	Create loop regions on the fly Angular position → Begin/end repeat time
Global controls (score position independent) Usually control only one parameter that can be modified rotating the puck	
Metronome	Global table tempo
Number of bars	Determines the number of bars for one complete tour [1-8]
Time Signature	Changes the global time signature
Temporal quantization	Quantizes notes onsets and durations according to a selected value, which is a subdivision of a quarter note (or no quantization)
Key signature	Changes the global key signature (or no signature)
Transposition	Transposition (can be either *tonal* or *real*)

2.4 Connecting to the *reacTable**

The *scoreTable** can operate by itself, sounding through an internal or external MIDI synthesiser (using the computer soundcard or sending it to the MIDI OUT port). It can also be connected to the *reacTable** using either MIDI directly, or sending OSC messages [21] through an Ethernet connection. In any case, four (RGBY) pucks on the *reacTable** receive pitch control information from the *scoreTable** and distribute it over the branches, according to the regular *reacTable** topological rules (see Fig. 4). In a complementary "Drum Machine Mode" each different note of the *scoreTable** can trigger an independent puck of the *reacTable**. In this case, the information is pitch-less and allows to control several independent *reacTable** branches directly from one *scoreTable** voice. This permits the control of complex and precise rhythmical structures difficult to obtain with the *reacTable**. This connection can also be used for sending other types of information, not necessarily pitched. Now that finger tracking is finally available on the *reacTable**, we plan to use this feature on the *scoreTable** for allowing users to draw envelopes of continuous time-based data. In this more flexible setup, the *scoreTable** fully becomes the time line of the *reacTable**.

3. FUTURE WORK AND CONCLUSIONS

3.1 Current Problems and limitations

We plan to overcome several "traditionalists" conditions, like the obligation of writing tonal music, but the *scoreTable** major limitations come from its hardware dimensions and technology. These will be harder to surmount. The size of the table (85 cm diameter) and the size of the pucks (5 cm diameter) are conditioned by the resolution of the camera and the computer vision engine we developed, *reacTivision* [1]. This conditions, on its turn, the size of the lines of the stave and leaves little room for additional control zones on the table surface.

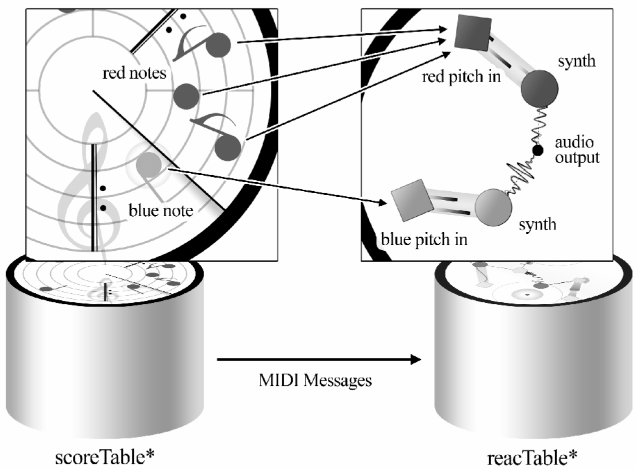


Figure 4. Connecting the *scoreTable to the *reacTable***

3.2 Virtual Operations

We are working on a new type of tools which will allow the generation of additional virtual material (i.e. notes that appear on the staff without associated pucks) as a result of the manipulation of physical notes. We are currently working on two operations, *Copy & Paste* and *Kaleidoscope*, but the list is not meant to be closed. These actions are more complex and require multidimensional control, possibly using more than one puck and sensing both the position and the orientation of each. They also constitute one of the more experimental and interesting interaction research topics of the project.

Copy & Paste tools are two-sided. One side allows to copy and store a sequence of physical notes, while the other allows pasting the contents (as is, transposed, augmented or diminished) in different parts of the score. *Kaleidoscope* operations on their turn, allow the creation of virtual notes on-the-fly. This is done by applying different types of symmetries to a slice of the score containing physical notes. *Kaleidoscope* will permit real-time control of musical set theory operations such as retrograde, rotation, inversion, transposition, multiplication, augmentation, diminution and combinations of them [7, 17]. *Copy & Paste* and *Kaleidoscope* will also be combined with paint pots, for cross-voice manipulations (e.g. copying the notes of a red voice fragment and pasting them to the blue voice).

3.3 The *scoreTable** and the *reacTable**: research on Tangible Musical Interfaces

"Traditional" instruments (acoustic, electric or electronic) as well as many non-traditional interfaces or controllers, force the performer to remain responsible, all the time, for all of the musical actions and nuances. This type of performance can be considered as the "synchronous musical activity" per excellence. On the other extreme, the *sequencer paradigm,* which still remains the most popular model of digital music creation, even if it typically incorporates some real-time actions, is mostly based on asynchronous interaction. A big mass of amateur musicians as well as professional composers and producers use a pool of standard sequencing software which try to melt, more or less seamlessly, the millennial model of the music writer with the ubiquitous and pervasive trends of the last twenty years of human computer interaction (WIMP, Drag & Drop, Copy & Paste…). The *scoreTable** is an odd hybrid that retains aspects of the "traditional" musical instrument (it is designed to be played in real-time, for "writing music performance"), while maintaining some typically asynchronous WIMP actions such as "cut & paste".

Using the TUI terminology introduced by Ulmer and Ishii [20], we can say that the *reacTable** is a relational system; it uses a homogeneous space and its topological properties are only defined by the relations between the objects on its surface, according to a building block strategy. The *scoreTable** follows a spatial approach instead; the positions where objects are placed determine their values and their functionalities. These objects can be on their turn, not mere tokens, but also *containers* (the bindings between the objects themselves and the digital information they convey becomes dynamic) or *tools*, which allow manipulating and changing the properties of other objects. In this sense, the *scoreTable** model is more "conventional" than the *reacTable**. It may also permit more "classic" research topics in Human Computer Interaction using TUIs, which could not be confronted in the *reacTable**. We thus believe that further research and brainstorming in "real-time tangible music writing" can thus bring some interesting results or ideas in music sequencing as well as in more generic and well-established human computer interaction areas.

From a musical point of view it is yet unclear if the *scoreTable** will be helpful for teaching musical notation to children, which was one of the first naïve assumptions, and something that has not been extensively tested yet. Still, it is already very fun to play and it promotes a very tight collaboration between the performers sharing the table surface. More toy-like, by itself, the *scoreTable** will probably not be either as compelling musical instrument as the *reacTable**, but it is our belief that a deeper understanding of all the concepts mentioned in this paper will lead to a positive cross-fertilization between both systems. By extension, the current parallel development of both projects, so related but so conceptually different, allows us to gain a deeper understanding of tangible user interfaces, their possibilities and their drawbacks.

4. ACKNOWLEDGMENTS

The *reacTable** team is currently constituted by Sergi Jordà, Martin Kaltenbrunner, Günter Geiger, and Marcos Alonso, who is also in charge of the *scoreTable**. Former members include Ross Bencina, who made several crucial contributions to the computer vision component and the OSC infrastructure and interns Ignasi Casasnovas and Gerda Strobl. This project is partially funded by the EU-FP6-IST-507913 project SemanticHIFI.

5. REFERENCES

[1] Bencina, R., Kaltenbrunner, M. & Jordà, S. (2005). Improved Topological Fiducial Tracking in the reacTIVision System. PROCAMS 2005, IEEE InternationalWorkshop on Projector-Camera Systems.

[2] Berry, R., Makino, M., Hikawa, N. & Suzuki, M. (2003). The Augmented Composer Project: The Music Table. Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03).

[3] Chadabe, J. (1975). The Voltage-controlled synthesizer. In John Appleton (ed.), *The development and practice of electronic music*. New Jersey: Prentice-Hall.

[4] Chadabe, J (1984). Interactive Composing: An Overview. *Computer Music Journal, 8(1)*, 22-28. Reprinted in Curtis Roads, ed., *The Music Machine*. Cambridge, MA: MIT Press, 1989.

[5] Cook, P. Principles for Designing Computer Music Controllers, NIME 2001, *ACM CHI 2001 Workshop on New Instruments for Musical Expression*, Seattle, April 2001.

[6] Costanza, E., Shelley, S. B. & Robinson, J. (2003). Introducing Audio D-Touch: A Tangible User Interface For Music Composition And Performance. *Proc. of the 6th Int. Conference on Digital Audio Effects* (DAFX-03), London, UK, September 8-11, 2003.

[7] Forte, A. (1973). Structure of Atonal Music. New Haven, Yale University Press.

[8] Hankins, T., Merrill, D. & Robert, J. (2002). Circular Optical Object Locator. Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland, May 24-26, 2002

[9] Jordà, S. (2002). *FMOL*: Toward User-Friendly, Sophisticated New Musical Instruments. *Computer Music Journal*, 26(3), 23-39.

[10] Jordà, S. (2005). *Digital Lutherie: Crafting musical computers for new musics performance and improvisation*. PhD. dissertation, Universitat Pompeu Fabra, Barcelona.

[11] Jordà, S. Kaltenbrunner, M. Geiger, G. & Bencina, R. (2005). The reacTable*. Proceedings of International Computer Music Conference 2005; Barcelona, 579-582.

[12] Kaltenbrunner, M. Geiger, G. & Jordà, S. (2004). Dynamic Patches for Live Musical Performance. In Proceedings of International Conference on New Interfaces for Musical Expression; Hamamatsu, Japan.

[13] Kaltenbrunner, M., Bovermann, T., Bencina, R. & Costanza, E. (2005). TUIO: A protocol for table-top tangible user interfaces. 6th International Gesture Workshop, Vannes 2005.

[14] Mathews, M. V. (1991). The Radio Baton and the Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music. *Computer Music Journal,* 15(4), 37-46.

[15] Norman, D. A. (1999). Affordances, Conventions and Design. *Interactions* 6(3), 38-43, May 1999, ACM Press.

[16] Pressing, J. (1990). Cybernetic Issues in Interactive Performance Systems. *Computer Music Journal, 14(1)*, 12-25.

[17] Rahn, J. (1987). *Basic Atonal Theory*. New York: London: Schirmer Books; Collier Macmillan.

[18] Spiegel, L. (1987a). *Operating Manual for Music Mouse: An Intelligent Instrument*. NY: Retiary.org.

[19] SIGCHI Bulletin, March/April 2002.

[20] Ullmer, B. & Ishii, H. (2001). Emerging Frameworks for Tangible User Interfaces. In *Human Computer Interaction in the New Millenium*, John M. Carnoll (Ed.), Reading, MA: Addison-Wesley, 579-601.

[21] Wright, M., Freed, A. & Momeini, A. (2003). OpenSound Control: State of the Art 2003. In *Proceedings of the 3rd Conference on New Interfaces for Musical Expression* (NIME03), Montreal, Canada, 2003.

Sound Rose: Creating Music and Images with a Touch Table

Alain Crevoisier, Cédric Bornand, Arnaud Guichard
University of Applied Science
Western Switzerland (HES-SO / HEIG)
Rue Galilée 15, CH-1400 Yverdon, Switzerland
{forename.surname}@heig-vd.ch

Seiichiro Matsumura, Chuichi Arakawa
University of Tokyo
7-3-1,Hongo,Bunkyo-ku, Tokyo
113-8656, Japan
sei@cfdl.t.u-tokyo.ac.jp
arakawa@mech.t.u-tokyo.ac.jp

ABSTRACT
Sound Rose is an interactive audiovisual installation created in the context of the research about new touch sensitive interfaces. This installation shows how a daily object, such as a table, can be transformed into a musical and visual instrument by using the sound produced when touching and interacting with it. This paper principally describes the hardware and software development behind the installation, as well as the design of the visual and musical interaction.

Keywords
Interactive audiovisual installation, tangible acoustic interfaces, new interfaces for musical expression.

1. INTRODUCTION
The installation was first presented at the Sound to Sense, Sense to Sound Summer School [1], held in Genoa in July 2005. It shows an application of tangible acoustic interface (TAI) as musical and visual instrument [3]. TAI interfaces are developed in the context of the European project of research TAI-CHI [2] (Tangible Acoustic Interfaces for Computer-Human Interaction), and rely on various acoustic-sensing technologies in order to detect the position of contact when touching or interacting with an object [4, 5, 6]. Similar researches have been conducted at MIT MediaLab since the late 90's for the localization of taps on various surfaces to drive graphics and sometimes audio [7]. However, the installation presented in this paper is the first one featuring the continuous tracking of finger touch using acoustic sensing technology. Continuous, multi-touch sensing has been demonstrated recently using the Frustrated Total Internal Reflection technique [9], but this approach is limited to transparent objects, such as acrylic rear-projection panes. The tracking algorithm that we are using has been developed by Politecnico di Milano, one of the partners of the TAI-CHI project, and is based on multiple sensor analysis using the dispersion property of in-solid acoustic wave propagation. In its current development, this method is able to localize and track only a single point of interaction but, on the other hand, it is compatible with a large variety of materials for flat and curved surfaces, which is a main condition if we are interested in building hybrid electro-acoustic musical instruments. In this case, the interface is not only used as a gestural controller, but also as a sound source for further real-time processing and manipulation of the sound, according to the gesture information. Therefore, the acoustic quality of the interface is of primary importance and its choice should not be subordinated to the tracking method.

In last year's first version of Sound Rose, the interaction was based on the localization of taps on a wooden table. In the new version presented in this paper, the interaction is based on the continuous touch-tracking algorithm, allowing enhanced expressiveness for the sonic and visual feedback. Other improvements include the development of a custom hardware module for signal acquisition and processing, with FireWire communication, which will be described in section 3. Accompanying software developed on the EyesWeb platform is described in section 4, and the design of the visual and musical interaction is developed in section 5. The tracking method will be detailed by its authors in a dedicated forthcoming article.

Figure 1. Audience interacting with the touch table.

2. GENERAL DESCRIPTION
The Sound Rose installation consists of a touch sensitive table with images projected from the ceiling. When users touch the table, rose-like graphics are displayed at the point of contact and

the raw sound created by the interaction with the table is processed in real-time in order to produce more elaborated sound sequences. Continuous tracking of touch allows to generate movements in graphics and to vary parameters of sound processing. Different sound presets and graphical behaviors can be selected by tapping on virtual buttons at the four corners of the table. Tracked positions and sounds are both recorded and overdubbed in loops, allowing the users to create complex visual and sonic patterns. Another button at the bottom of the table allows to erase recorded data and sounds of the selected preset.

2.1 Layout and Construction

The touch table stands in the middle of the room (about 4x4 m), surrounded by four speakers in the corners (see Figure 2). The video projector is fixed on a stand at the back of the table, about 2 meters above the projection surface. The table is made of a 100x70 cm MDF board laminated with white plastic. A black frame is delimiting the sensitive projection area to 80x60 cm. The tabletop has a tilt of 10° for ergonomic reason. The empty central piece supporting the table allows for hiding all electronic devices. There are two sets of acoustic sensors, eight dedicated to the location of touch (three sensors only are necessary but a higher number increases robustness) and two to pick up the raw sound of the table for further processing. The former are fixed slightly outside the projection area, on the upper face of the table (corners and mid-sides), and the latter under the table, at a position determined experimentally for providing the best sonic response.

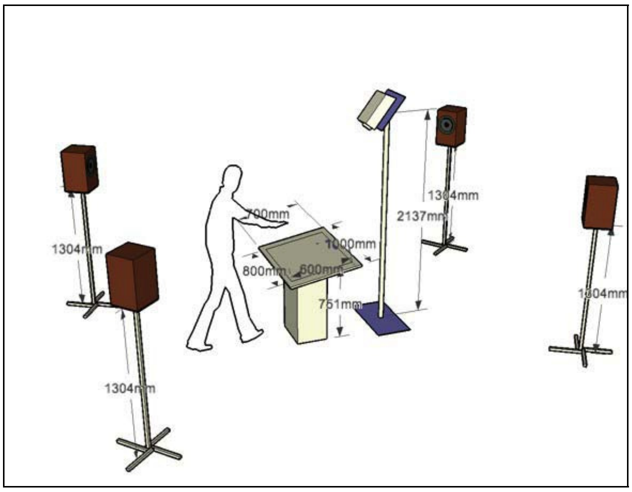


Figure 2. Layout of the installation.

2.2 System Setup

Sensors are connected to the eight preamp inputs of the hardware DSP module. In the current implementation, signals are converted to digital and sent through FireWire to EyesWeb for tracking the touch position (see Figure 3). In the future, the position of contact will be calculated directly on the DSP and sent through Ethernet using OSC (Open Sound Control) [9]. EyesWeb is also detecting if a virtual button is pressed and sends its number to a second computer (Apple, Powerbook G4, 1GHz) running Max, along with the tracked position and signal amplitude. Max is recording the position and amplitude in loop in the custom live sequencer and uses it internally to control the graphic display with Jitter.

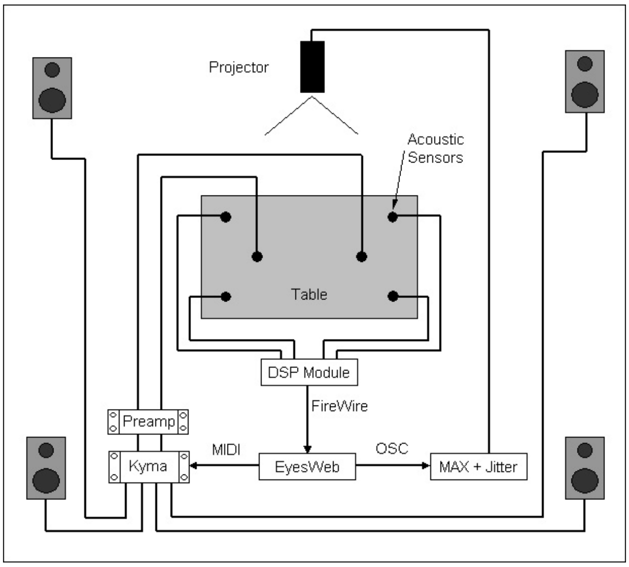


Figure 3. System Diagram.

Position and selected button numbers are also sent via MIDI from EyesWeb to KYMA, which is processing the raw sound of the table, spatializing it, and overdubbing it on a multitrack looping recorder (see figure 4).

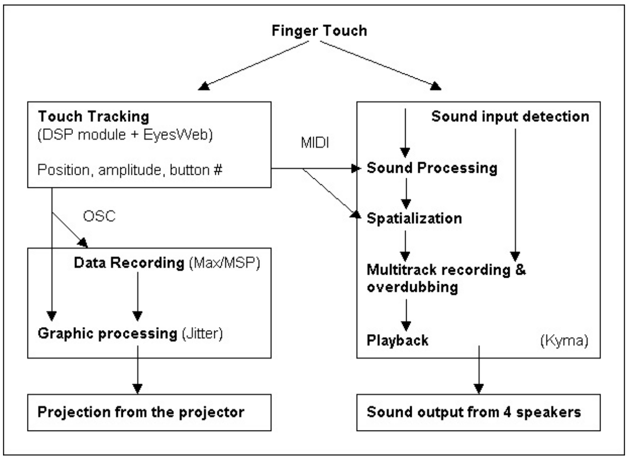


Figure 4. Process Flow-Chart.

3. HARDWARE

3.1 Architecture

The hardware used for the Sound Rose installation is part of an ongoing development concerning the realisation of a modular hardware toolkit for the processing of multimodal interfaces. It can be considered as an electronic Lego for interactive interfaces, capable of processing any kind of signals, such as audio, images, or sensors, thanks to an embedded Digital Signal Processor (DSP). The modular architecture is basically comprised of three different types of cards that can be combined together:

- Acquisition (audio, video, sensors)
- Processing (fixed point, floating point)
- Communication (Midi, Ethernet, FireWire, Wireless)

Several options exist or are under development for each kind of card, allowing to choose the right combination for each application. Cards are staked one above the other, thanks to a common bus for data transfer that is crossing them vertically.

For the Sound Rose application we have used a high sampling rate acquisition board for acoustic sensors, a fixed point DSP board and a FireWire communication board to connect to the PC.

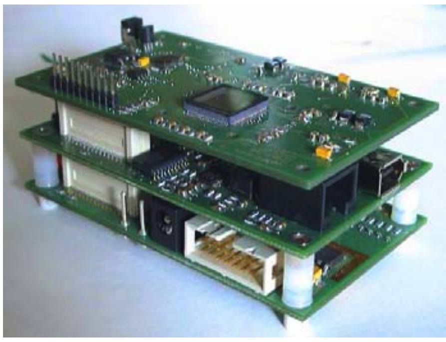


Fig. 5. The Modular Embedded Signal Processing System

3.2 Acquisition

An analog to digital conversion board was designed with 8 simultaneous acquisitions at 500 KHz with 12 bit resolution. A low noise signal conditioning circuit was developed for the acoustic signal amplification, filtering and the signal edge detection. In order to avoid signal loss and interference, a preamplifier placed in the vicinity of the sensor was designed with high input impedance and balanced, low impedance output.

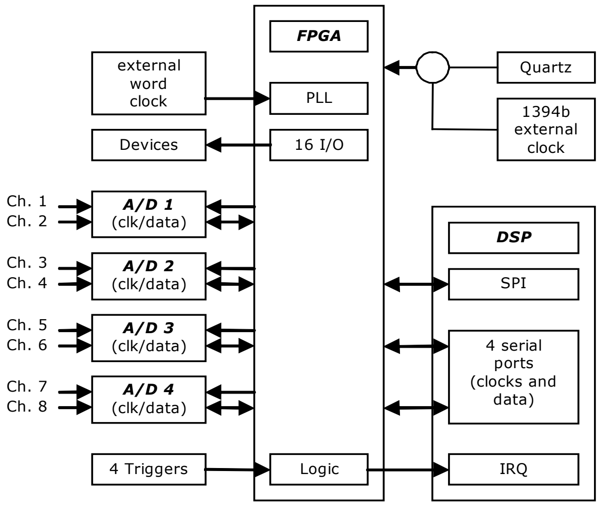


Fig. 6 A FPGA is controlling and interfacing the ADC

3.3 Processing

The processor can be either a floating point SHARC ADSP21161 or a fixed point BlackFin ADSP-BF533 of Analog Devices. The latest is running at a clock frequency up to 600 [MHz] with several interfaces such as 4 serial ports, a 16 bits asynchronous bus, a SPI and a 16 bits PPI and offers a very interesting power/price ratio. A double Blackfin version is under development, with cores running up to 750 MHz.

In the center of this classical architecture, an FPGA shown in Fig. 6 is used mainly to synchronize the signal acquisitions with one of the available clocks: internal (quartz), real-time FireWire bus (1394a external clock), and serial ports of the DSP or external word clock. The handling of the multiple interrupt signals is also done by the FPGA. Finally, 16 digital I/Os are also available on the FPGA for the configuration of the signal conditioning electronics. The 1394 FireWire board was added to the system to provide communication between the DSP module and the PC.

4. SOFTWARE

4.1 IEEE 1394a communication layer

A 1394a communication layer has been realized allowing fast transfer (up to 400 Mbits/s) of video, digital signals and parameters. The communication layer can be divided into two distinguish parts: a DSP library based on a fully compliant 1394 stack and a C/C++ library build upon a 1394 driver for the PC side. Isochronous and asynchronous communications are both available.

The combination of these two elements provides end-users or developers an easy and versatile interface with the hardware. It is possible to configure directly several parameters of the acquisition system. For instance, the user can select the sampling rate of the A/D converters or the frame rate and size of the acquired video. This communication layer consists of a generic library suitable for several platforms such as: EyesWeb which is described in the next paragraph, Matlab, Visual Studio for C/C++ programming, etc.

4.2 EyesWeb

After acquisition and pre-processing, the eight signals are transmitted to EyesWeb [11] using a specific self-build block based on the communication layer (1394 driver) described above.

The tracking algorithm has been implemented in a custom module, as well as the detection of the depressed virtual button. EyesWeb's OSC and MIDI blocks are used to transmit the touch position, button number and signal amplitude to Max and Kyma. The achieved accuracy for the tracking system is about 1 cm.

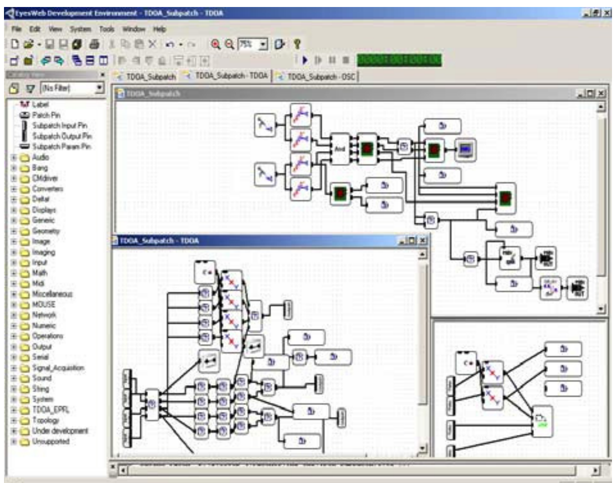


Fig. 7 EyesWeb patch and subpatches developed for the Sound Rose installation.

5. INTERACTION DESIGN

5.1 User Interface

As mentioned before, the user interface is a touch table with graphics projected on it. Interaction can take place in two ways:

a) By tapping on the virtual buttons at the four corners and at the bottom of the display area (see Figure 8).

b) By dragging or tapping one finger anywhere else in the display area.

The four virtual buttons in the corners are used for selecting a different preset. Each preset corresponds to a different sound and a different graphical behavior. The active button is highlighted with a brighter color. The virtual button at the bottom of the display area is used for erasing the sounds and graphics corresponding to the active preset.

Figure 8. Snapshot of the display area

5.2 Graphics

This part is handled by the second computer, running Max and Jitter. Max is used for recording the data sent from the first computer running EyesWeb. Data include the position's coordinates of taps or continuous touch, and the value of the sound's amplitude. The program features a 4 tracks data sequencer capable of recording and playing back data simultaneously. The Jitter part has 2 functions. One is to draw 5 virtual buttons projected on the table. Buttons colored red, blue, yellow and purple, at the corners, are used for selecting the recorded track. A green one indicated at the bottom is used for erasing data. The other function is to convert the set of data to drawing rose-like shapes and control movements of shapes in OpenGL. The direct data received when touching the table and the data played back by the sequencer both trigger the graphics drawn by Jitter. Therefore when the user touches the table, rose-like shapes appear at the same location. The user can add touches one by one to build up the visual sequence or erase data (cf. above). The amplitude value determines the size of each shape. Periodic appearances of rose-like shapes leave traces of user's interaction.

5.3 Music

This part is handled by Kyma [12], in conjunction with the position and control data received from EyesWeb. According to the preset selected by tapping on the virtual buttons, the input sound is directed to a different branch of the program patch. Each of the four sub-patches is featuring a different processing algorithm, with two parameters controlled by the x, y position of the touch provided by the tracking system. Examples of controlled parameters are the cut off frequency and gain of a resonant filter, or the amplitude and frequency of the carrier in modified FM synthesis [10], where the modulating oscillator is replaced by the picked up audio signal.

At the same time that processing parameters are varying, the sound is spatialized and panned with the same x, y coordinates. It is then recorded on a 4x4 multitrack loop recorder. There are four tracks for each preset or sub-patch, in order to keep the panning information of the four speakers. When the input sound exceeds a certain threshold, the processed sound is "punched in" the tracks, by mixing the new material with the one recorded previously. This process could also be described as automatic overdubbing. When the erase button is pressed, the four tracks of the selected preset are erased synchronously.

6. ACKNOWLEDGMENTS

This project has been realized with the support of the TAI-CHI project. The tracking method was developed at Politecnico di Milano by Prof. Augusto Sarti, Diego Rovetta, Gabriele Scarparo, Giovanni De Sanctis, and Prof. Stefano Tubaro.

7. REFERENCES

[1] S2S2 Summer School 2005 - www.s2s2.org

[2] TAI-CHI project - www.taichi.cf.ac.uk

[3] Crevoisier A. and Polotti P. Tangible Acoustic Interfaces and their Application for the Design of new Musical Instruments. *Proceedings of NIME-05*, Vancouver, Canada, 2005.

[4] Rolshofen W, Pham D.T. and al. New Approaches in Computer-Human Interaction with Tangible Acoustic Interfaces. *IPROMs 2005 Virtual Conference*, May, 2005.

[5] Polotti P., Sampietro M., Sarti A., Crevoisier A. Acoustic Localization of Tactile Interactions for the Development of Novel Tangible Interfaces. *Proc. of the 8th Int. Conference on Digital Audio Effects* (DAFX-05), Madrid, Spain, 2005.

[6] R. K. Ing, N. Quieffin, S. Catheline, M. Fink. *Tangible interactive interface using acoustic time reversal process.* Submitted to Applied Phys. Lett., 2005.

[7] Paradiso J. and Leo, C.-K. Tracking and Characterizing Knocks Atop Large Interactive Displays. In *Sensor Review* (special issue on vibration and impact sensing), Vol.25, No.2, 2005, pp. 134-143.

[8] Han, J. Y. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology,* 2005.

[9] Wright M., Freed A. and Momeni A. Open Sound Control: State of the Art 2003. *Proceedings of the 2003 Conference on New Interfaces for Musical Expression* (NIME-03), Montreal, Canada, 2003, pp.153-160.

[10] Poepel C. *Synthesized String for String Players*. Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME-04), Hamamatsu, Japan, 2003.

[11] EyesWeb Open Platform - www.eyesweb.org

[12] Symbolic Sound Corporation - www.symbolicsound.com

Synthesis and Control on Large Scale Multi-Touch Sensing Displays

Philip L. Davidson Jefferson Y. Han

Courant Institute of Mathematical Sciences
New York University
719 Broadway New York, NY 10003
{ philipd, jhan }@mrl.nyu.edu

ABSTRACT
In this paper, we describe our experience in musical interface design for a large scale, high-resolution, multi-touch display surface. We provide an overview of historical and present-day context in multi-touch audio interaction, and describe our approach to analysis of tracked multi-finger, multi-hand data for controlling live audio synthesis.

Keywords
multi-touch, touch, tactile, bi-manual, multi-user, synthesis, dynamic patching

1 INTRODUCTION
The musician's need to manipulate many simultaneous degrees of freedom in audio synthesis has long driven the development of novel interface devices. Touch sensors integrated with graphical display functionality can provide intuitively direct interactivity with richly dynamic context; however they are typically only able to respond to a single point of contact a time, making them quite limiting for musical input. *Multi-touch* sensors on the other hand permit the user fully bi-manual operation as well as chording gestures, offering the potential for great input expression. Such devices also inherently accommodate *multiple* users, which makes them especially useful for larger interaction scenarios such as interactive tables.

These devices have historically been difficult to construct, but we have taken advantage of a new rear-projectable multi-touch sensing technology with unique advantages in scalability and resolution, to create novel musical interfaces for synthesis and control in a large format dynamic workspace.

2 PREVIOUS WORK
2.1 Multi-Touch Interfaces
Boards composed of a plurality of individual controls such as sliders, knobs, buttons, keys, and touchpads, can in a sense be considered multi-touch interfaces. Advanced devices of this class include large arrays of position-sensitive touch sensors such as Buchla's *Thunder* [2], Eaton and Moog's *Multiple-Touch Keyboard* [7] and the *Continuum Fingerboard* [8]. However, we are more interested in homogeneous interaction surfaces that allow for dynamic contextualization.

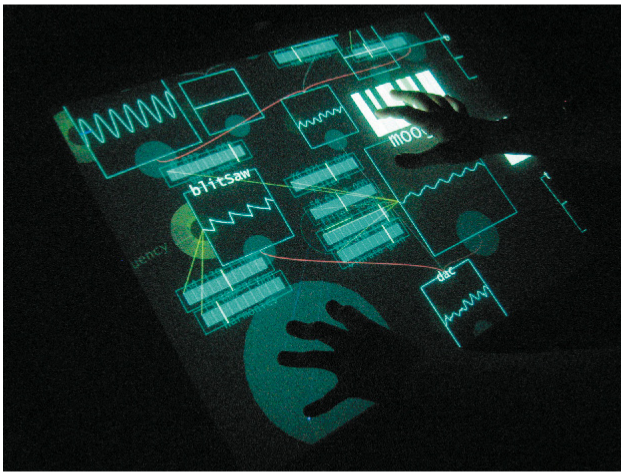


Figure 1: Rear-projected, multi-touch interaction session

Buxton experimented with continuous touch-sensing [22] as well as multi-touch sensing devices for music with the *Fast Multiple-Touch-Sensitive Input Device* [3][14]. This device was an active matrix of capacitive touch sensors, 64×32 in resolution. Instead of integrating it with a display, Buxton utilized cardboard template overlays to partition the interaction surface to provide context, in addition to kinesthetic feedback.

Tactex more recently experimented in the marketplace with a product directly aimed at musicians called the *MTC Express* [23]. This device optically measured the compression of a translucent compressible foam, and though it only had a spatial resolution of 8×9, it has an impressive temporal sampling rate (200Hz) and dynamic range in pressure, making it mostly useful for percussive control.

The recent *Lemur* from JazzMutant [11] is a multi-touch sensor that is tightly integrated with an LCD display. The device is sized for , and functions as a software-configurable controller board. However, the device is low resolution (128×100) and provides no pressure information, limiting the sophistication of the interface widgets that are provided. Furthermore, the system is not open enough to allow access to either the raw sensor data stream or to the raw display itself, limiting its usefulness for the exploration and development of novel interfaces.

All of the systems above have a complexity on the order of the number of tactels, which limits both resolution (though interpolation and other signal processing techniques can mitigate

Figure 2: AudioPad, reacTable*, and Lemur

this for a sparse set of contacts) and physical scale, reducing their role in musical performance to a component within a larger system. Other more scalable multi-touch sensing technologies are starting to become available [6][21][26], but these are still difficult/expensive to obtain, and we have not yet seen any reports of their usage in a musical context.

2.2 Tangible Interfaces

Larger scale musical interfaces have also developed around the concept of the manipulation of trackable tangible assets, such as blocks or pucks. These tangible interfaces [10] can accommodate more than one hand and/or more than one user, and take advantage of the user's sense of kinesthesia and skills in three-dimensional spatialization.

The *AudioPad* [19], is a tabletop instrument which utilizes modified Wacom tablet systems to track the position and orientation of a limited number of pucks. This tabletop environment enabled the dynamic control of loops of other synthesis through marking menus, and also allowed the pucks to act as dials and other controllers to vary parameters. Pucks could also be equipped with a pushbutton, which could be regarded as 1-bit pressure sensitivity.

d-touch [5] and the *reacTable** [12] are more recent tabletop instruments based on vision-based tracking of optical fiducials. They track many more pucks without compromising the sensing update rate, and have developed several tangible musical interface paradigms.

We find that these, and other tangible instruments [1][16][17][18] provide an intuitive and approachable environment for musical control, but face challenges as the complexity of the environment increases.

3 SYSTEM OVERVIEW

Through the usage of a scalable high-resolution multi-touch sensing technique, we build a system that encompasses the functionality of both the virtualized controllers possible on multi-touch devices such as *Lemur*, and the space and scale of multi-user patching systems such as *AudioPad* and *reacTable**[13].

The technique is based on *frustrated total internal reflection* [9], implemented in the form factor of a 36"x27" drafting table, at a sensing resolution of ~2mm at 50Hz. It provides full touch image information without any projective ambiguity issues whatsoever. The touch information is true- it accurately discriminates touch from a very slight hover, while also providing pressure information. The sensor image sequence is analyzed and parsed into discrete stroke events and paths with a processing latency of about 3.5ms on a 3GHz Pentium 4. Measurements including position, velocity, pressure, and image moments are sent to client applications using the lightweight OSC protocol [27] over UDP. The system is notably graphically integrated via *rear*-projection, preventing undesirable occlusion issues.

For our experiments with audio control, we built a simple set of synthesis modules using STK [4], controlled by a modular patching interface.

4. DISCUSSION

4.1 Graphical Context

As Buxton first demonstrated, context is a critical issue for touch interfaces. While we are a few steps beyond cardboard overlays, context for interaction on continuous control surfaces is a challenging problem. Although the pucks used in *AudioPad* and *reactTable** are visually passive, information is projected on and around the puck to provide additional feedback to the user. As such, they are a convenient metaphor for control in contextualizing the surface.

4.2 Basic Gestures

Pucks emphasize our ability to precisely manipulate objects between our fingers. True multi-touch surfaces should provide a similar capacity for manipulation, in contrast to a discrete set of continuous controls. We begin by extending the dextrous manipulation concept to the touch surface by creating regions of the surface that act as virtual puck-like widgets. Touch information captured by each widget is processed together as a single complex gesture. As with pucks, we use the space in and around these controllers for rich visual feedback.

4.3 Interpretation Model

Free from the limitation of the physical world, we can start to extend the metaphor of the basic puck- for instance, the control region associated with a widget can be dynamically resized or reshaped in the course of a performance.

We can also flexibly divide inputs into separate control groups, and selectively constrain degrees of freedom while maintaining a robust handling of under- or overconstrained input cases. As an example, constraining the transformation to rotation and translation is equivalent to the degrees of freedom in a physical puck, while constraint to single-axis translation acts as a slider.

We implemented the more traditional interface widgets such as sliders, knobs, and keys, which the performer can manipulate any set of simultaneously. Additionally, the availability of pressure information allows for more sophisticated revisions of these basic controls. We also use a 'deadband' model [15] to differentiate between tracking and control, permitting the precise acquisition of control elements by the user. Pressure data is also heavily used for more novel controls such as Zliders [20], as well as control pads which interpret relative pressure values as tilt measurements.

4.3 Complex Gestures

With the input captured from two or more hands, we can start to simulate physical manipulations such as strain, twist, or bending motions. Through this we can consider virtual instruments controlled by simplified physical systems - for example, we could monitor volume of a deformable object to determine the flow rate for a wind controller, or use strain measurements to modify string tension or resonance modes. We are currently exploring

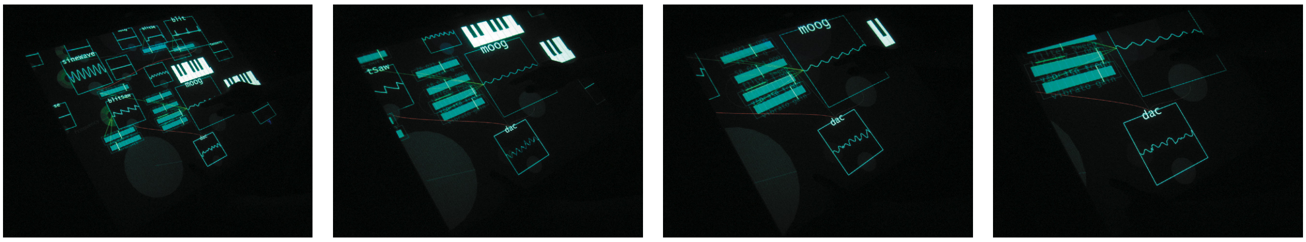


Figure 3: Dynamic workspace- users easily pan/zoom/rotate with a bimanual gesture

the possibilities using a fretboard and plucked string model to produce an autoharp, or koto-like instrument.

4.3 Structural Flexibility

We find that contextualizing manipulation through widgets allows similar precision in parametric control as a physical puck model, and that multi-touch gestures are a natural extension of the control space. Capturing the wide gestural range possible with the hand [24] requires that the sensor accurately track points in close proximity, and control gestures must recognize the limitations of hand geometry as described in [25], to prevent painful or impractical gestures. One advantage to virtualization is that each arrangement can conform to the size and shape of the user's hands, preventing undue stress. As with any continuous control surface, widgets may be adjusted, expanded or repositioned without the synchronizing the location of their physical counterparts. In Figure 3, we show the use of a two-dimensional view manipulator, actuated with a simple two-fingered gesture, allowing the user to pan, zoom, and rotate the workspace and inspect a modular element in detail with no loss of context, giving the performer the ability to manage large workspaces much more effectively.

5 FUTURE DIRECTIONS

There are some limitations in the core implementation that we would like to address that would further increase its usefuless for musical applications. For instance, our current sample rate of 50Hz is good but not great, particularly for percussive input, although this is mitigated by the fact that a large amount of simultaneous information can be updated for each frame. We will be immediately upgrading the system to achieve 120Hz or more.

Also, our current setup provides context only through visual means, but we are definitely looking to be able to provide some degree of haptic feedback as well.

We will continue to explore new and design of new widgets in this new domain. While the table has its advantages over traditional control surfaces, we are primarily interested in controls that take full advantage of the multi-touch data. A uniform control surface also raises the possibility of flexible interfaces - for example, a piano keyboard interface that adjusts spacing based on a user playing a set of prompted chords. In provided a customized scaling of the interface we can adapt to different players to better fit their stature, or to reduce RSI related conditions.

The versatility of the sensor allows for much more interesting form-factors than the console table we have shown here. In particular, for multi-user collaborative setups, we can envision a wider setup where two musicians perform on the same surface, while passing or linking sonic elements in a shared workspace.

Multi-touch sensing is currently an active field in HCI research, so we stand to harness the fruits of much other work in advancing the intuitiveness, efficiency, and usability of this unique family of interfaces.

6 REFERENCES

[1] Berry, R., Makino, M. , Hikawa, N. and Suzuki, M. 2003. The Augmented Composer Project: The Music Table. In *Proceedings of the 2003 International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, 2003.

[2] Buchla, D. 1990. Thunder. http://www.buchla.com/historical/thunder/

[3] Buxton, W., Hill, R., and Rowley, P. 1985. Issues and Techniques in Touch-Sensitive Tablet Input. In *Proceedings of the 12th Annual Conference on Computer Graphics and interactive Techniques* SIGGRAPH '85. ACM Press, New York, NY, 215-224.

[4] Cook, P. R. and G. Scavone. 1999. The Synthesis Toolkit (STK). In *Proceedings of the International Computer Music Conference*. International Computer Music Association, pp. 164-166.

[5] Costanza, E, Shelley, S. B., and Robinson, J.. Introducing Audio d-touch: A Tangible User Interface for Music Composition and Performance. In *Proceedings of the 2003 International Conference on Digital Audio Effects*, London, UK, September 8-11 2003b.

[6] Dietz, P. and Leigh, D. 2001. DiamondTouch: a Multi-User Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology* (Orlando, Florida, November 11 - 14, 2001). UIST '01. ACM Press, New York, NY, 219-226.

[7] Eaton, J. and Moog, R. 2005. Multiple-Touch-Sensitive Keyboard. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada.

[8] Haken, L. 2005. Continuum Fingerboard. http://www.cerlsoundgroup.org/Continuum/

[9] Han, J. Y. 2005. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology* (Seattle, WA, USA, October 23 - 26, 2005). UIST '05. ACM Press, New York, NY, 115-118.

[10] Ishii, H. and Ullmer, B. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms.

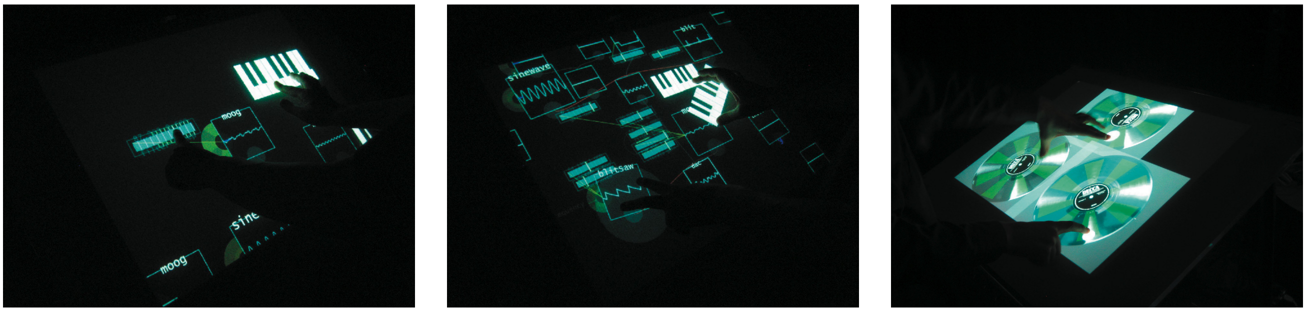


Figure 4: Experiments in multi-touch interfaces

In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, United States, March 22 - 27, 1997). S. Pemberton, Ed. CHI ‘97. ACM Press, New York, NY, 234-241.

[11] JazzMutant. 2004. Lemur. http://www.jazzmutant.com/

[12] Jordà, S. & Kaltenbrunner, M. & Geiger, G. & Bencina, R. The reacTable*. In *Proceedings of the International Computer Music Conference (ICMC2005)*, Barcelona (Spain)

[13] Kaltenbrunner, M. & Geiger, G. & Jordà, S. 2004. Dynamic Patches for Live Musical Performance. In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04)*, Hamamatsu, Japan

[14] Lee, S. K., Buxton,W. and Smith, K. C. 1985. A Multi-Touch Three Dimensional Touch-Sensitive Tablet. In *Proceedings of CHI ’85 (April 1985), ACM/SIGCHI*, NY, 1985, pp. 21–25.

[15] Miller, T. and Zeleznik, R. 1999. The Design of 3D Haptic Widgets. In *Proceedings of the 1999 Symposium on interactive 3D Graphics* (Atlanta, Georgia, United States, April 26 - 29, 1999). SI3D ‘99. ACM Press, New York, NY, 97-102.

[16] Newton-Dunn, H, Nakao, H., and Gibson, J. 2003. Block Jam: A Tangible Interface for Interactive Music. In *Proceedings of the 2003 International Conference on New Interfaces for Musical Expression*, Montreal, Canada, May 22-24 2003.

[17] Paradiso, J. and Hsiao, K. 1999. A New Continuous Multimodal Musical Controller using Wireless Magnetic Tags. In *Proceedings of the 1999 International Computer Music Conference*, pages 24–27, Beijing, China, October 22-28 1999.

[18] Paradiso, J. A. 2002. Several Sensor Approaches that Retrofit Large Surfaces for Interactivity. Presented at the *UbiComp 2002 Workshop on Collaboration with Interactive Walls and Tables*, Gothenburg, Sweden, September 29, 2002.

[19] Patten, J., Ishii, H., Hines, J., and Pangaro, G. 2001. SenseTable: A Wireless Object Tracking Platform for Tangible User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, United States). CHI ‘01. ACM Press, New York, NY, 253-260.

[20] Ramos, G. and Balakrishnan, R. 2005. Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation. In *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology* (Seattle, WA, USA, October 23 - 26, 2005). UIST ‘05. ACM Press, New York, NY, 143-152.

[21] Rekimoto, J. 2002. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI ‘02. ACM Press, New York, NY, 113-120.

[22] Sasaki, L., Fedorkow, G., Buxton, W., Retterath, C., Smith, K.C. 1981. A Touch Sensitive Input Device. In *Proceedings of the 5th International Conference on Computer Music*. North Texas State University, Denton Texas.

[23] Tactex. Smart Fabric Technology. http://www.tactex.com

[24] Waisvisz, M. 1984. The Hands. http://www.crackle.org/TheHands.html

[25] Wessel, D., Wright, M., and Schott, J. 2002.Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, Dublin, Ireland, May 24-26, 2002

[26] Westerman, W., Elias, J. G., and Hedge, A. 2001. Multi-Touch: A New Tactile 2-D Gesture Interface for Human-Computer Interaction. In *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting*, Vol. 1, pp. 632-636. 2001.

[27] Wright, M. 2003. OpenSound Control: State of the Art 2003. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003.

Towards a Coherent Terminology and Model of Instrument Description and Design

Tellef Kvifte
Dept. of musicology
University of Oslo
P.O.Box 1017 Blindern
0315 Oslo, Norway
tellef.kvifte@imv.uio.no

Alexander Refsum Jensenius
Dept. of musicology
University of Oslo
P.O.Box 1017 Blindern
0315 Oslo, Norway
a.r.jensenius@imv.uio.no

ABSTRACT

This paper discusses the need for a framework for describing musical instruments and their design, and discusses some possible elements in such a framework. The framework is meant as an aid in the development of a coherent terminology for describing, comparing and discussing different musical instruments and musical instrument designs. Three different perspectives are presented; that of the listener, the performer, and the constructor, and various levels of descriptions are introduced.

Keywords

Musical instrument design, mapping, gestures, organology.

1. INTRODUCTION

In the literature on musical instruments and musical instrument construction, one central theme is the relation or mapping between gestures[1] used to control an instrument and the resultant sounds. In later years, an increasing number of papers describe aspects of such gesture-sound relationships, many of them basing the discussion on specific examples of newly constructed controllers and/or instruments. In many cases, however, these discussions do not address more general principles, and even if the instruments described are interesting, the discussions do not necessarily add to a broader understanding of musical instrument construction. Part of the problem is a lack of consensus on terminology, and this paper is an attempt to start a discussion of necessary, fruitful and convenient terminology in the study of musical instrument description and construction. Through this, we advocate a field of study which might be called *theoretical organology.*

The construction of new instruments and mappings raises a number of considerations. One set concerns the listener: What kind(s) of sounds do we expect when we see a certain gesture? What kind(s) of gestures do we imagine when we hear certain sounds. Another set concerns the performer: What are intuitive and natural mappings in the instrument? What are exciting mappings that stimulate creativity?

[1]"Gesture" is here defined as body gestures, i.e. physical movement.

This is not to say that new mappings should always follow such expectations, but a better knowledge of gesture-sound relationships would certainly help in both making mappings that both conform to, *and* possibly also violate, our expectations. In both cases a set of conceptual tools are needed; and to be able to draw on the great number of studies that have been, and will be, published, we need to start discussing to coordinate the terminology being used.

2. THREE PERSPECTIVES

There is of course a close connection between overall perspective and terminology. We have already mentioned two different perspectives, that of the performer and that of the listener. A third is that of the instrument constructor. The following sections will present three models of an overall performer-instrument system corresponding to these three perspectives; and how they can be described.

2.1 The listener

Seen from a listener's perspective, it is important to be able to characterize the general relationship between gestures and the emerging sound (Figure 1).

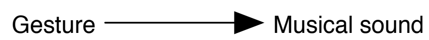


Figure 1: Gesture - Sound

The musical sound may also invoke some kind of listener activity, for example dancing, foot tapping or applause. Figure 2 presents this distinction of Gesture P(erformer) and L(istener).

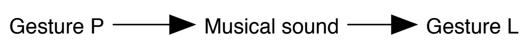


Figure 2: Musical sound also evoking new gestures

The gestural activity of the audience may in turn influence the musicians and their gestures; and we will have a closed loop of information flow (Figure 3).

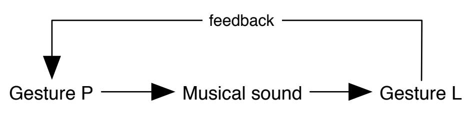


Figure 3: Closed information loop

In this paper, where we focus on the description and construction of musical instruments rather than the interplay between musicians and audience, this larger loop is not in the center of attention. We will rather concentrate on how the listener perceives the interplay between the musician, the instrument, and the resulting sound, as seen from a smaller or greater distance. We want to characterize how the actual connection between gestures and musical sound, i.e. the *mapping*, is perceived (Figure 4).

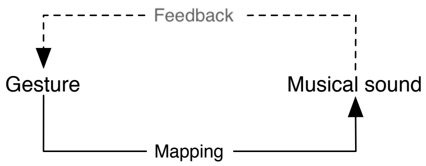


Figure 4: Model of the listeners perspective

Seen from a distance, smaller details of finger movements will probably not be as important as larger bodily movements, and one may not be able to distinguish easily between sound-producing gestures and other movements. Purely expressive and/or optional gestures may nevertheless be experienced as relevant to the sound by the audience, and it is not obvious where to draw the analytical line. As Wanderley et. al observe: "[...] clarinetists' ancillary gestures are not randomly produced or just a visual effect, but rather they are an integral part of the performance process." [10, 98]

To account for this, descriptions of the mapping between gesture and sound at this level might therefore include several levels of resolution.

2.2 The performer

In a model of the performer's perspective, we add a *device* and include the feedback. The performing subject is only implicitely represented in the figure, as the agent performing the Gesture, and receiving the Feedback (Figure 5).

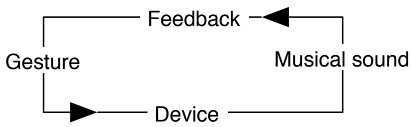


Figure 5: A first approximation of a model of the performer's perspective

Several models building on this basic scheme have been proposed. One rather abstract model is found in [7], and includes just the performer's perceptions and intentions about the playing on the one hand, and what is called the *instrument control* on the other (Figure 6). Implied in the model is that the feedback from the instrument control to the performer is in terms of sound, understood as music. This model is focused primarily on the performer, to the extent that the instrument as such is not present; only the parts of the instrument that are sensitive to control. These are called *control organs* in this model, and we will use this term to denote instrument parts like keyboards, buttons, finger-holes etc., as well as sensors of various kinds that are put to use for controlling sound-producing hard- and software.

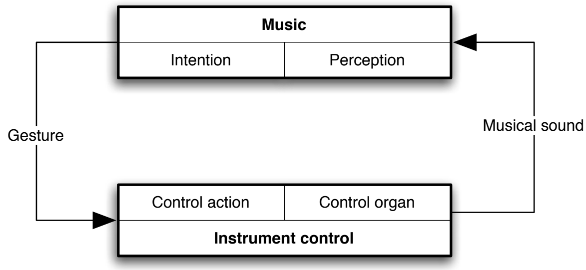


Figure 6: The playing technique perspective after [7]

Choi's model (Figure 7) is a bit more detailed, as the instrument here also includes the sound-producing parts, and the instrument control, here called *interface*, is exemplified to a certain extent. Note that both these models only present auditory feedback; none of them take haptic or tactile feedback into account.

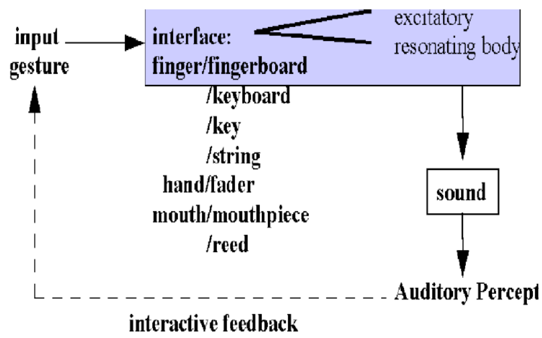


Figure 7: Interactivity of solo performer with musical instrument. After [2]

An even more abstract model is found in [5], as shown in Figure 8. In this model, performer actions and input devices correspond to the control actions and control organs in Figure 6. Here, the sound producing device is included, but the performer and feedback loops are not shown. This model highlights the mappings between performer actions and the sound producing device as a possibly rather complex system of connections.

A more comprehensive model should include the map-

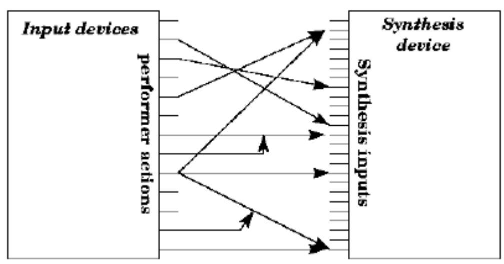


Figure 8: Mapping of performer actions to synthesis parameters. After [5]

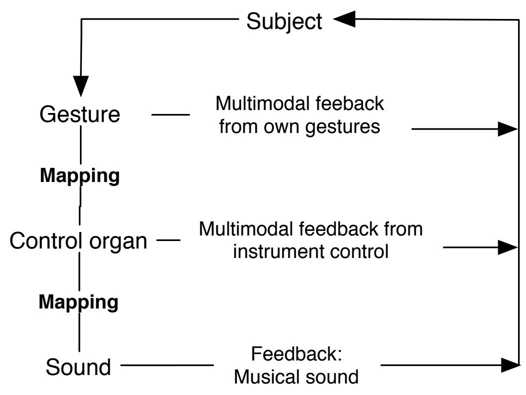


Figure 9: A model of the performer's perspective

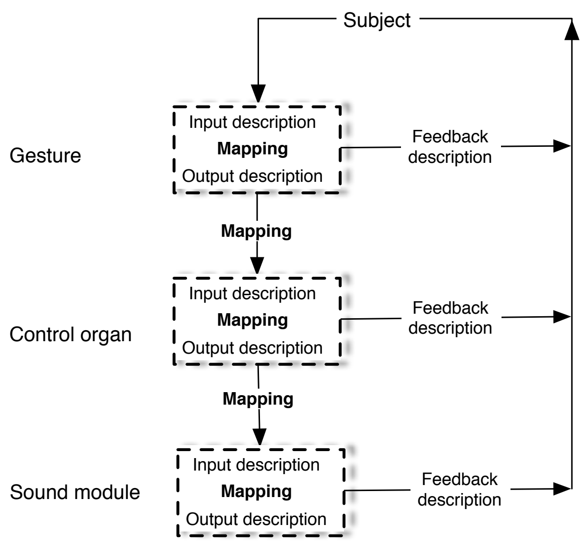


Figure 10: Model of the constructor's perspective

pings "gesture – control organ" and "control organ – sound", but in the musical information flow, the inner workings of the sound module is not of interest to the performer, so that part is reserved for the constructors perspective.

We also find it important to explicitly show the feedback from the different parts of the system. The performer may receive and use feedback from his own gestures, from the contact with the control organs of the instrument, as well as from the actual sound produced (as illustrated in Figure 9). Sometimes the feedback is essential — it is literally impossible to play a theremin without hearing the sound and pitches produced. In other cases feedback may be helpful, but not essential, like the visual feedback from a piano style keyboard, as is illustrated by e.g. professional, blind pianists.

2.3 The Constructor

The constructor needs a far more detailed view of the system than the listener and performer presented in the previous sections. There are several models in the literature, each focusing on different aspects. A relatively comprehensive model might look like the one in Figure 10.

This model suggests that the mappings may be viewed as chained: the output of one mapping is the input to the next. Also, the phenomena to be described are quite diverse; from space-time trajectories (gestures), to interface layout, to sound synthesis descriptors, to descriptions of sound as music.

This model is quite similar to the one found in [1], where the interesting concept of 'related–to–perception' parameters is introduced, (Figure 11), but slightly more explicit in the number of descriptions and mappings. Unlike Arfib's, however, the model in Figure 10 does not say anything about actual parameters to form the descriptions, or about the mappings between them.

2.4 Technical vs. musical construction, or the role of information and energy

In models of the kinds shown here, we will find accounts of both information flow and flow of physical energy, and it is not necessarily obvious whether Figure 10 refers to information or energy, or both.

One reason for this, is that the concept of 'energy' is used in many different ways, both as a concept of physics, but also to describe perceived qualities of music, like when we talk about 'energetic playing', 'forceful sounds' etc.

On closer inspection, it is obvious that the upward flow in the figure refers to information exclusively, and not to physical energy, as the only entities flowing are *feedback descriptions*. This is not to say that physical energy is not involved in the process, but only to point out that what we are interested in, also concerning energy, is the information conveyed.

Striking a piano key requires a performer's energy, and this energy is musically relevant since it determines the force with which the hammer strikes a string, which in turn determines the energy of the sound produced, which in turn determines the energy reaching the ears of the performer and listener. This energy, however, is not the same as perceived loudness, when loudness is seen as a musical parameter. While the physical energy level decreases with the square of the distance from the sound source, the perceived musical loudness may be almost invariant over quite a distance from the sound source. For the performer, the actual energy used is less interesting than the *difference* in energy needed to produce a *difference* in a perceived musical parameter.

In general, as far as Figure 10 is seen as a model of music-making, one should only be concerned with information. The *musical construction of an instrument* is a matter of information processing. However, for the constructor, a piano key must be made so that differences in output levels, corresponding to musically meaningful differences, are physically feasible for the performer. And the chain of energy that carries the information around the circuit must be geared to the equipment used at the different stages.

Such considerations may be kept separate from the musical construction of the system, and we believe the musical construction will become clearer if this separation is made.

3. PARAMETERS

3.1 Parameter types

The parameters involved in the description of a musical instrument may be organized in the following general types:

- Gestural parameters
- Technical parameters

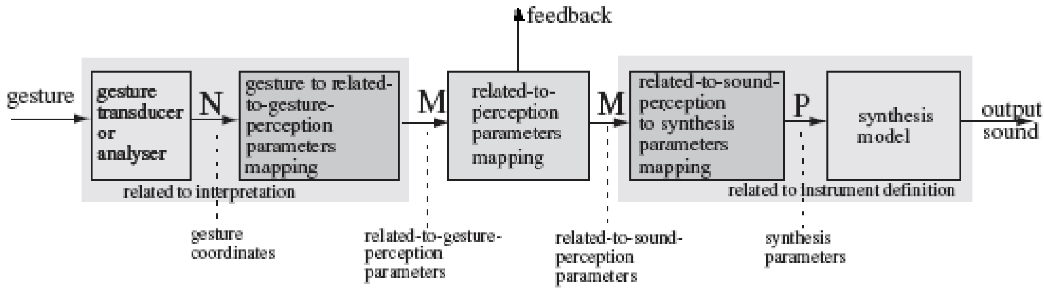


Figure 11: Mapping chain after [1]

- Musical parameters

Typically, the input parameters of the whole system will belong to the first type, while the output will, as argued above, best be described in terms of musical information.

The different types of parameters must be described in different ways, and the challenge is to find ways to connect the descriptions to find common or corresponding properties in the different kinds of description.

3.2 Parameter description

In this section, we will concentrate on properties that are valid across different kinds of parameters and their description in detail. This concerns level of specificity, the distinction discrete/continuous, and the concept of measurement levels. These are all considerations that may be helpful when looking for how properties of parameters at one point are reflected in properties of parameters at another point in the chain.

3.2.1 Levels of specificity

It is obvious that there is a need for descriptions on different levels, or with various degrees of detail. The gestures used by performers might be best described in rather broad terms in the perspective of the listener, while one will need a far greater amount of detail when describing them from the performers perspective and the constructor may need even more detail.

3.2.2 Musical parameters

One of the real challenges, is to define relevant descriptions of musical output. At a very general level, the parameters *pitch, loudness, timbre* and *duration* may be used. As soon as one wants more specific descriptions, however, there is a large number of possible descriptions, and meaningful descriptions are very much depending on musical style, as well as on experience with the instrument in question.

The descriptions needed, develop with experience, and with development of new instruments. From the very start of synthesizer construction, a large number of new parameters, especially for timbre, became available for exploration and incorporation in musical practice through new user interfaces. More recently, a similar development concerning musical manipulation of time in various software packages, widens our musically relevant parameters in the field of rhythm, tempo and time.

In this context, we will touch upon only the most general level.

3.2.3 Levels of measurement

One way to characterize parameters in a general way, is through the concept of *levels of measurement* [4]:

- Nominal level: Values may only be distinguished from each other, and not ordered.
- Ordinal level: Values may also be ordered in a sequence.
- Interval level: Values may be ordered, and there is a way to measure distance between values.
- Ratio level: Values may be ordered; distances measured, and there exists an absolute zero value, so that division and multiplication of values are meaningful operations.

At a general level, these four levels may be associated with the four general musical parameters in the following way:

- Nominal level: Timbre. There is no generally accepted way to characterize and organize timbre, but it is still possible to differentiate between different instruments, and it is possible to construct scales for *aspects* of timbre (e.g. brightness).
- Ordinal level: Loudness. It is obvious how to order loudness levels, but not necessarily how to describe precise *intervals* of loudness.
- Interval level: Pitch. Musical pitches are ordered into classes with perceptually meaningful comparable distances, like half tone, whole tone etc. But there is no meaningful zero point.
- Ratio level: Durations may be ordered and measured, and there is an obvious zero (no duration), and ratios are meaningful — a half note is half the duration of a whole note.

The implication for the control of musical instruments is that an input parameter has to be on at least as high a level of measurement as the output parameter it is meant to control. But there is not always a need for controlling an output variable at the maximum level, as will be explained in more detail below.

3.2.4 Discrete vs. continuous parameters

The perception of the four general musical parameters may be described as having two different dimensions, *discrete* and *continuous*. The discrete dimension is tied to categorization or sets of concepts; the continuous to gradual variations. Pitch may be perceived both as belonging to discrete pitch-classes (c, d etc), *and* as a continuous entity as in vibrato and glissandi. Similar considerations may be done for the three other general parameters (See e.g. [3] and [7]).

There are several implications of this for the connection between input (gestural) parameters and output (musical) parameters (Figure 12). First of all, a continuous output parameter needs a continuous input control to be controlled in detail. But it is also possible to use a discrete input to *trigger* a preprogrammed continuous variation of output like a vibrato on an synthesizer. On the other hand, a discrete output may be controlled by a discrete input, like pitch classes controlled by a keyboard, but there are also numerous examples of a continuous controller controlling a discrete output parameter, like a trombone slide controlling discrete pitch classes (as well as continuous pitch variations).

Input	Output: Discrete	Output: Continuous
Discrete	Piano keyboard: discrete pitch	LFO triggers in synths. Rare in acoustic instr.
Continuous	Trombone slide: discrete pitch	Trombone slide: continuous pitch Striking force: loudness

Figure 12: Combinations of discrete and continuous input and output

The various combinations of discrete and continuous variations for input and output for different musical parameters give different demands and possibilities for the performer. This is generally acknowledged by performance teachers and students, even though we have little general knowledge on the effects of different combinations.

3.3 The description of mappings

3.3.1 The mapping chain

In the three perspectives discussed above, the mapping chain was described in increasing detail; with only one mapping from gesture to sound in the listeners perspective, to a chain of five different mappings in the constructors perspective.

All the mappings in the last chain will at some point have to be described during the construction of an instrument, but they will in a sense be subordinate to the overall gesture–sound mapping, that might be called the *defining mapping* for the instrument. It is this overall gesture–sound mapping which defines the identity of an instrument, and this is what we will focus on in the following sections.

3.3.2 Basic mapping strategies

There are three basic approaches to the actual mapping. The traditional way is to describe and construct a fixed or static mapping, where the relations between input and output parameters stay the same. Traditional acoustic instruments are usually well suited to such descriptions.

Another possibility is a variable mapping, where mappings may be changed by the performer. Most commercial synthesizers are good examples of this approach.

Finally, mappings may be the outcome of a dynamic (learning) process where the performer can choose the gestures, or the mapping is modified by the actual behaviour of the performer (e.g. [1] and [9]).

All three cases need a common way of describing mappings.

3.3.3 A general mapping description problem

Several authors mention that mappings may take different forms with regard to how many input parameters are controlling how many output parameters. In [7], the term ‘coupling’ is used, differentiating between ‘control couplings’ and ‘sound variable couplings’, corresponding to the many-to-one and one-to-many examples in Figure 13 respectively (see also [1] and [5]).

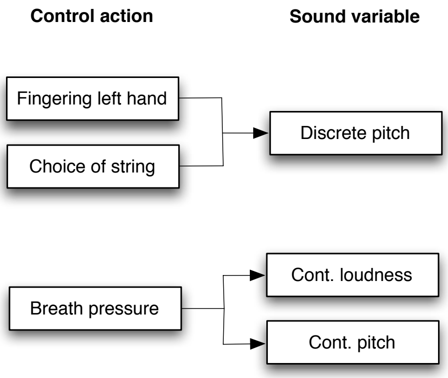


Figure 13: Examples of many-to-one and one-to-many mappings of input to output.

While there are a number of high-level characterizations available, there is a need for a more detailed typology of mappings. Traditional acoustical instruments may be a good starting point, because they represent a quite diverse set of possibilities that are well known through long practice.

One way is to make general overviews of mappings for a number of instruments, where control actions are mapped to the generalized musical parameters as described above. In [7], this is done for 23 instruments, and a few general points are made (one example in Figure 14).

With such overviews, it might be possible to create ‘gestural ensembles’ based on the gestures used to control an instrument. ‘Bowing’, ‘string-stopping’, ‘finger-hole-fingering’, ‘piano-type-keyboarding’, ‘concertina-buttoning’ etc. might be examples of such ensembles. Such entities will represent a specification of the control of single instruments, and at the same time be useful over a range of instruments; possibly useful also as a basis for a more general typology of gesture–sound mappings in musical instruments.

Also relevant to such a typology, is the more detailed model found in [8], where an analysis of how a single musical tone — a musical *event* — is controlled, is used to develop terminology for the description of what is called ‘musical control space’. Here, a musical event is broken

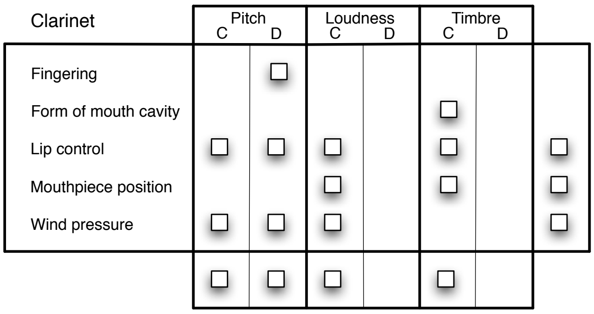


Figure 14: Mapping chart for clarinet. C and D means Continuous and Discrete (see text for explanation)

down into five phases, called *Selective preconditioning, Beginning, Middle, End* and *Terminus*, and analytical categories introduced for each phase.

All these approaches, however, raise a general problem. While input parameters may be relatively well-defined gestures, the same is not the case with output parameters. As discussed in section (3.2.2), meaningful descriptions of musical output in more detail depends heavily on musical style and experience with the musical instrument being discussed.

This means that development of a more general mapping theory should be closely connected to development in *music theory*. Without some kind of consensus on relevant musical entities, what they are, and how they are related, no coherent theory of mappings is possible.

4. CONCLUSIONS

As is obvious from the brief literary overview in this paper, there is no general agreement on terms for describing musical instrument design and mapping. This holds on almost every level of description, and almost every aspect of the models presented here. We believe that a common set of descriptors could be valuable for both analysts and constructors in the field, and may also help to unify various models being proposed, as well as clarifying real differences of opinion.

This paper has not addressed questions of how gestures can be formalized. This is discussed further in in [6].

5. REFERENCES

[1] D. Arfib, J.-M. Couturier, L. Kessours, and V. Verfaille. Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces. *Organised Sound*, 7(2):135–152, 2002.

[2] I. Choi. Gestural primitives and the context for computational processing in an interactive performance system. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 139–172. Ircam - Centre Pompidou, 2000. 352.

[3] E. F. Clarke. Categorical rhythm perception and event perception. In *Proceedings of the International Music Perception and Cognition Conference*. Keele University, Department of Psychology, 2000.

[4] O. Hellevik. *Forskningsmetode i sosiologi og statsvitenskap*. Universitetsforlaget, Oslo, 1971.

[5] A. Hunt, M. M. Wanderley, and R. Kirk. Towards a model for instrumental mapping in expert musical interaction. In *Proceedings of the 2000 International Computer Music Conference*, pages 209–212. International Computer Music Association,, San Fransisco, 2000.

[6] A. R. Jensenius, T. Kvifte, and R. I. Godøy. Towards a gesture description interchange format. *NIME06, Paris*, Fortchoming.

[7] T. Kvifte. *Instruments and the Electronic Age*. Solum Forlag, Oslo, 1989.

[8] D. J. Levitin, S. McAdams, and R. L. Adams. Control paramteres for musical instruments: a foundation for new mappings of gesture to sound. *Organised Sound*, 7(2):171–189, 2002. 344.

[9] D. J. Merrill. *FlexiGesture: A sensor-rich real-time adaptive gesture and affordance learning platform for electronic music control*. Master of science in media arts and sciences, MIT, 2004.

[10] M. Wanderley, B. W. Vines, N. Middleton, C. McKay, and W. Hatch. The musical significance of clarinetists' ancillary gestures: An exploration of the field. *Journal of New Music Research*, 34(1):97–113, 2005.

Vibrotactile Feedback in Digital Musical Instruments

Mark T. Marshall
Input Devices and Musical Interaction Laboratory
McGill University - Music Technology
555 Sherbrooke St. West, Monteal, QC, Canada
mark.marshall@mail.mcgill.ca

Marcelo M. Wanderley
Input Devices and Musical Interaction Laboratory
McGill University - Music Technology
555 Sherbrooke St. West, Monteal, QC, Canada
mwanderley@music.mcgill.ca

ABSTRACT

This paper discusses vibrotactile feedback in digital musical instruments. It compares the availability of intrinsic vibrotactile feedback in traditional acoustic musical instruments with the lack of vibrotactile feedback in most digital musical instruments. A short description of human sensory ability with regard to this form of feedback is given and the usefullness of vibrotactile feedback to musical performers is also briefly discussed. A number of devices are examined which can be used to provide vibrotactile feedback in a digital musical instruments and some experiments to evaluate these devices are also described. Finally, examples are given of a number of instruments which make use of some of these devices to provide vibrotactile feedback to the performer.

Keywords

Digital musical instruments, tactile feedback, vibro-tactile feedback

1. INTRODUCTION

Most traditional musical instruments inherently convey an element of tactile feedback to the performer in addition to their auditory and visual feedback. Reed instruments produce vibrations which are felt in the performer's mouth, string instruments vibrations are felt through the fingers on the strings, or through contact between the performer's body and the resonating body of the instrument [3]. This tactile feedback leads to a tight performer-instrument relationship which is not often found in digital musical instruments.

Studies have shown that while beginners make extensive use of the visual feedback provided by musical instruments, in expert performance it is the tactile and kinaesthetic which is the most important [7]. The majority of digital musical instruments provide only auditory and visual feedback to the performer, which results in a less complete sense of the instrument's response to the player's gestures than is available with traditional instruments [3]. It has also been stated that only the physical feedback from an instrument is fast enough to allow a performer to successfully control articulation [11].

This paper begins by discussing how we sense vibration in traditional musical instruments and goes on to discuss how the vibrations of a traditional instrument can be simulated in a digital musical instrument to enhance the "feel" of the instrument which results from these vibrations in acoustic instruments [6].

2. TACTILE FEEDBACK

Tactile (or vibrotactile) feedback results from contact between the body of the performer and the vibrating body of the musical instrument. Mechanoreceptors in the skin are sensitive to these vibrations. The fingers are capable of sensing vibrations in the region of 40 Hz to 1000 Hz and are most sensitive at 250 Hz [15]. These frequencies are within the audible range and are also frequencies which are among those produced by acoustic instruments.

As these vibrations are created by the resonating elements of the musical instrument in a traditional instrument and a digital musical instrument may not contain any resonating elements it is necessary to simulate the vibrations in order to provide some form of tactile feedback to the performer. In order to best simulate the vibrations of an acoustic instrument, the method used to provide vibrotactile feedback in a digital musical instrument should be variable in both frequency and amplitude and should be directly related to the sound the instrument is producing [13]. These leads to certain requirements for such a device which are different from the requirements of system which use vibrotactile feedback for information communication, which often use amplitude or location of vibrations as indicators and are generally fixed in frequency (for example [5] and [8]). These requirements are:

- Wide range of requency repoduction (at least 40–1000Hz)
- Control of amplitude of vibration
- Fast transient response
- Easy to control from a synthesis system (i.e. controlled using a signal rather than a complex protocol)

3. DEVICES FOR VIBROTACTILE FEEDBACK

A number of different types of devices are available to produce vibro-tactile feedback in digital musical instruments. These include:

- Tactors
- Piezo-electric elements
- Voice coils
- Motors
- Solenoids

3.1 Tactors

A tactor is a device containing small plates which can be moved electrically to create vibrations. A tactor was used in experiments in adding tactile feedback to the LaserBass [2], but was found to offer too low an amplitude of vibration and to have a small delay when driven, which made it unsuitable for a vibrotactile element of a digital musical instrument. Tactors are commonly used to convey information to users in simulations and interfaces for the blind [17] or tactile information systems [5].

3.2 Piezo-electric elements

Piezo-electric elements are crystal elements which vibrate when an electrical current is applied to them. Normally used as sound producing devices in low-cost buzzers, they can also be used as vibro-tactile transducers. They do not seem to have found use as vibro-tactile sound producers in digital musical instruments, but have been used in other tactile displays, such as the Optacon system for tactile representation of text [9]. Other tactile interfaces have tried to use piezo-electric elements, but found the sound generated to be too loud for their requirements [5].

3.3 Voice coils

Voice coils are coils held in the field of an electromagnet which vibrate when an AC current is applies to the electromagnet. Voice coils have been used to provide tactile feedback in a number of digital musical instruments, including [3], [12] and [1]. As they can be driven using the same audio signal as that creating the audio output of the instrument they are easily used as vibro-tactile devices in digital musical instruments.

3.4 Motors

Miniature motors can be made to rotate at different speeds, which combined with an unbalanced shaft can cause vibration. They are commonly used as vibrational alarms in pagers and mobile phones and can also be found in many vibrotactile game controllers.

3.5 Solenoids

Solenoids have pins which are forced in and out of the solenoid body by applying a DC voltage and can be used to produce vibrations. The Tactile Ring [2], which contains a miniature solenoid has been succefully used to add tactile feedback to a number of instruments, including the LaserBass and the SonoGlove.

4. EVALUATING DEVICES FOR VIBROTACTILE FEEDBACK

In order to evaluate which of these devices might be suitable for use in providing vibro-tactile feedback in digital musical instruments, a series of experiments were run. These experiments attempted to determine the range of amplitudes of vibration which each device could generate, the range of frequencies of these vibrations and the transient response of the devices when attempting to move from one frequency to another and one amplitude to another. This would allow us to see which of the devices are capable of meeting our requirements. All the devices used (with the exception of the tactor) were off the shelf components, which were not specifically designed for use as vibrotactile feedback devices.

4.1 Methodology

In order to accurately measure the amplitude and frequency of the vibration of each of the devices an apparatus was built making use of a 2-dimensional accelerometer mounted to a small board. This board was placed in contact with the active area of each device and so vibrated with the devices. The accelerometer produces two voltages, which are proportional to the acceleration in each of the devices two axes. These voltages, along with the input signal being sent to each device (either a varying DC voltage, a varying frequency sine wave or a pulse width modulation (PWM) signal) were logged using a National Instruments DAQ and Labview 2.1 software, operating at a sampling frequency of 10 kHz. The logged data was then analysed using GNU Octave.

The data logged allowed for the analysis of a number of aspects of each of the devices. The ability of each device to output frequencies in the 40Hz to 1000Hz range was tested, along with the maximum amplitude of vibration the device could create for frequencies in this range. This gives a measure of the usefulness of the device for creating vibrations at the frequencies which are felt through the fingers. Next, the range of amplitudes which each device can create was measured for a series of frequencies in this band, giving a measure of the amplitude control available with each device. Finally, the transient response of each device was tested for changes in frequency of 100Hz, 200Hz and 500Hz.

4.2 Results

Table 1 indicates the results found for each device during the testing along with some other important characteristics of typical devices of each type. As can be seen from the table, all of the devices are capable of reproducing the necessary frequencies of vibration (although many tactors display a peak in the frequency response at 250Hz, the frequency of vibration to which the skin is most sensitive). The motor and the solenoid display worse transient responses than the other devices. Examining the amplitude response of the three audio signal-driven devices shows that the voice coil and the tactor typically give the largest range of vibration and the voice coil can also produce the strongest vibration of all of the devices.

Another item of note is that the tactor's are generally capable of lower amplitudes than the other devices, perhaps making them more suited to applications where they will be mounted in direct contact with the skin, rather than through another surface. Also of interest is that the amplitude and frequency output of the motor are inherently linked and that many solenoids are incapable of changing the amplitude of vibration.

Simulating the vibration of an acoustic instrument requires the ability to reproduce a range of vibration frequencies. While performers may not be able to accurately discriminate between a large number of frequencies, some ability to distinguish gross frequency cahnges does exist. In fact, the ability to distinguish between different frequencies has been shown to range anywhere from between 3 to 5 distinct values between 2 and 300Hz [12] to 8 to 10

Table 1: Comparison of results for each device

	Tactor	Motor	Solenoid	Piezo-electric element	Voice coil
Frequency response (over tactile range)	40–1000Hz (peak at 250 Hz)	40-1000Hz	40–1000Hz	40–1000Hz	40–1000Hz
Maximum amplitude	low	high	high	high	high
Amplitude range	good	good	single value	good	good
Amplitude and Frequency Control	independant	dependant	only frequency	independant	independant
Transient response	good	poor	good	excellent	excellent
Driving signal	Audio signal	PWM signal	PWM signal	Audio signal	Audio signal
Typical size	3cm dia., 0.7cm height	0.6cm dia., 1.5cm length	1.5 cm dia., 2cm length	2.5cm dia., 0.3cm height	from 1cm dia. up to 20cm dia.
Availability	uncommon (from manufacturer)	common	common	common	common
Typical Price	US$60	US$3	US$5	US$2	US$2 – US$100

distinct values between 70 and 1000Hz [16]. A qualitative difference has also been found in tactile perception of frequencies above and below 100Hz, with a different sensation being reported for frequencies in each range [16].

This would seem to indicate that for a vibro-tactile feedback system in a digital musical instrument the ability to reproduce a range of frequencies is important and so a device capable of this might be most suitable for providing this feedback. The tactor, the piezo-electric element and the voice coil devices can cover the frequency range. The voice coil offers a greater amplitude range and maximum amplitude output, but along with the piezo element, will also generate sound, which the tactor will not do.

5. USING VIBROTACTILE FEEDBACK IN DIGITAL MUSICAL INSTRUMENTS

A number of instruments and controllers have been built which make use of vibrotactile feedback to improve the interaction between the performer and the system. Chafe used a voice coil to simulate vibrations in the mouthpiece to help players control a physically-modelled brass instrument [3]. The VR/TX system [12] made use of vibrotactile feedback to augment a glove-based non-contact system, again using voice-coils. Bongers [2] discusses a number of systems which were augmented with a Tactile Ring, which uses a miniature solenoid to provide tactile feedback to the performers.

Each of these systems made use of vibrotactile feedback to the performer, in many cases providing this feedback through an audio-driven device. With some of these, as with many other tactile feedback systems, attempts were made to cover the audio output from the device, so that it would not be heard by the performer. The following sections detail a number of instruments which have been developed to include vibrotactile feedback, which make use of voice coils to provide this feedback, but rather than attempting to quiet the devices instead make use of this sound output to provide both vibrotactile and audio feedback to the performer. The aim is to produce an instrument which has a "feel" most like that of an acoustic instrument [2], by integrating the sound production into the instrument, giving vibrotactile feedback which is directly related to the sound being produced [13] and producing a sound output which is local to the instrument rather than being created at another point by a speaker system.

5.1 The Viblotar

Figure 1 shows the Viblotar[1][10]. It is an instrument designed to be played in a similar fashion to a traditional monochord and can be played on the performers lap or on a table or stand. The synthesis engine for this system consists of a physical model running in the Max/MSP environment. The model comes from the PeRColate [14] externals which is a port to Max/MSP of instruments from the Synthesis ToolKit (STK) [4] [2]. The physical model used is a hybrid model called the *blotar* which is a hybrid of an electric guitar model and a flute model.

Figure 1: The Viblotar

The Viblotar is played using the right hand to both select and excite pitches. It has a range of 3 octaves of continuous pitch, which are played using a linear position sensor. Excitation is caused by the pressure of the hand which is selecting the pitches. This allows for dynamic control of both pitch and amplitude using a single gesture. Two pressure sensors are also available to the left hand, to allow for pitch bend and vibrato effects.

The audio and tactile feedback to the performer is created using a pair of small 1W BTL amplifier circuits to drive a pair of 8 Ω 3W speakers. The body of the Viblotar functions as a resonating box and has been designed to maximize the frequency output of the speakers, from the determined small signal parameters of the speakers. As the audio output from the synthesis engine is used to drive these speakers, both the audio and vibrotactile feedback to the player are directly related to the sound being produced and so create a more tightly coupled interaction between the performer and the system.

[1] http://www.music.mcgill.ca/~marshall/projects/viblotar
[2] http://ccrma.stanford.edu/software/stk/

5.2 The Vibloslide

The Vibloslide is a small electronic wind instrument. It is a monophonic instrument, but unlike many wind instruments it is played using a continous position sensor rather than discrete keys or holes. This allows it to produce any pitch over an octave range and to produce effects such as glissando's which are not available with many wind instruments. A small piezo-electric film element is mounted at one end of the tube and is used to detect air being blown into the tube, to control the excitation and dynamics of the sound. Overall, this allows for a performance technique similar to a traditional slide whistle. Figure 2 shows the Vibloslide.

Figure 2: The Vibloslide

Again, in order to give tactile feedback to the performer, a small speaker is mounted at the far end of the tube. This is driven by the synthesis system through another small 1W BTL audio amplifier. The generation of the sound output therefor occurs at the instrument itself and the resulting vibrations in the instrument body can be felt by the performer through the fingers and the lips. This results in a similar feeling to the Touch Flute [1], which use a number of small voice-coil actuators to produce tactile feedback, but with only a single actuator to provide the vibration and the addition of integrated sound production.

5.3 Discussion

While a number of instruments have been developed which incorporate vibrotactile feedback and many of these instruments have made use of voice coils to generate this vibration, not many seem to have also used the voice coils to generate sound from the instrument. The addition of one or more small speakers and amplifiers to a digital musical instrument can provide the benefit of both integrated sonic and vibrotactile feedback and lead to a tighter performer-instrument interaction loop. However, as the sound quality of many small speakers and amplifiers is quite poor, an means of connecting the sound output from the synthesis system to external amplifiers and speakers is also necessary to allow for maximum sound quality.

The two instruments described here each make use of speakers in this way. People who have played these instruments comment on the feeling of a "complete" instrument rather than a controller. As expected the addition of vibrotactile feedback and sound production in the instruments give a "feel" which is more like that of a traditional instrument and less like a computer interface.

6. CONCLUSIONS

This paper discussed vibrotactile feedback in digital musical instruments. It compared the vibrotactile feedback available in traditional acoustic musical instruments with the lack of this feedback in many digital musical instruments. A number of devices which could be used to simulate the vibration of an acoustic instrument were instroduced and compared. Finally two instruments were introduced which have been developed and which make use of integrated audio speakers to produce both audio and vibrotactile feedback for the performer. This additional feedback to the performer would seem to improve the "feel" of the instrument, so that it is associated more with being an instrument rather than a computer controller.

7. ACKNOWLEDGEMENTS

This work was funded by a student research grant from the Centre for Interdisciplinary Research in Music Media and Technology and a research assistantship from the Natural Science and Engineering Research Council of Canada.

8. REFERENCES

[1] D. Birnbaum. The touch flute: Exploring roles of vibrotactile feedback in musical performance. Technical Report (Unpublished), McGill University, 2003.

[2] B. Bongers. Tactile display in electronic musical instruments. In *IEEE Colloquium Developments in Tactile Displays*, London, UK, January 1997.

[3] C. Chafe. Tactile audio feedback. In *Proceedings of ICMC 1993*, pages 76–79. ICMA, 1993.

[4] P. Cook and G. Scavone. The synthesis toolkit (stk). In *Proc. of the International Computer Music Conference*, Beijing, China, 1999.

[5] F. Gemperle, N. Ota, and D. Siewiorek. Design of a wearable tactile display. In *Proc. of the Fifth International Symposium on Wearable Computers*, Zurich, Switzerland, October 2001.

[6] B. Gillespie. Introduction to haptics. In P. R. Cook, editor, *Music, Cognition and Computerised Sound: An Introduction to Psychoacoustics*, pages 229–245. MIT Press, Cambridge, MA, USA, 1999.

[7] S. W. Keele. *Attention and Human Performance.* Goodyear Publishing Company, 1973.

[8] D. Kontrarinis and R. Howe. Tactile display of vibratory information in teleoperation and virtual environments. *PRESENCE*, pages 387–402, 1995.

[9] J. G. Linvill and J. C. Bliss. A direct translation reading aid for the blind. In *Proc. IEEE*, volume 53, pages 40–52, 1966.

[10] M. T. Marshall. The viblotar - a big box that makes noise (and vibrates too!). Technical Report (Unpublished), McGill University, 2005.

[11] M. Puckette and Z. Settel. Non-obvious roles for electronics in performance enhancement. In *Proc. of the International Computer Music Conference*, pages 134–137, San Francisco, USA, 1993.

[12] J. Rovan and V. Hayward. Typology of tactile sounds and their synthesis in gesture-driven computer music. *Trends in gesture control of music*, 2000.

[13] A. Tanaka. Musical performance practice on sensor-based instruments. *Trends in gesture control of music*, 2000.

[14] D. Trueman and R. L. DuBois. Percolate. http://music.columbia.edu/PeRColate.

[15] R. T. Verillo. Vibration sensing in humans. *Music Perception*, 9(3):281–302, 1991.

[16] R. T. Verillo and G. Gescheider. Perception via the sense of touch. *Tactile aids for the hearing impaired*, pages 1–36, 1992.

[17] Y. Yanagida, M. Kakita, R. W. Lindeman, Y. Kume, and N. Tetsutani. Vibrotactile letter reading using a low-resolution tactor array. In *Proc. of 12th Int. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 400–406, Chicago, IL, USA, March 2004.

Paper FSRs and Latex/Fabric Traction Sensors: Methods for the Development of Home-Made Touch Sensors

Rodolphe Koehly
Input Devices & Music Interaction lab.
McGill University, Montreal, Canada
rodolphe.koehly@mail.mcgill.ca

Denis Curtil
EFPG – Materials Science
INPG, Grenoble, France
denis.curtil@efpg.inpg.fr

Marcelo M. Wanderley
Input Devices & Music Interaction Lab.,
McGill University, Montreal, Canada
mwanderley@music.mcgill.ca

ABSTRACT
This paper presents the development of novel "home-made" touch sensors using conductive pigments and various substrate materials. We show that it is possible to build one's own position, pressure and bend sensors with various electrical characteristics, sizes and shapes**,** and this for a very competitive price. We give examples and provide results from experimental tests of such developments.

Keywords
Touch sensors, piezoresistive technology, conductive pigments, sensitive materials, interface design

1. INTRODUCTION
Many technologies are used to develop sensors, most commonly using conductive inks, and numerous musical interfaces were developed with industrial sensors such as Interlink *Force Sensing Resistors* (FSR) or Tekscan's Flexiforce, among others. Although their electrical behaviour are not necessarily the same [1], commercial sensors have basically the same drawbacks: they are available in few normalised sizes and shapes, with pre-defined electrical characteristics. Researchers and interface designers then need to adapt the characteristics of their interface to these existing sensors. Many times, their work could certainly be more efficient if custom sensors were available.

2. HOME-MADE SENSORS-MATERIALS
In commercial touch sensors – force/pressure, bend and position sensors – conductive inks are used to make a surface or a whole material conductive. The idea is to mix pigments into a medium that will provide the desired electro-mechanical behaviour of the device. The process is similar in the development of home-made sensors.
The first type of material that we studied was conductive ink: Considering the fact that it is hard to find and expensive, it appeared much more clever to produce one's own conductive ink from carbon black pigments and various basic mediums like polyvinyl acetate, varnish or liquid black inks such as china inks or ink-jet inks[4]. As long as you reach a sufficient pigment concentration, you can easily produce a conductive ink with any medium.

The choice of medium is fundamental: a) it must keep the ink liquid enough to enable a uniform printing, b) it must remain flexible when dried (to be used on flexible surfaces), c) it must bond the pigments to the support efficiently. Also, the choice of the pigments, of the inking support and of the printing process (that can be manual) will influence the printed result.
It is important to notice that industrial conductive inks only exist for industrial printing process and not for personal printers using ink-jet or laser technologies. Indeed, the ink used for such machines are not conductive because their pigment concentration is too low or because their pigments are not conductive. The following table gives examples of the resistance we can reach on a coated paper.

Table 1 Ink resistance for various pigments' concentrations: we reached a saturation point for 16% pigments.

% of Carbon Black Pigments in China Ink	Obtained surface resistance for 5*1cm roll printed samples
2 %	0.2-0.4 MΩ
5 %	30-50 KΩ
8.5 %	9-11 KΩ
13.5%	1.8-2 KΩ
16% and more	1.6-1.8 KΩ

3. EXAMPLES OF HOME-MADE SENSORS

3.1. Linear Touch Potentiometer
Video Tape is a well-known material for home-made linear touch potentiometers. In fact, it is just a thin inked polymer strip: the ink is conductive and printed by a machine. Figure 1 shows the resulting evolution of the resistance with the tape length measured on a JVC sample with a multimeter.

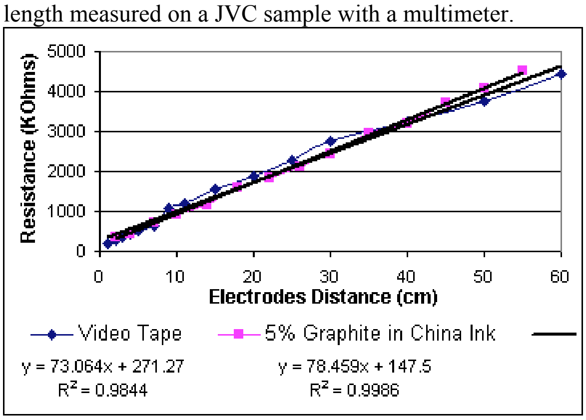


Figure 1 Comparison between 0.5 in width Video Tape and Inked band (the resistance of the inked material was multiplied by 20 for better comparison)

Videotape has a remarkably high resistivity, around hundreds of kΩ/cm (conductivity of 10^{-5} S/cm) and the printed process used to make it, provides it uniform conductivity. Furthermore, videotape is very fragile: any scratch on the inked surface will alter the resistance linearity.
It is then really easy to build a linear potentiometer such as the one represented below in figure 2[1]. In fact, several examples of position sensors using videotapes are available on the internet[1].

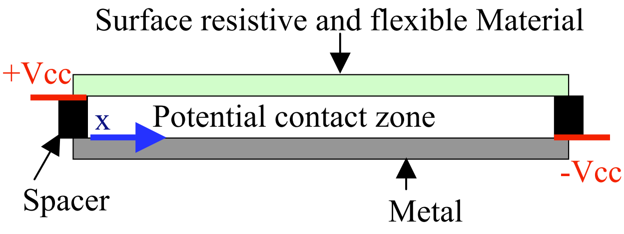


Figure 2 Home-made linear potentiometer along the x-axis

Separated connexions are placed on opposite ends of each band. The sensor resistance increases with the distance of the compressed zone from the connector on the resistive band.
Video tape is the best disposable material for such a sensor. However it can only be found in limited sizes. Home-made conductive surfaces were successfully performed replacing videotape by other conductive materials (tissues, papers, printer rolls,..), inked band or even pigments stacked on tape (less efficient and dirty) and enabling a much wider range of size possibilities.

3.2. Home-made Bend Sensors

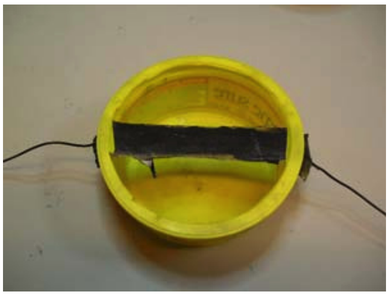


Figure 3 Multi-Couch Bend sensor: a pigment line is stuck between two latex layers making a sort of tube

Initial bend sensors were developed using conductive ink onto a flexible and elastic substrate (a 0.2 inch thick PET polymer sheet), the bending of the support increasing the ink resistance. This trial was not very successful, as we could not find any medium that would prevent the dried ink from cracking under strain.
We replaced the ink medium by elastic polymers such as latex, and silicone-based products. There was no need for support any more as the pigments were mixed with the polymers and moulded into the appropriate shape. A similar method was used by Mikael Fernström for the development of a pressure sensitive floor called "ForSeFIELDS" [3]. However, this method was not completely successful, as it required a high pigment load (from 15 to 20%) to prevent from the insulating effect of the various polymers. For such a pigment loading, the resulting material looses much of its mechanical strength.

Because pigments were insulated when mixed with polymers, we found it was a better idea to insert a pigments line inside 2 polymers layers. The polymers would then provide the material elasticity needed and confine pigments into a defined zone. Figure 3 shows a similar sensor mounted on a plastic support.
Traction along the sensor's length decreases the local pigment concentration along the line, thus decreasing conductance. Figure 4 shows the evolution of this conductance with a traction of 50% of the sensor length at initial state.

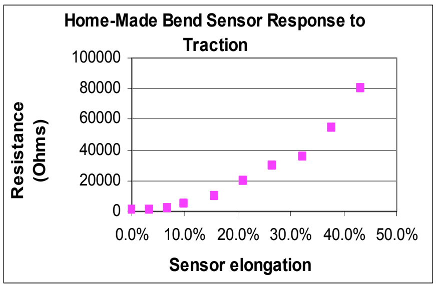


Figure 4 Multi-layer Sensor response to traction

We can see that for 50% traction, a continuous variation from 1 to 80 kOhms is obtained: although this sensor appeared to be very sensitive, the configuration did not provide a good repeatability. Indeed, pigments are quite free inside the tube and so any mechanical stress on the sensor will randomly change the pigment distribution, therefore affecting its resistance. Moreover the choice of latex as elastic material is not optimal: Silicone or Polyurethane based polymers should provide less viscosity and fragility.
We then decided to use a porous and elastic matrix that was tinted with conductive pigments; such porous materials can be papers, foams, tissues or polymers.[2] The use of a fibre skeleton limits the distribution possibilities of pigments into the elastic material and therefore provides better repeatability. Latex or silicone (or any other flexible container) might then be used to embed the sensor and so protect and prevent it from variations due to temperature, moisture or bad manipulation, as well as preventing pigments from leaking from the sensor. Figure 5 is an example of a medical care bandage that was tinted with carbon black and specific chemical products (retention aids, flocculants...) to improve the pigments bonding. The sensor is set on three various positions into a transparent plastic envelope.

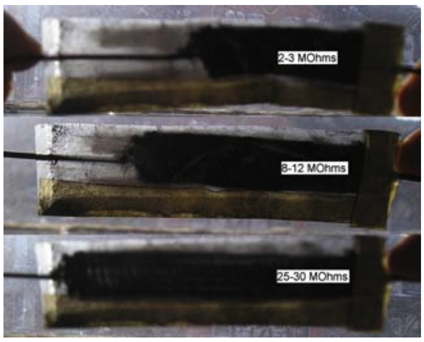


Figure 5 Tissue Bend Sensor under Strain: the resistance increases with traction

[1] Websites for instructions on how to build linear position sensors:
http://www.electronicpeasant.com/projects/ribbon/controller.html
http://www.geocities.com/tpe123/folkurban/synthstick/synthstick.html
http://www.angelfire.com/music2/theanalogcottage/ribcont.htm

[2] Many fibres can bond pigments through hydrogen bonding or other chemical forces: a process that is more complex than mixing pigments with mediums, requiring specific chemical component used in the paper or textile industry. This implies that research with porous materials be realized in specialized research centres.

The repeatability of this material was still not optimised, as the tissue was not enough elastic. However, "smart textile" development recently became a highly active research field ("WEALTHY" European Project [7]) and strain fabric sensors were already successfully performed by other researchers, such as F. Lorussi & Al [8].

4. HOME-MADE PRESSURE SENSORS

Various techniques can be used to develop touch sensors sensitive to force and/or pressure. One of these techniques consists in embedding pigments into latex or other materials. Some of these sensors provided excellent sensitivity, as shown in figure 6.

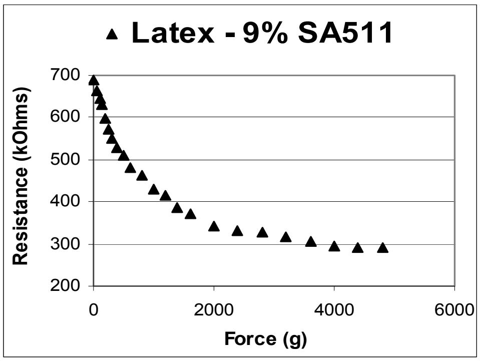


Figure 6 Latex Pressure Sensor [4]

Nevertheless, depending on the pigments used and on their concentration, the mixing process was not always successful.

Tinted porous materials appeared to be more efficient for pressure sensing. Any tinted polymer or organic foam, tissue or fibre network could provide an efficient pressure sensor, depending on its elasticity. However, the tinting process is not obvious and it requires some chemical knowledge and the use of specific products to optimise the bonding of pigments into the fibres.

We focused on tinted paper and succeeded in developing efficient paper pressure sensors, as seen in figure 7: as paper is around 50% porous and quite compressible and elastic, the compression of a stack of tinted sheets will increase the contact between pigments, thus increasing conductivity inside the sheets thickness.

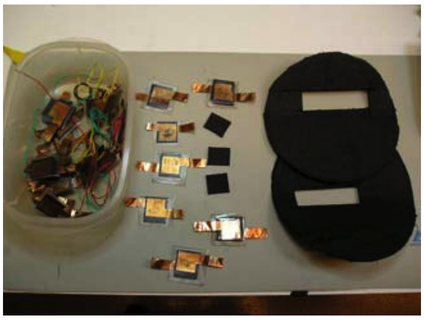


Figure 7 Home-made paper pressure sensors

As you can see, the design can be pretty simple : the sensitive material basically just requires connectors on each side of the pressure direction which can be made with metal foils or totally conductive ink. A plethora of tricks it would be too long too add here can be easily found using more complex designs. Also a major point of discussion would be connection systems as it must not generate noise. Soldering is not possible with ink, paper and some metal foils. Hopefully, one can find efficient alternatives such as mechanical connecting systems.

5. THE EFFICIENCY OF HOME-MADE SENSORS

The quality of sensors for musical performance can be evaluated through a few parameters: linearity, repeatability, resolution, drift and time-response. Home-made sensors' performance will mainly depend on material uniformity, design, and connexions' quality.

We can already notice from figure 1, that it is feasible to create surface conductive bands with home-made conductive inks, their range and linearity being sufficient for the making linear touch potentiometers.

Home-made bend sensors are not optimised in terms of repeatability: however there is a good potential for multi-layer latex bend sensors, as shown in figure 8.

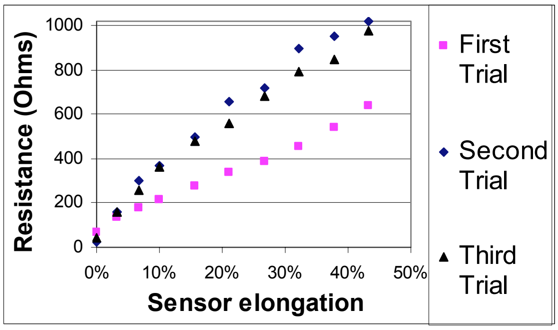


Figure 8 Repeatability of a multi-layer bend sensor response to 3 consecutive tractions [5]

The response is highly linear and the range is quite sufficient to be treated through a simple tension divider. Unfortunately, it presents a slow time-response (around 3s max). We can also see that the first trial is different from the other 2, a fact that is quite frequent with visco-elastic materials. During the first trial, we reached a plastic deformation state: the limit of elastic deformation of the material is then moved to this point and then as long as we remain in this deformation limit we can hope to get repeatable results. This sensor can then be used efficiently to sense, for instance, the bending of a finger, of the elbows or of the knees.

Results concerning paper pressure sensors are also quite impressive. A special laboratory testing machine was designed at EFPG, Grenoble, to observe the evolution of the sensor resistance under compression between 100g and 5kg on a 10*2cm paper sample with clipped copper foils as connectors on each end of the paper. Figure 9 shows the testing results.

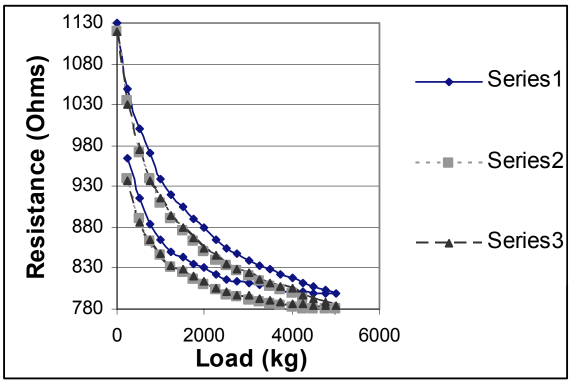


Figure 9 Repeatability testing of a paper pressure sensor under 3 series of gradual strain from 0 to 5 kg and return [6].

After a first trial during which there was a plastic deformation of the paper (like for the bend sensors), the following tests were similar: the sensor has an excellent repeatability, comparable to commercial ones [1].
Let's first notice the poor range we get in comparison with the range announced for Interlink FSR. The resistance range given in figure 8 would only provide 0.5 volt range with a simple voltage divider. Hopefully, a Wheatston bridge or an Op-Amp would increase this range. Also, more complex designs of paper and spacer stacks, as well as paper repulping enabled to provide an even more linear range of up to 4 volts (out of 5 in input) with a simple 1kOhm tension divider.
We can notice from this experiment that the hysteresis represents at worse 15% of the full range, which is better than for the Interlink FSR sensors (15 to 25% variation specified in the Interlink FSR user guide).
Second, the force resolution, which is the smallest measurable difference in force, is also competing with the FSR resolution. Even for high loads, there is a difference of 0.3 Ohms for 5g variation. We can then say that the resolution is lower than 0.1% of the full scale (5/5000g). The resolution given for FSR is 0.5% of the full scale.
A third experiment, drift testing, consisted in placing a 1kg load on the sensor for 1 minute. The conductive response of the material stabilised in a few seconds and remained the same at +/- 2% during the whole time, which is once again comparable to FSR specifications.
Last, the time-response was measured through an oscilloscope: a 3 kg load was exerted on the sensor and the screen of the oscilloscope was shot during the load release. The following picture shows that it took 75ms for the paper to recover its original state. This time-response is not totally satisfying for musical applications, as for instance, if we used such a sensor for percussion sensing, it would enable to sense only 6 beats per second to satisfy the Nyquist theorem.

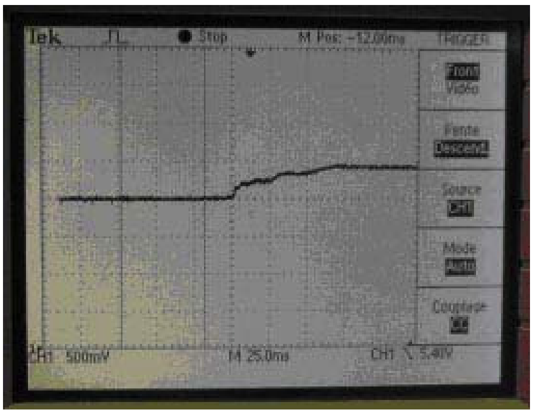


Figure 10 Time-response detection of a paper pressure sensor for 3kg load release.

6. CONCLUSION

To get its own custom sensors, it is worth spending some handwork time: this will make one save money and improve the quality and originality of the interfaces such as the one of Figure 11 [5]. These interfaces can even be granted an environmental-friendly label as one can use only recycled materials to produce them. This work shows once more that creativity not only exists in musical performance but also in development of technology.

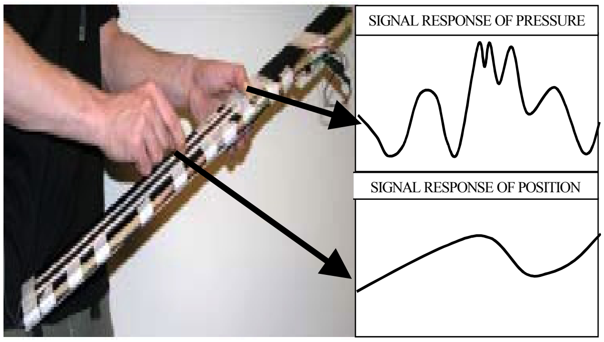


Figure 11 A performance with the Cheapstick, composed of one Paper pressure Sensor and one Linear touch potentiometer with Video Tape [7]

7. ACKNOWLEDGEMENTS

Many thanks to my various Tutors: Prof. Theodorus Van der Ven (McGill University) and Mr. Peter Herdman (ArjoWiggins, France). Thanks to various industrial providers for raw material: Sid Richardson for their Carbon Black Pigments, ArjoWiggins for fibres and advises, the EFPG and the Pulp and Paper Research Centre at McGill. Thanks also to several people for their precious help during the various testings and collaboration: Alexander Refsum Jensenius, Bertrand Scherrer, Darry Cameron, Thomas Hoffman, Stephane Dufreney and Yves Chaves.
Part of this research was funded by an Natural Sciences and Engineering Research Council grant to the second author (Discovery Grant).

8. REFERENCES

[1] F. Vecchi, C. Freschi, S. Micera, A.M. Sabatini, P. Dario, "Experimental evaluation of two commercial force sensors for applications in biomechanics and motor control", Proc IFESS Conf, Aalborg (DK), June 17-23, 2000.

[2] Lisa McElligott , Michelle Dillon , Krispin Leydon , Bruce Richardson , Mikael Fernström , Joseph A. Paradiso, "ForSe FIElds' - Force Sensors for Interactive Environments", Proceedings of the 4th international conference on Ubiquitous Computing, p.168-175, September 29-October 01, 2002, Göteborg, Sweden.

[3] R. Koehly, "Study of Various Technologies for Home-Made Sensors," *Master's thesis*, Sound Processing and Control Laboratory, McGill University, 2005, Master AST 2004-2005, Grenoble, France.

[4] M. M. Wanderley and R. Koehly. "Methods for the In-House Development of Sensors for Musical Applications." *Submitted for publication*.

[5] A. R. Jensenius, R. Koehly, and M. M. Wanderley, "Building Low-Cost Music Controllers." in *Proc. of Computer Music Modeling and Retrieval 2005 Conference*, Pisa, Italy, 2005.

[6] R. Koehly, "Study of Resistive Paper For Pressure Sensors 2nd step", *Internship Report*, EFPGrenoble France, December 2005.

[7] WEALTHY: Wearable Health Care System
http://www.wealthy-ist.com/

[8] F. Lorussi, A. Tognetti, M. Tesconi, P. Pastacaldi, and D. De Rossi, "Strain sensing fabric for hand posture and gesture monitoring," in P*roc.Int. Workshop New Generation Wearable Syst. Ehealth: Toward a Revolutionof Citizens' Health, Life Style Management?*, Lucca, Italy, Dec. 2003, [CD-ROM].

Creating Ad Hoc Instruments with Pin&Play&Perform

John Bowers
School of Music
University of East Anglia
Norwich, UK
+44-1603-592452
jmb78@btinternet.com

Nicolas Villar
Computing Department
Lancaster University
Lancaster, UK
+44-1524-510372
nvillar@gmail.com

ABSTRACT
Drawing from conceptual discussions and a reflection on extended techniques in musical performance, we offer the notion of an 'ad hoc instrument' as an instrument constructed during the course of interacting with it. To concretise the idea, we explore the utility of 'Pin&Play' technology where interface elements like dials, sliders and buttons are pinned to a conductive substrate. We describe an ad hoc sound mixer and ad hoc synthesizer, and report on our performance experience with them. We conclude by discussing ad hoc instruments in relationship to physical interfaces, live coding, 'living scores', and issues in contemporary music performance.

Keywords
Ad hoc instruments, Pin&Play, physical interfaces, music performance, new interfaces for musical expression.

1. INTRODUCTION

1.1 The Concept of an Ad Hoc Instrument
In this paper, we explore and demonstrate a realisation of the concept of an 'ad hoc instrument'. By this we wish to pick out instruments which are, in some significant way, constructed during the course of interacting with them. An ad hoc instrument is made and played at the same time. By interleaving performance with the fabrication of the instrument itself, one can explore extended possibilities for music performance. In our general concept of ad hoc instruments, we do not discriminate between hardware instruments, software instruments, acoustic, electronic, or computational ones, or hybrids. What is of interest to us is how activities normally thought of as separate and sequenced in time (building an instrument then playing it) can be interleaved in ways which can be interesting for an audience. Just as, on some definitions, improvisation interleaves composition and performance, we seek ways of blurring the boundaries between otherwise separate musical activities.

Another way of seeing our concept of an ad hoc instrument is by means of contrast with the instrument destruction performances of artists such as those associated with the Fluxus group. Nam June Paik's *One For Violin Solo* involves the destruction of a violin as the performance. The composer remarked that he was intrigued by the possibility that a piece would feature an instrument's terminal sounds, rather than its typical ones. In a contrasting yet complementary fashion, we could see the building of an ad hoc instrument equally as performance but (in the concept's purest instances) of pieces where the instrument's *very first* sounds are featured [cf. 1].

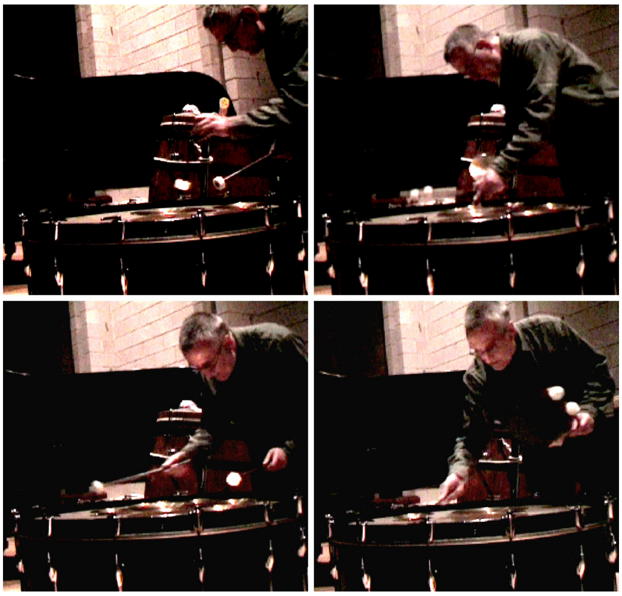


Figure 1. Eddie Prévost of The AMM constructing an 'ad hoc instrument' in improvised performance (see text).

1.2 Extended Instrumental Techniques
Of course, a degree of ad hoc construction is familiar from existing music practice. A mute may be placed over the bell of a brass instrument or a loop of guitar effects pedals switched in or out. Our notion of an ad hoc instrument arises at the extreme of these tendencies where somewhat less is fixed in advance. Modifications of an instrument and associated extended techniques are commonly explored in improvisation. Figure 1 shows four frames image-enhanced from a rather dark video of British improvisors The AMM in live performance. Top left, percussionist Eddie Prévost has just placed two small cymbals on top of a horizontally mounted bass drum and is exploring strategies for beating and damping the cymbals and affecting how they pass their energy to the resonances of the bass drum. Top right, about 30 seconds later, Prévost adds another cymbal to the surface. Bottom left, some two minutes further on, Prévost now has six cymbals of various sizes on the bass drum. With this packing of cymbals on the surface, they start colliding with each other as well as offering Prévost the opportunity of striking each one individually or in various combinations. After an intense minute or so exploring these complexities, the music begins to relax and Prévost starts removing the cymbals. Bottom right is about seven minutes after the first frame: two cymbals are being removed to leave just one. In

this episode, Prévost can be said to assemble and disassemble a complex percussion instrument which offers varying musical possibilities along the way. As he makes and unmakes his ad hoc instrument, so an episode is constructed in the music which, similarly, starts small, grows, and returns at the end to its components.

1.3 Configurable Interaction Devices

Based on these general reflections, our work explores the utility of a particular approach to building dynamically configurable interfaces for musical purposes, thereby making a particular kind of hybrid (software/hardware) ad hoc instrument. Research on physical interfaces that involve ad hoc composition and customisation by their users has an established history in fields such as Human Computer Interaction (HCI) and ubiquitous computing (ubicomp). For example, the Phidgets project [e.g. 4] works with a notion of customisable physical interfaces and provides widget taps to let the user bind physical controls to graphical controls, effectively as physical shortcuts for GUI applications. Behavior Construction Kits [12], Electronic Blocks [18] and Triangles [3] support a variety of functionalities (behaviour animation, information access, simple programming) through exploration of variable interface configurations sometimes in a playful fashion. Commercial products enabling the ad hoc construction of interfaces and control surfaces are emerging. For example, the ergodex input system [2] provides a way to easily arrange a number of buttons on a tablet to suit the user's ergonomic preferences. Though the design is proprietary, it is likely that RFID tags in the buttons are detected by a reader in the tablet.

From time to time, authors in the New Interfaces for Musical Expression (NIME) and allied research communities have explored instruments and other artefacts which manifest a degree of ad hocery. For example, the Flexipad element in Vertegaal and Ungvary's Sensorg [15] allows controls to be arranged on a metallic plate. However, Sensorg supports a somewhat limited number of controls which are all hardwired, constraining the ease with which they can be added to the ensemble and freely moved. As with ergodex, the motivation seems mainly ergonomic (e.g. allowing varied arrangements for ease of manipulation) rather than to explore a more extended utility of ad hoc interfaces for musical interaction.

BlockJam enables users to arrange blocks to construct musical structures [9] while Audiopad [11] also affords a level of ad hoc interaction, allowing the performer to compose by manipulating the arrangement of tokens on a surface. ReacTable [6] and Round Table [5] involve the manipulation of physical artifacts on a table surface and have both found application in supporting collaborative performances from multiple musicians.

Bowers and Archer [1] entertain the possibility that the juxtaposition of incomplete or half-made 'infra-instruments' could be a viable form of performance. Like that paper, we are concerned with ways of reformulating the instrument design 'life-cycle', in our case making means for dynamically configuring interaction surfaces available in/as performance. We do this by extending the musical applicability of another technology reported at NIME '05: Pin&Play&Perform [16].

2. PPP: PIN&PLAY&PERFORM

Pin&Play&Perform (PPP) builds on the more generic Pin&Play platform [16]. Pin&Play deconstructs the interface into atomic interaction units or 'widgets'—such as sliders, buttons, dials and joysticks—and provides a mechanism that allows these elements to be freely arranged on a substrate material (Figure 2). The substrate is a flexible laminate, produced in sheets that can be cut to size and used to augment existing surfaces.

Widgets are equipped with small pin-like connectors that allow them to attach anywhere on the surface of the substrate, and in any orientation. This attachment is both physical and digital, in the sense that the substrate acts as a communication medium between the widgets and a connected computer. As soon as a widget is attached it becomes connected, detected, identified by the system and is ready for interaction. All physical interaction on the substrate (attachment, detachment and manipulation of widgets) is reflected as software events on the computer. The net effect is the ability to construct and modify a functional physical interface on the fly without interrupting its operation. A number of potential applications for Pin&Play are discussed in [14].

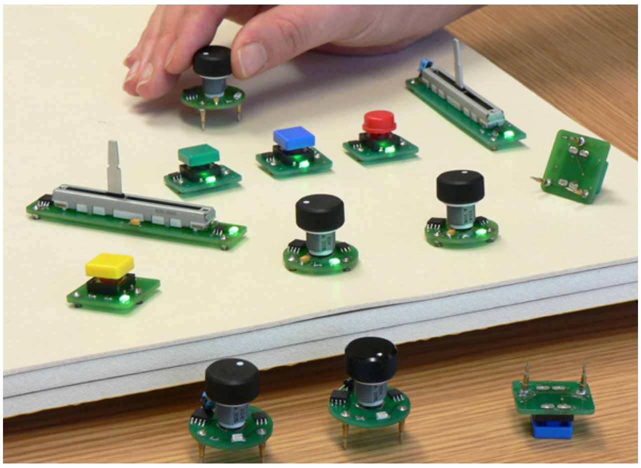


Figure 2. Pin&Play widgets can be easily inserted and removed from the substrate.

Pin&Play&Perform (PPP) [16] operates as a bridge between the underlying Pin&Play system and MIDI (and more lately OSC) enabled applications so as to support musical uses. [16] shows how the PPP approach supports the construction of interfaces which are very varied in their lay-out, thereby making for a high degree of customisation. In that earlier work, the imagined use scenario was one of setting up an interface to taste, or in cognizance of critical ergonomic concerns, and then using it in performance. However, this under-exploits the flexibility with which PPP can be configured to raise control messages in response to interaction events. PPP (and the underlying Pin&Play platform) can support a more radical interleaving of configuration and use. Accordingly, we sought to explore PPP as a platform for realising our emerging concept of ad hoc instruments.

2.1 Implementation

To demonstrate how ad hoc instruments could be made using PPP, we have built and performed with two demonstrators: a PPP mixer and a PPP synthesizer. Both of these allow one to incrementally build a Pin&Play performance interface without interrupting the music. In this way, we intend that the construction of an interface becomes part of the gestural repertoire of a performer. Furthermore, as we shall see, many of the background enabling actions that performers commonly have to do in advance of interaction (e.g. load patches or choose presets) can be elegantly folded in to their interface building and performance activity.

For clarity of demonstration, our PPP mixer and synthesizer are built from just three kinds of hardware widget—sliders, dials and buttons—though PPP can support several more. When a widget is inserted into the substrate, a characteristic MIDI message is raised from which can be extracted the type of widget it is, a unique numerical identifier, and a notification that the widget is attached and 'enabled' (i.e. ready to transmit data). The numerical identifier allocated is the smallest available. When a widget is interacted with, MIDI messages are raised from which can be extracted the widget type, its identifier, and the control data it is generating. When a widget is removed from the substrate, a MIDI message represents the widget type, its identifier and notifies that the widget is no longer enabled. This frees up the previously allocated widget identifier.

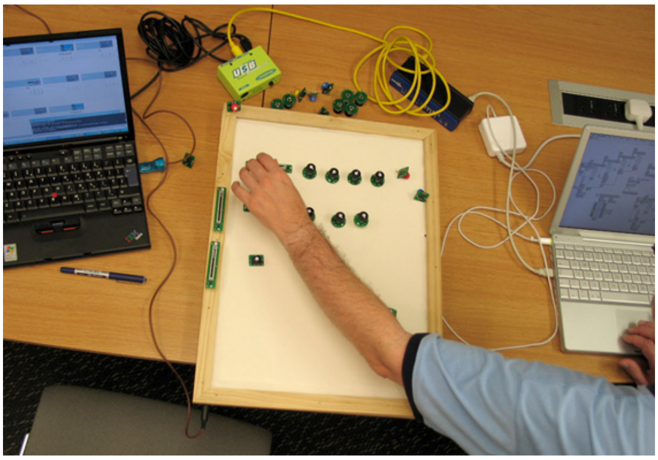


Figure 3. A Thinkpad (left) receives events from the substrate (centre), encodes them into MIDI messages and transmits them via a MIDI link (centre, top) to a PowerBook running Max/MSP (right).

In our demonstrations so far, an IBM Thinkpad X40 has been used to accept data from the substrate and generate MIDI messages which encode widget interaction events. These are passed via a conventional MIDI cable to an Apple G4 PowerBook running Cycling 74's Max/MSP. A patch (parseMIDI) receives the MIDI messages and extracts the information encoded in them—widget type (button, slider or dial), ID, enable/disable or control data values—and makes this available to other patches in the Max/MSP environment. Another patch (showStatus&Values) gives a graphical display of all messages received (both raw MIDI and interpreted).

2.2 PPP Mixer

PPP Mixer allows the performer to build a mixer interface interleaved with the performance of a mix. Up to four stereo soundfiles can be played back with amplitude control by means of sliders. In addition, each mixer 'channel' has associated with it a resonant filter, the centre frequency of which can be set with a dial. A typical interaction with PPP Mixer might proceed as follows.

Placing a slider on the substrate enables mixer Channel 1 and identifies a soundfile to be played back. The initial default amplitude is zero but this is supplanted when the slider is manipulated (Figure 4, top). The performer might, for example, raise the amplitude of the soundfile to moderate levels before inserting a second slider in the substrate. This would enable mixer Channel 2 and identify a second soundfile for playback (Figure 4, middle-top). Having adjusted the amplitude level to taste, the performer may then wish to filter Channel 1 (Figure 4, middle-bottom). A third mixer channel could then be created and so forth. Imagine now that the performer has created four mix channels, each with its own resonant filtering, and the music has reached its most dense moments. The performer may then wish to thin out the mix. Removing one of the sliders will stop the playback of the associated file (Figure 4, bottom). Removing a dial associated with a channel that is still playing will remove the effects of the resonant filter. A performance might be completed by taking the interface down to just one slider with its associated soundfile playing out. The performer could at any moment cut to silence by removing the last slider.

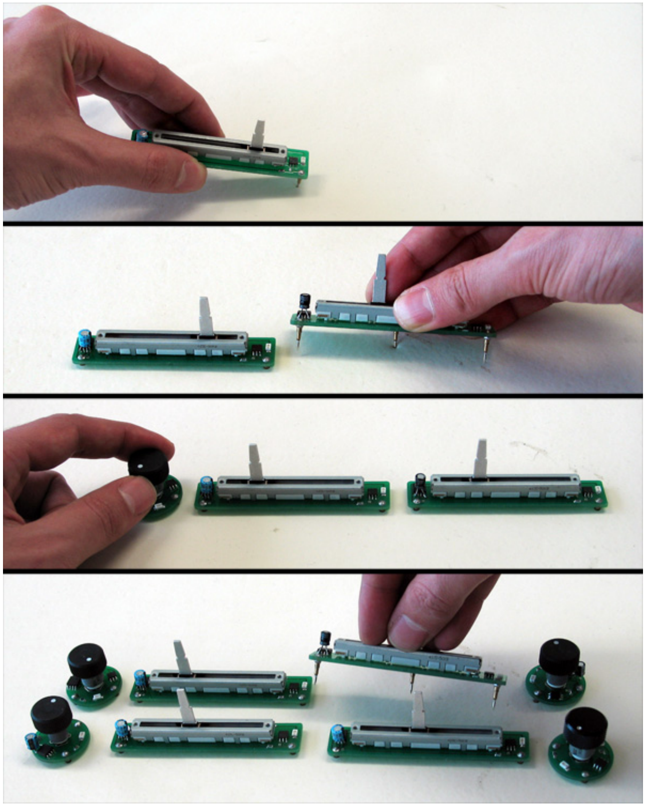


Figure 4. An example PPP Mixer performance sequence. Sliders start, stop and control the volume of associated soundfiles. Dials enable, disable and modify the centre frequency of a resonance filter for each file.

Naturally, we could have built our PPP Mixer application in many different ways, with different effects and default behaviours. It should also be clear from this exposition that the order in which soundfiles are enabled for playback and the resonant filtering effect becomes available is fixed in 'channel order'. This directly reflects the operation of our method for allocating identifiers to widgets. Our PPP Mixer application could be reprogrammed to allow different orderings and/or we could explore different methods for allocating identifiers if we sought different behaviour. Our point is not, however, to build a fully featured mixer with a Pin&Play interface at this stage. Rather, we seek to demonstrate the concept of an ad hoc interface through Pin&Play technology.

2.3 PPP Synthesizer

While we became aware that our PPP Mixer exhibited certain ordering constraints in the behaviours it was capable of, we wondered whether these could also be exploited in interesting ways. Our PPP Synthesizer demonstrates how a synthesis patch could be interacted with by means of incrementally building the interface to it. In contrast to the PPP Mixer, though, the order in which interface widgets are added is 'parsed' so as to further inform how the synthesizer should be configured. Placing a slider on an empty substrate makes available a sine wave oscillator with default amplitude and frequency. Manipulating the slider gives amplitude control. The next dial to be placed on the substrate will give frequency control for that oscillator (sweepable through the MIDI note number range). The next three dials will control phase modulation of the oscillator with the first controlling modulation depth, the next the frequency of a first modulator, and the last controlling the frequency of a second modulator which phase modulates the first (Figure 5, top). In this way, a synthesizer can be configured which has a single audible sine wave oscillator with two modulators cascading phase modulation. Placing a second slider on the substrate makes available a second 'voice' which can be incrementally added to in the same fashion as the first to create another cascade of phase modulation (Figure 5, middle).

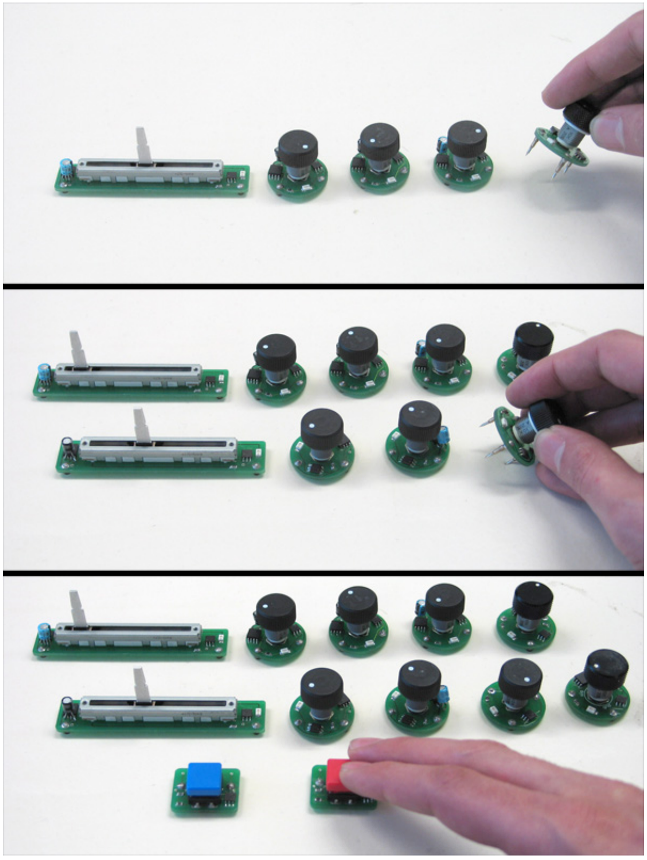


Figure 5. Constructing a PPP Synthesizer.

In our PPP Synthesizer, the exact significance of a widget depends (at least for the dials) on where in order they appear on the substrate. The first dial is a frequency control. Subsequent ones control various aspects of phase modulation. This contrasts with our PPP Mixer where a dial was always associated with the centre frequency of a resonant filter.

Our PPP Synthesizer can be completed by adding up to two buttons (Figure 5, bottom). These have fixed frequency oscillators associated with them. Pressing the button alternately turns the oscillator on and off. The time intervals between presses of the button are measured and used to automatically pulse the oscillator. This rhythm can be interrupted at any time and reset though three (or more) successive manual button presses or the pulsing can be stopped altogether by removing the button from the substrate.

Altogether then, the PPP Synthesizer has (up to) two frequency-variable oscillators which can be complexly phase modulated and two fixed frequency oscillators whose pulsing behaviour can be manually shaped. While this is a simple synthesizer, it is nevertheless capable of a variety of pulsing, rhythmic effects in a retro sort of fashion. The important point, however, is that it demonstrates how we can use Pin&Play technology to interface to synthesis, building interfaces as we configure the topology of synthesis units and do all that without interrupting the music. Furthermore, our PPP synthesizer shows how we can, in rudimentary ways, 'parse' the interface building activity of the performer to make more varied assignments between interface elements and their underlying function.

2.4 Spatial Arrangement of Widgets

The Pin&Play system can track the location of widgets by using additional sensors attached to the substrate. In our PPP Synthesizer example we have shown how meaning can be parsed from the sequence in which the interface is constructed. Additional mechanisms for implicitly extracting significance from the interface can be applied to the way that the components are arranged. Here are some possibilities we have demonstrated.

- The substrate can be regionalized so that different placements of the same widget can have a different significance (e.g. place a slider here and it's a mixer fader, here it creates an oscillator).
- The relative spatial arrangement of widgets (their location with respect to each other) can be used to specify associations between widgets. For example, a dial can become associated with the closest slider to it so that both widgets act together on the same underlying sound source or set of synthesis units. Further subtlety to this can be given depending on the above/below or left/right organization of placements. For example, a dial placed below a slider can control the playback rate of the soundfile whose amplitude is given by the slider, while placing the dial to the right of the slider might allow the dial to control a resonant filter.
- The surface location of a widget can be used to encode (at least) two dimensions of data which would otherwise have to be pre-set. For example, one could make specific file selections in a PPP Mixer in terms of where the fader is put or one can set the frequency of a PPP Synthesizer's pulsing fixed oscillator unit (e.g. higher up the substrate gives a higher pitch). Voodoo Cows, an installation by the second author, specifically exploits such features for pitch determination and sound spatialisation.

Generally, location sensing means that an application is less governed by constraints arising from the exact ordering in which widgets are placed on the substrate. Our simple PPP Mixer and Synthesizer demonstrations were constrained by such affairs (e.g. placing a slider on the substrate in PPP Mixer would always initiate the playing of the soundfile associated with the lowest available slider ID). While this is quite elegant in some applications, it is constraining in others. Being able to pick up location gives added flexibility.

3. PERFORMANCE EXPERIENCE

We have demonstrated our PPP Mixer and Synthesizer on a number of occasions. In particular, one of us (NV) performed as part of *Circuits of Malpractice*, a concert of new performance and installation work at the School of Music, University of East Anglia, Norwich, UK, 3rd October 2005.

In discussions with people that attended the performance we gathered that they enjoyed watching the process of 'construction'. All of the actions involving placement of widgets on the substrate were functional even if their significance was not immediately revealed (e.g. a slider would only be heard to control amplitude when it was moved). This gave the performance a kind of subtle legibility. Even if the exact mappings between action and effect were not immediately transparent, it was clear that the instrument was gradually, as one audience member put it, "written down on the blank sheet" as the piece grew and developed. It was interesting to find that, even though the underlying implementation was imagined to be complex or "very clever", the actual operation of the interface hardly needed explaining. Our audience members recognised the basic controls in our widget set and what can be done with them: knobs are for twisting, sliders are for sliding, buttons are for pressing. Sharp pins on the bottom of the controls and the soft, membrane-like rubberised substrate provide a strong sign of how the two can be used together. Amongst some musicians in our audience, the fact that the actual programming of the instrument is done in Max/MSP gave a sense that this was a technology that they could use and appropriate. It also encouraged speculation about how our applications could be tweaked and modified in their behaviour. The aesthetics of the components was appreciated—the mixture of electronics and mechanics involved. When people had a chance to play with the interface, someone commented on the pleasant tangibility of inserting a pin into the substrate describing it as "walking on snow".

A common suggestion from technically aware audiences and attendees at demonstrations was that we should be using OSC rather than the MIDI link we were using in our setup. This has been taken into account and the alternative protocol is now included in the PPP specification.

4. CONCLUSIONS

Drawing from conceptual considerations and a particular reflection on extended techniques in improvised music, we have offered our notion of an ad hoc instrument as one which supports the interleaving of instrument-construction and performance. To concretise our work, we have explored the utility of a hardware platform for building physical interfaces (Pin&Play) and examined a musical specialization of it (Pin&Play&Perform) which enables us to create simple kinds of ad hoc instrument (e.g. PPP Mixer and PPP Synthesizer). Let us review our experience.

4.1 Pin&Play Constraints

The Pin&Play platform currently has several constraints which impact upon its musical uses. There is a delay between interaction and resultant action due to the slow data-rates supported by the underlying network protocol. However, this is constant (<1s. for enabling/disabling, <200ms. for control events) and does not increase as more widgets become connected. Nevertheless, this means that there are certain musical gestures which cannot be adequately supported by Pin&Play at the moment. One cannot *rapidly* remove and reinsert a slider to punch a sound in and out of the mix, for example. Of course, it is arguable whether one should perform such gestures this way, rather than using a more appropriate widget already fixed in the substrate (a button in the example given). While delays on enabling/disabling are often tolerable, and indeed can encourage certain kinds of usage, the delays on control events are more annoying. We are currently investigating means for ameliorating them.

The idea of pins as a connection mechanism was inspired by Pushpin Computing [7] where pins are used to provide power to distributed sensor nodes. Pin&Play [3] builds on this by using pins to also provide a pathway for data communication. However, pins require a quite deliberative application to the surface. To get them to insert straight you need to choose your location and push them in with some care. This has benefits and problems. It makes for a durable construction which resists being unintentionally knocked but it prevents doing a sweep of the widget from one place to another without breaking connection. It also further inhibits the insertion or removal of widgets as a fast moving performance gesture.

An alternative could be to redesign the substrate to work like the network surface described in [13]. This subdivides the surface into an ingenious arrangement of tiles which enable an object to make the required connections no matter how it is placed on the surface. The Sensetable [10] platform provides a way to wirelessly track the position of objects on its surface, and has been used to implement the Audiopad [11]. Yet another alternative has been proposed in Magic [8]. A surface is embedded with a matrix of electromagnetic coils. Devices are equipped with small modules that allow them to communicate with the surface by means of magnetic induction. All of these would be alternative platforms for realising ad hoc instruments, each with their own technical idiosyncrasies and performance 'feels'. They wouldn't be like walking on snow.

4.2 PPP for Live Coding

Our concept of building an instrument while playing it has much in common with the philosophy of live coding. Indeed, Wang et al [17] discuss how, through writing code live, one can dynamically reconfigure controller mappings, reporting on successful experiments with commercially available control surfaces. However, one can go further and reverse this picture. The creation of an ad hoc control surface could become the means by which live coding takes place. That is, the substrate and widget-set could be simultaneously a means for editing code and performing it. In some respects our PPP Synthesizer simulates this for a very small synthesis universe. It is a simulation because we are activating code which has already been authored rather than properly coding live. With a richer set of widgets, and a more comprehensive approach to parsing user activity with them, one could readily create a live coding programming environment which was also a control surface construction kit. Indeed, this would unify our approach to ad hoc instruments not merely with live coding but also with the existing interest of researchers in physical programming interfaces [12, 3].

4.3 PPP and Living Scores

Another line of future interest for us is in the possibility of coordinating widget deployment with visual representations printed or projected on the substrate. Although much of our work with the PPP applications has been improvisatory, there are interesting possibilities involving pinning widgets into score-like representations to support the realisation of music with a notational/compositional element. Being able to locate widgets on the substrate, and hence their relation to whatever graphical entities are used in the score (and we are certainly not confining the discussion to common notation), is important here. In this way, we might be able to explore a notion of 'living scores', which are not so much instructions in how to realise the music (as scores are commonly conceived) but also the environment in which the music's instruments will be built, as well as providing a space for the gestural illustration of its form as an implicit part of performance. The use of moving image and projections onto the substrate could further enhance the liveliness of graphically notated material in such scenarios.

4.4 Making Performance Activity Legible

Much of the point of building new interfaces (or instruments) for musical expression is to enable a musician's performance activity to be made legible for an audience in new ways. This is sometimes seen as especially important for forms of music which might otherwise be difficult to follow. Our work with Pin&Play technologies adds to this concern in an interesting way. As a PPP Mixer or Synthesizer is being built in performance, in front of the performer and those audience members within perceptual range (and we do ensure that people can see what's going on) are just those interface elements which are needed for interacting with the music. In contrast to using a subset of the sliders or dials on a conventional controller and leaving some idle, the emerging and changing complexity of the interface, over the course of a performance, parallels and helps illustrate the complexity of the ongoing music. There are no surplus widgets to distract the performer or to enigmatically be left untouched. A performer's 'latitude for action' (variation, choice) is clearly displayed in terms of the available widgets: not just what you are doing but what you can immediately turn to. Bringing out a new widget presages an imminent change in the music, helping the audience anticipate and listen for transitions. The coming and going of dials and sliders can give a literal impression of the overall arc of the music and the return of the substrate to emptiness (and silence) ending a performance has been found aesthetically pleasing by many of our audience members.

It is a common complaint of contemporary digital music that watching a performer hunched over a laptop and a MIDI control box is a boring affair. In many ways, our task has been the re-enchantment of dials, sliders and buttons, to return a degree of fascination and intrigue to their use. By regarding these as elements to be worked with constructing an ad hoc instrument in the time of performance itself, we feel we have gone some way towards achieving this. The act of building the interface draws attention to the organization of the music and its relation to performer-activity, matters which are hidden in much conventional technology-rich performance. While we have explored just one way of making ad hoc instruments, we hope we have shown how this concept might help engender some innovative approaches to music making.

5. ACKNOWLEDGEMENTS

JB thanks the UK's Engineering and Physical Sciences Research Council (EPSRC) for a Research Fellowship which supported his contribution to the work reported here. NV thanks the Equator Interdisciplinary Research Collaboration also supported by the EPSRC.

6. REFERENCES

[1] Bowers, J. and Archer, P. Not hyper, not meta, not cyber but infra-instruments. In *Proc. of New Interfaces for Musical Expression (NIME 05)*, 2005.
[2] Ergodex Input System. http://www.ergodex.com/
[3] Gorbet, M. G., Orth, M., and Ishii, H. Triangles: tangible interface for manipulation and exploration of digital information topography. *Proc. CHI '98*, ACM Press, 1998.
[4] Greenberg, S. and Boyle, M. Customizable physical interfaces for interacting with conventional applications. *Proc. UIST '02: User Interface Software and Technology,* ACM Press, 2002.
[5] Hoch, M., Bowers, J., Jää-Aro, K.-M., and Hellström, S.-O. http://www.nada.kth.se/erena/table.html
[6] Kaltenbrunner, M., Geiger, G., and Jorda, S. Dynamic patches for live musical performance. In *Proc. of the New Interfaces for Musical Expression (NIME 04)*, 2004.
[7] Lifton, J., Seetharam, D., Broxton, M., and Paradiso, J. A. Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks. *Proc. Pervasive '02*, Springer-Verlag, 2002.
[8] Minami, M., Nishizawa, Y. Hirasawa, K., Morikawa, H., and Aoyama, T. Magic-surfaces: Magnetically interfaced surfaces for smart space applications. *In Adjunct Proc. Pervasive 2005*.
[9] Newton-Dunn, H., Nakano, H., and Gibson, J. Blockjam: A tangible interface for interactive music. In *Proc of New Interfaces for Musical Expression (NIME 03),* 2003.
[10] Patten, J., Ishii, H., Hines, J., Pangaro, G. Sensetable: a wireless object tracking platform for tangible user interfaces. *Proc. CHI '01*, ACM Press (2001), 253–260.
[11] Patten, J. Recht, B. and Ishii, H. Audiopad: A tag-based interface for musical performance. In *Proc. of New Interfaces for Musical Expression (NIME 02)*, 2002.
[12] Resnick, M., Behavior construction kits. In *Communications of the ACM* (1993), 36(7):64–71.
[13] Scott, J., Hoffmann, F., Addlesee, M., Mapp, G., and Hopper, A. Networked surfaces: a new concept in mobile networking. *In Proc. of Mobile Networks and Applications 2002*.
[14] Van Laerhoven, K., Schmidt, A. and Gellersen, H.-W. Pin and play: Networking objects through pins. *Proc. Ubicomp*, ACM Press (2002), 219–229.
[15] Vertegaal, R. and Ungvary, T. The sensorg: A musical cyberinstrument with cognitive ergonomical touch. In Proceedings of IEEE SMC'98 Interntational Conference on Systems, Man and Cybernetics. IEEE, 1998.
[16] Villar, N., Lindsay, A., and Gellersen. H. A rearrangeable tangible interface for musical composition and performance. In *Proc. of New Interfaces for Musical Expression (NIME 05)*, 2005.
[17] Wang, G., Misra, A., Kapur, A. and Cook, P. Yeah, ChucK It! => dynamic, controllable interface mapping. In *Proc. of New Interfaces for Musical Expression (NIME 05)*, 2005.
[18] Wyeth, P., and Wyeth, G. Electronic blocks: Tangible programming elements for preschoolers. *Proc. Interact 2001*, IFIP (2001).

Synthesis and control of everyday sounds reconstructing Russolo's Intonarumori

Stefania Serafin
Medialogy
Aalborg University
Copenhagen
Lautrupvang 15, 2750
Ballerup, DK
sts@media.aau.dk

Amalia de Götzen
Dep. of Information
Engineering
DEI/CSC,
Padova, Italy
degotzen@sci.univr.it

Niels Böttcher,
Steven Gelineck
Medialogy
Aalborg University
Copenhagen
Lautrupvang 15, 2750
Ballerup, DK
nboe05@imi.aau.dk

ABSTRACT

In this paper we introduce the Croaker, a novel input device inspired by Russolo's Intonarumori. We describe the components of the controller and the sound synthesis engine which allows to reproduce several everyday sounds.

Keywords

Noise machines, everyday sounds, physical models.

1. INTRODUCTION

At the beginning of the 20th century, the Italian composer and painter Luigi Russolo designed and built a family of new musical instruments which he called Intonarumori (noise intoners). Each Intonarumori was made of a colorful parallelepipedal sound box with a speaker on its front. Inside the box, a gut or metal string was excited by a rotating wheel. The speed of the wheel was changed by the player by using a crank, while the tension of the string was varied by using a lever. Such instruments were acoustic noise generators which allowed to simulate different everyday noisy sonorities.

The Intonarumori were a consequence of Russolo' s theories regarding the structure of the Futuristic orchestra. With the belief that the traditional orchestra needed some new sonorites, in his Futuristic manifesto *The Art of Noises* [12], Russolo proposed a taxonomy of noisy sounds divided in six families, organized as shown in Table 1. The different instruments designed by Russolo were a consequence of his Futuristic ideas. As can be seen in Table 1, the instruments were named according to the kind of sonorities they were able to produce.

Russolo's ideas were certainly very innovative: the composer was trying to design new interfaces for musical expression, to cope with the limitations of the traditional orchestra. Unfortunately, his ideas were probably too progressive for his time, so during his concerts the audience was merely laughing at his instruments rather than trying to understand the novelties introduced.

Moreover, during World War II all the original Intonarumori got destroyed. Since then, several attempts to rebuild such instruments were made. Among them, the ones shown in Figure 1 are some reproductions displayed at the exposition Sounds and Lights at the Pompidou Center in Paris in December 2004.

In this paper, we are interested in designing a controller and sound synthesis engine able to reproduce the different instruments designed by Russolo.

2. RUSSOLO'S INTONARUMORI

Figure 1: Different Intonarumori as shown in the exposition Sounds and Lights, Paris, Pompidou Center, December 2004.

Figure 2 shows Luigi Russolo and his colleague Ugo Piatti playing the original Intonarumori at around 1913.

As can be seen in Figure 2, each Intonarumori was made of a parallelepipedal box, with a crank and a lever in the outside. The player, by rotating the crank, was able to rotate a wheel placed inside the box, which excited a vibrating string. The string, stretched at the two extremities of the box, was attached to a vibrating drum connected to a radiating horn. By moving the lever back and forth, it was

Family 1	Family 2	Family 3	Family 4	Family 5	Family 6
Rumbles roars explosions crashes splashes booms	Whistles hisses snorts	Whispers murmers mumbles grumbles gurgles	Screeches creaks rustles buzzes crackles scrapes	Noises made by percussion on metal wood skin	Voices of animal and man shouts,screams, groans,shrieks, howls,wheezes and sobs.
Roarer Burster	Whistler Hisser	Gurgler	Croaker Crackler	Rubber	Hummer Howler

Table 1: Different families of noises as described by Russolo in [12]. The top part of the Table lists the different families, while the bottom part shows the corresponding musical instruments designed by Russolo.

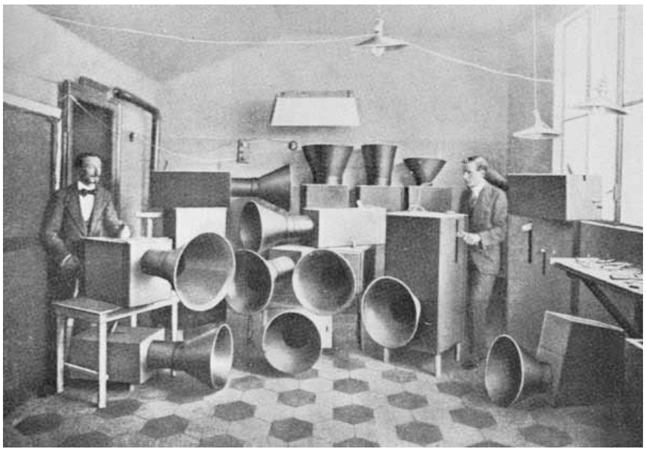


Figure 2: Russolo and his colleague Ugo Piatti playing the original Intonarumori. From [8].

possible to change the tension and length of the vibrating string, and therefore its fundamental frequency.

The 27 varieties of Intonarumori built by Russolo and his colleagues aimed at reproducing different varieties of noises. The names of the instruments were assigned according to the sound they produced. As an example, in the *Gracidatore* (the Croaker), whose excitation mechanism is shown in Figure 3, the shape of the rotating wheel allows to obtain plucked string sonorities. The wheel, rotating at a speed controlled by an external crank, excites a vibrating string attached at two extremities of the wooden soundbox. The player, as in the other instruments, is able to control the tension of the string by using an external lever.

In the Crepitatore (the Cracker), shown in Figure 4, the excitation mechanism is a metal wheel, and two levers are present, as well as two vibrating strings. This allowed the string attached to the drumskin to be different from the one excited by the rotating wheel. The same idea was also adopted in the Stroppicciatore (the Rubber). A second lever was also added to the Burster (Scoppiatore), the Whistler (Sibilatore) and the Gurgler (Gorgogliatore). In his writings, Russolo does not explain the need for such second lever. Moreover, documents and patents did not succeed in explaining the role of the two strings in the resulting sonorities produced by the instruments [8].

In the Ululatore (Howler), described by Russolo as *soft, velvety and delicate* and the most musical among the instruments [12], shown in Figure 5, the excitation mechanism was a smooth wooden wheel. Russolo underlines the fact that this instrument could produce very long notes, since the duration of the notes depended by how long the performer turned the crank.

Russolo and his assistant Ugo Piatti researched all the physical aspects that could be varied to obtain different timbres and sonorities, in order to achieve a satisfactory simulation of the families of noises described above.

As an example, the string was made of either steel or gut, the wheel was made of metal or wood, with its rim notched with small teeth or smoother, and the skins were soaked in a variety of special chemical preparations. Furthermore, the pressure of the wheel against the string, stronger than is necessary with a violin bow, created a louder and noisier sound quality.

Russolo also experimented with more radical Intonarumori, based on electrical rather than mechanical control, such as the one used in the Hummer (Ronzatore), which was more a percussion than a string instrument. It has been suggested that the electrical control might have been due to the need for a speed that was too rapid to have been achieved manually.

3. THE CROAKER

In the attempt to create a modern reconstruction of Russolo's Intonarumori, which could be used both as a musical instrument on its own and as an interface for real-time sound synthesis, we designed the Croaker, shown in Figure 6.

The Croaker can be classified as an instrument-like controller [13], since it emulates the control interface of an existing, although not popular, acoustical instrument.

The first prototype of the Croaker, shown in Figure 6, is an interface built with Lego blocks. The name of the instrument derives from one of the original Russolo's instruments.

As in the original Intonarumori, the Croaker is provided with a one degree of freedom lever moving vertically, and a rotating crank. The position of the lever is detected by a potentiometer, attached as shown in the right side of Figure 6. The rotation of the crank is also sensed by a second potentiometer, attached to the wheel as shown in Figure 6.

The second prototype of the Croaker is shown in Figure 7.

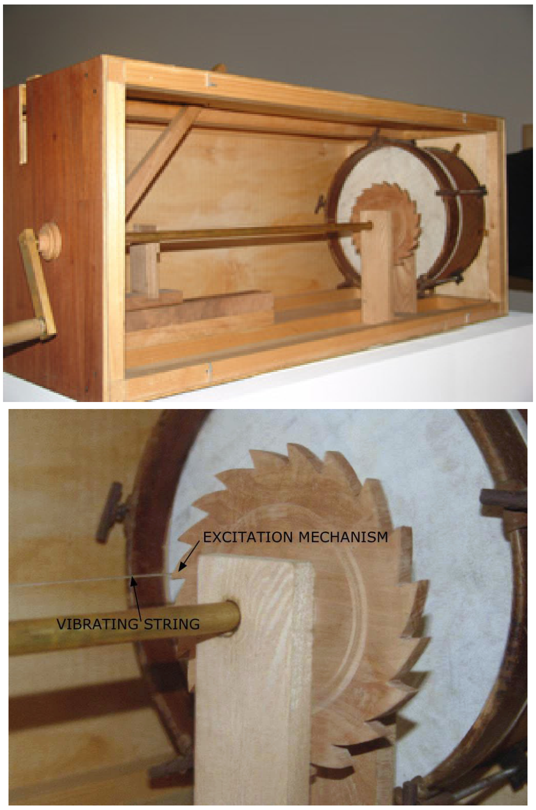


Figure 3: A view of the Gracidatore (top), and its excitation mechanism (center).

Compared to the one shown in Figure 6, the instrument has a more compact shape, and a linear slider is provided. Such slider allows to vary the frequency range of the instrument.

The current prototype of the Croaker is shown in Figure 8. As the original instruments, the Croaker is now made of wood. The more robust design allows the instrument to be used in performances.

In all prototypes, the sensors are attached to a Teleo microprocessor manifactured by Making Things.[1] The microprocessor is connected to a computer through the USB port. In the current prototype, the microprocessor is placed inside the Lego box.

By using the Max/MSP and Jitter software,[2] some adhoc external objects have been developed by MakingThings, which convert the sensors data into numerical input which can be read by Max. Such data are used as controllers to different sound synthesis engines, as described in the following section.

The Croaker is an interface which is easy to learn how to play. It is played by controlling the position of the lever with the left hand, while rotating the crank with the right hand. It is also possible to vary the frequency range of the instrument by using the linear slider.

4. THE SOUND SYNTHESIS ENGINE

The Croaker is a controller which can drive several sound synthesis algorithms. In developing the sound synthesis engine, we followed the approach of decomposing a vibrating system into exciter and resonator [6].

In particular, we simulated the vibrating string positioned inside the instruments as a modal resonator [1, 10]. The parameters of the string are controlled directly in the software engine, as described later. The string is excited by different mechanisms, which allow to create different everyday sonorities. We are interested in simulating scraping and screeching sounds, as well as percussive sounds, rumbling, roars and voices.

It is interesting to notice that by simply varying the excitation mechanism and the resonant frequencies of the resonator it is possible to simulate different kinds of everyday

[1] www.makingthings.com
[2] www.cycling74.com

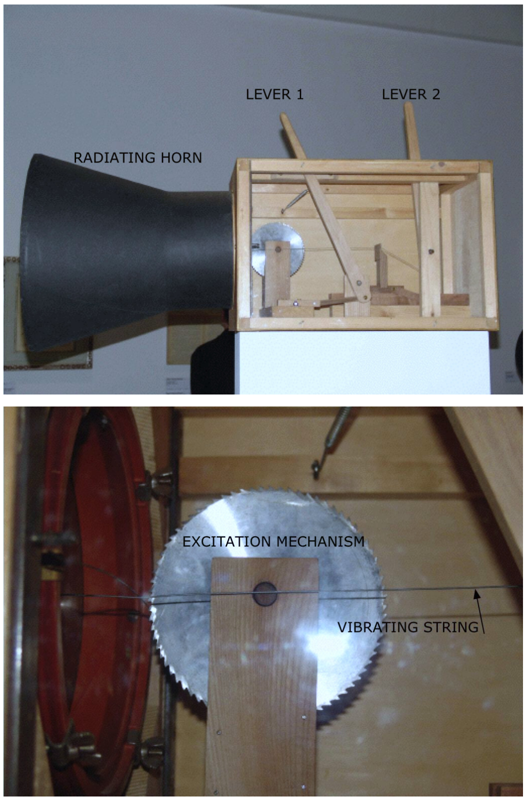


Figure 4: Reproduction of the Crepitatore (top). In this instrument, two levers are present. Bottom: the excitation mechanism of the Crepitatore.

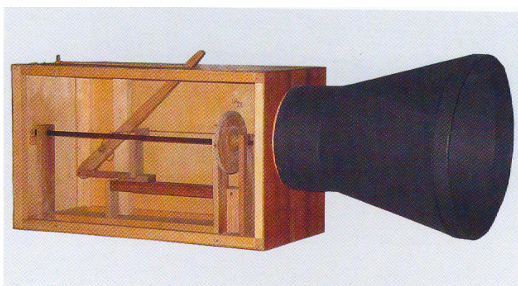


Figure 5: Reproduction of the Ululatore.

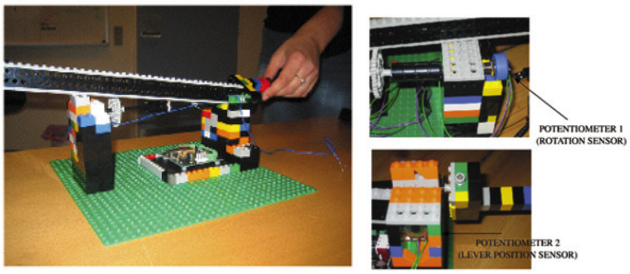


Figure 6: The first prototype of the Croaker (left). Placement of the two potentiometers in the Croaker. The first potentiometer detects the position of the lever, while the second detects the position of the crank (right).

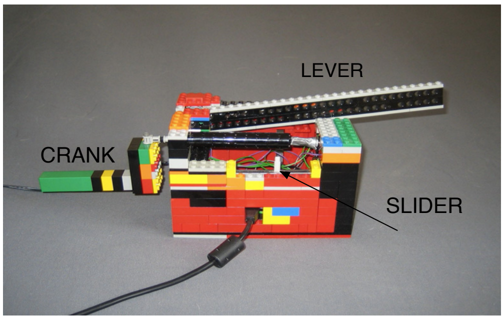


Figure 7: The second prototype of the Croaker.

sounds, from scraping to laughing sounds.

4.1 Scrapes and screeches

Instruments such as the Howler are characterized by a smooth rotating wheel, which continuously interact with the vibrating string. To model the sustained excitation between the rotating wheel and the string, the elasto-plastic friction model proposed in [9], and already adopted for sound synthesis purposes in [2], is used.

In this model, the interaction between the string and the rotating wheel is described by using a differential equation. A detailed description of the use of this model for real-time sound synthesis is proposed in [3].

4.2 Rumbles, roars and percussive sounds

Rumbles, roars and percussive sounds were obtained using the physically informed sonic model (PhiSM) algorithm proposed by Perry Cook [5]. This algorithm has proved to be suitable for the synthesis of everyday percussive sounds. In this situation, the lever controls the fundamental frequency of the particles, while the lever controls the probability of contact among particles.

4.3 Breaking sounds

In his documents, Russolo described the sound produced by the Bursters instruments, claiming that such instruments produced two kinds of sonorities. The first resembles the sound of a motor, while the second reminded the sound of breaking objects. To simulate breaking sonorities, we adopt the algorithm suggested in [7]. In this algorithm, the fundamental frequency of the resonators increases over time, to

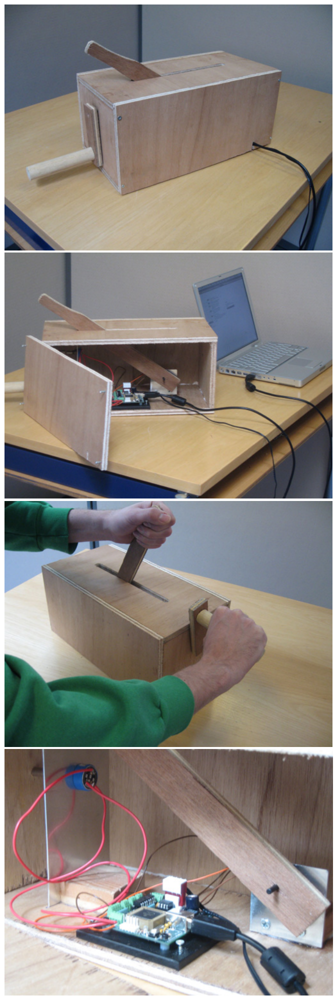


Figure 8: The current prototype of the Croaker. From top to bottom: a view of the instrument, a view of the microcontroller and the sensors inside the instrument, use of the instrument and a close view of how the sensors are connected to the lever and crank.

simulate the size reduction of the broken object. Moreover the breaking sound is simulated by having impact events increasing over time.

4.4 Laughing sounds

By choosing the appropriate modal frequencies of the resonator, it is possible to simulate simple laughing sonorities. In particular, we used the time domain formant wave function synthesis (FOF) technique [11], to generate different vowels by combining particles together, each representing a fundamental period of a signal corresponding to a formant. Notice that PhiSM can be seen as a generalization of FOF, as described in [4].

5. MAPPING

Figure 9 shows the connection between the control parameters of the instrument and the sound synthesis engine. As described before, the player is able to control two param-

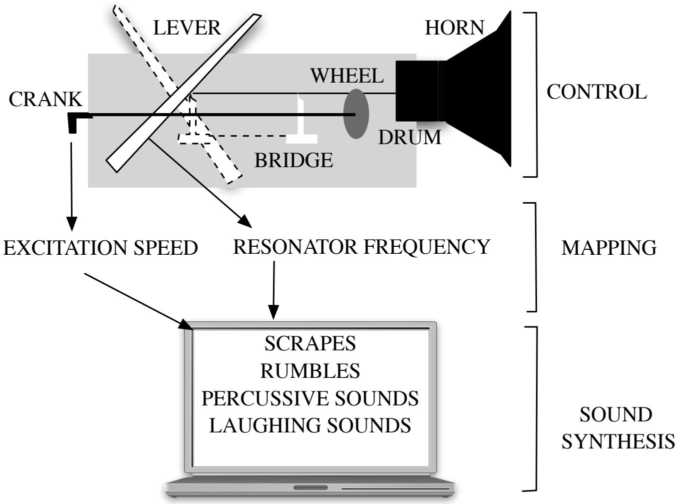


Figure 9: Mapping between the control parameters of the Croaker and the sound synthesis engine.

eters: excitation velocity, which is given by the rotational speed of the rotating crank, and position of the lever. Additionally, a slider allows to change the frequency range of the sound synthesis engine. The mapping strategy chosen reflects the design by Russolo. Infact, the rotating crank controls the speed of the rotating wheel inside the instrument, i.e., the speed of the excitation mechanism.

The lever, on the other end controls the fundamental frequency of the resonator.

This mapping is both intuitive from the player's perspective and faithful to the initial design by Russolo.

6. IMPLEMENTATION

The Intonarumori model has been implemented as an extension to the Max/MSP environment. In the Intonarumori originally designed by Russolo, the control parameters of the instruments are the type of excitation mechanism (plucked or rubbed), which corresponds to the simulation of different instruments of the family, the rotational velocity of the excitation wheel, controlled by the player through the external crank, and the string tension, controlled by the player by moving the lever on top of the instrument. Additionally, it is possible to control the frequency range of the vibrating string by using a continuous linear slider. Figure 10 shows the Max/MSP patch which simulates the Intonarumori described in the previous section. In this patch, it is possible to identify three main components. The top part contains the objects which implement the connection between the Teleo sensors board and Max/MSP. These objects are already available from the Making Things website (www.makingthings.com). The central part contains the mapping strategies to connect the data of the sensors to the sound synthesis engine. The position of the lever is mapped linearly to the fundamental frequency of the string. The fundamental frequency of the string can also be varied by using the linear slider. The rotational velocity of the crank is obtained by calculating the derivative of the position, and is mapped to the excitation velocity of the sound synthesis model. In the case of a transient excitation, the excitation velocity affects the number of bumps per second. In the case of the sustained excitation, the velocity affects the exciter velocity. As is the case in the original instruments, the excitation force cannot be controlled by the player, but it is predefined in the physical model. Similarly, the parameters of the friction model, as well as the different strings' material need to be chosen in the Max/MSP patch, and cannot be chosen using the controller.

In the original Intonarumori, all the instruments had the same control mechanism, and sonorities could be varied by changing the instruments. The same happens in our case: we have a single controller able to drive all the instruments, but in order to change from one instrument family to another it is currently necessary to select a different option in the Max/MSP engine.

7. CONCLUSION

In this paper we introduced the Croaker, a new input interface inspired by Luigi Russolos Intonarumori. Experiments with the instrument show that users find it easy to learn how to play and at the same time entertaining. We found that users quickly adjusted to the mapping strategies and control parameters of the instrument, given the limited amount of control possibilities. This of course can represent both an advantage and a limitation of the instruments. If the instruments had to be used in a performance, it would be helpful to have several of them, as Russolo was doing in his concerts. This could allow novice performers to get involved in new music.

Currently, the sound synthesis engine is imitating the sonorities produced by Russolo's original instruments. Sound synthesis could of course also be used to extend sonic possibilities of the original instruments, especially if they had to be used in a performance.

There are different reasons while rebuilding Russolo's instruments is interesting. First of all, studying these instruments allows to achieve a better understanding of their sound production mechanism. Moreover, preserving these instruments is important from a cultural heritage perspective. Nowadays, although some people might have heard of Russolo's Intonarumori, few are able to describe their design and their sound production mechanism. Since all the original instruments have been destroyed, and few reconstructions are present around the world, it is important to preserve such important contribution to the musical heritage of the 20th century.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the anonymous reviewers for their useful comments and composer Juraj Kojs who will use the Croaker in his music.

9. REFERENCES

[1] J. Adrien. *The Missing Link: Modal Synthesis.* Representation of Musical Signal, G.. De Poli, A. Picialli, and C. Roads, eds, Cambridge, Massachusetts, MIT Press.

[2] F. Avanzini, S. Serafin, and D. Rocchesso. Modeling interactions between rubbed dry surfaces using an elasto-plastic friction model. In *Proc. DAFX*, 2002.

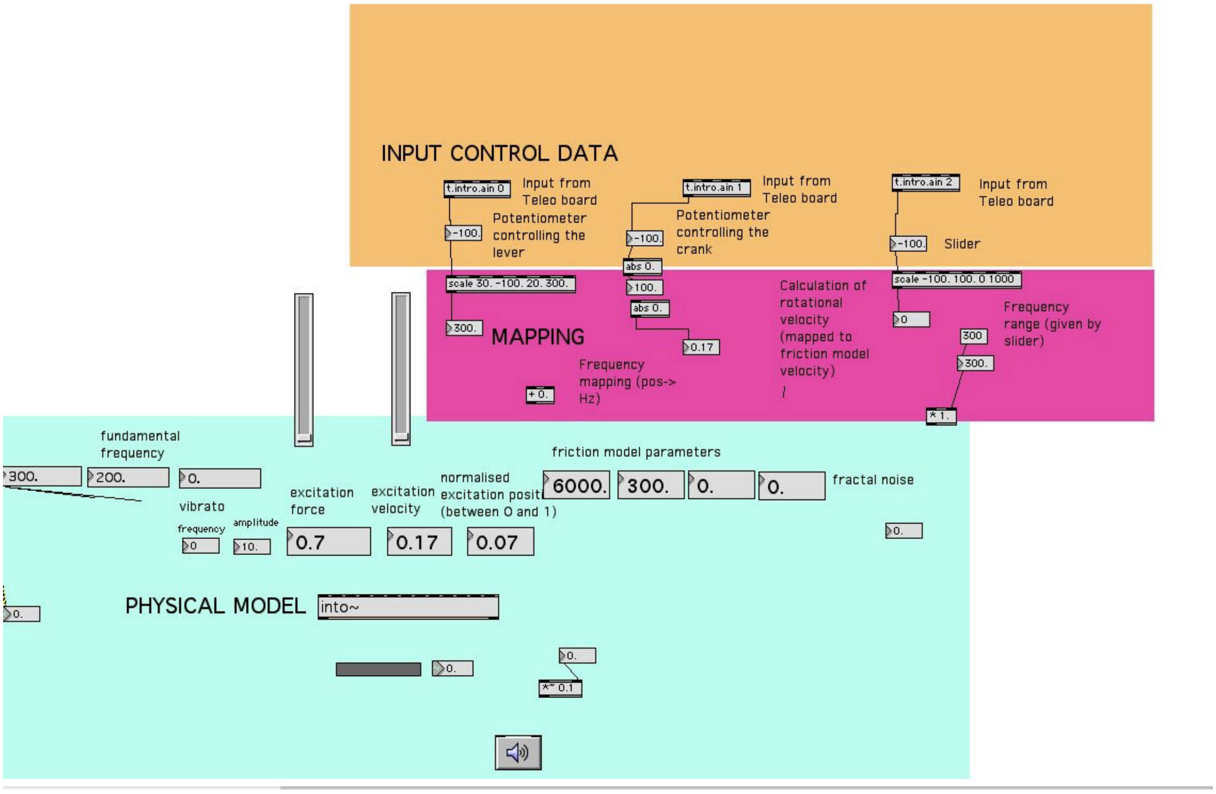


Figure 10: Max/MSP patch which implements the sound synthesis engine. Top part (orange): connection between the Teleo microprocessor and Max/MSP. Center (pink): the mapping from sensors data to the sound synthesis engine; bottom (light blue): the sound synthesis engine.

[3] F. Avanzini, S. Serafin, and D. Rocchesso. Interactive simulation of rigid body interaction with friction-induced sound generation. *IEEE Transactions on Speech and Audio Processing*, 13(5), 2005.

[4] P. Cook. *Real sound synthesis for interactive applications.* AK Peters., 2002.

[5] P. R. Cook. Physically informed sonic modeling (phism): Synthesis of percussive sounds. *Computer Music Journal*, 21(3), 1997.

[6] G. D. P. G. Borin and A. Sarti. Sound synthesis by dynamic systems interaction. *Readings in Computer-Generated Music. IEEE Comp. Soc. Press*, 1992.

[7] W. Gaver. Synthesizing auditory icons. In *INTERCHI*, 1993.

[8] G. Maffina. *Luigi Russolo e l'Arte dei rumori, con tutti gli scritti musicali.* Martano editore, 1978.

[9] B. A. P. Dupont, V. Hayward and F. Altpeter. Single state elasto-plastic friction models. *IEEE Transactions on Automatic Control*, 47(5), 2002.

[10] D. Rocchesso and e. F. Fontana. *The Sounding Object.* Mondo Estremo, Firenze, Italy, 2003.

[11] X. Rodet. Time-domain formant wave-function synthesis. *Computer Music Journal*, 8(3), 1984.

[12] L. Russolo. *L'arte dei rumori (The art of noises).* Edizioni Futuriste di Poesia, 1916.

[13] M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, 26(3), 2003.

Interactive Sonification of Neural Activity

Gil Weinberg
Georgia Institute of Technology
840 McMillan St. Atlanta, GA 30332
Tel: 404-894-8939
Email: gil.weinberg@music.gatech.edu

Travis Thatcher
Georgia Institute of Technology
840 McMillan St. Atlanta, GA 30332
Tel: 404-894-8939
Email: travis.thatcher@gatech.edu

ABSTRACT
We discuss our ongoing research in sonification of neural activity as demonstrated in the “Fish and Chips” and “BrainWaves” projects. We argue that sonification can serve as an effective technique for the representation of complex spatial information such as neural activity due to the auditory system’s ability to perceive stimuli at a wide spatial cover and its inclination to perceive spatial patterns in sonic input. The paper discusses aesthetic and functional aspects of sonification in this context and describes the evolution of our technique, artistic approach, and interaction design – from the low-resolution graphical user interface in “Fish and Chips” to the high-resolution tangible interaction with newly developed controllers in “BrainWaves”. We conclude with an evaluative discussion and a number of suggestions for future work.

Keywords
Sonification, interactive auditory display, neural patterns.

1. BACKGROUND AND MOTIVATIONS
With new developments in biological research, scholars are gaining more accurate information about complex systems such as the brain. These developments create a need for effective mechanisms for representation and new approaches for user interaction with such complex datasets. Several visualization techniques have been proven to be useful in addressing similar challenges [1, 5]. For other applications in fields as diverse as oceanographic buoy readings [13] to stock market trends [7], sonification of data has been found effective, utilizing the auditory system’s unique strengths such as wide spatial cover and inclination for pattern recognition. Some sonification systems focus on scientific and functional depiction of data [8, 16, 17] while others focus on aesthetics and musical representation [9, 10]. Several recent efforts also addressed user interaction design with sonification, guiding users through dataset queries [3, 6]. Most interactive sonification systems, however, do not support dynamic hands-on interaction that can provide both functional and aesthetically pleasing sonic experiences. We believe that tactile dynamic interaction with auditory displays, which immerse players artistically and scientifically in the experience can provide deeper and more intimate sonic familiarity with complex real-world data. Our main research goal, therefore, was to provide an aesthetically satisfying and educationally useful representation of complex datasets of neural activity through hands-on interaction. Our preliminary effort in this area was in the framework of the “Fish and Chips” project (in collaboration with the Symbiotic Research Group, Perth Australia [14]), where we sonified low-resolution audio signal from an in-vitro culture of fish neural cells. For this project we developed a simple graphical interface for users to choose between datasets. Based on observations from this study we developed a new system, titled “BrainWaves” (in collaboration with the Laboratory for Neuro-Engineering at Georgia Tech [11]), in which we analyzed higher-resolution signals, allowing a group of players to interact and manipulate the data in a performance setting using new tactile controllers that we have developed. Our main goal in “BrainWaves” was to enable players to explore and perceive the neural information in an immersive manner, providing a direct and intimate connection to the information through tactile interactions. We believe that such an approach for interaction with auditory displays can provide artistic, scientific, as well as educational merits.

2. “FISH AND CHIPS”
“Fish and Chips” was a bio-cybernetic research project, originating from the SymbiocA research group in the University of Western Australia, which explores aspects of creativity and artistry in the age of biological technologies [2]. As part of the project, information from fish neurons grown over silicon wafers was sent to control a robotic arm programmed to draw pictures based on the real-time neural activity. Our involvement in the project entailed the development of a musical module that mapped the neural information to generate and manipulate digital music. In addition, we developed a graphical user interface that allowed users to browse and interact with different neural patterns datasets. By creating such a temporal "artist" that drew and controlled music in real-time, the project explored questions concerning art and creativity, such as how humans will interact with cybernetic entities with emergent and unpredictable behaviors and how society would treat notions of artistry and creativity produced by such “semi-living” entities. As part of the project, which was presented at the Ars Electronica Festival 2001, we used the audio representation of real-time electrical activity of fish neurons to generate and control electronic musical output based on the energy in different frequency bands in the signal. We also created a feedback loop, feeding the music back to the neural culture as external stimulus, allowing the algorithm to process the fed back information and produce second derivative output. The installation was featured as a laboratory set-up and included real-time generation of art and music as well as a set of pre-composed and pre-drawn examples.

Figure 1. A microscopic photograph of the neuron culture used for sonification in "Fish and Chips"

In an effort to depict the activity in the culture in a clear manner, we created a simple correlation between the energy in 25 frequency bands in the audio signal and the velocity of real-time generated MIDI notes. For each spike in each frequency band in the signal, a MIDI note was triggered, representing the center pitch at that frequency band at the appropriate loudness. To provide hands-on familiarization with the data we developed a graphical user interface in Max/MSP which depicted the 25 frequency bands, allowing users to choose between the output of live and recorded data, raw signals and filtered signals, pink noise for comparison, and the output from a live feedback loop (see Figure 2.)

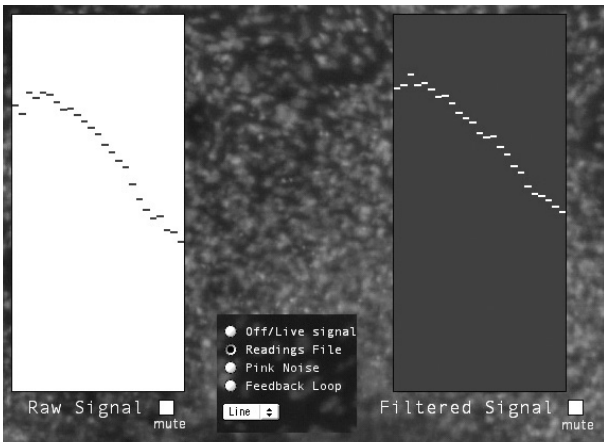


Figure 2. The graphical user interface in "Fish-and-Chips"

In the installation at "Ars Eelectronica" in 2001 hundreds of visitors interacted with "Fish and Chips" musical module. Most participants found the installation interesting and effective. A number of points for improvement, however, also became apparent. First, it was clear that the crude one-channel audio signal does not provide high-resolution spatial representation of the activity in the culture. The simple mapping strategy we used provided only a symbolic gesture for the possible richer musical outcome that can be generated based on detailed spatial neural data. Secondly, the graphical user interface that we designed allowed for only one person to interact with the data, and did not provide a tactile and immersive hands-on familiarization with the information at hand.

3. "BRAINWAVES"

In "BrainWaves" we attempted to address some of the deficiencies that were observed in "Fish and Chips". First, we were interested in representing the spatial propagation of neural activity in the brain based on accurate high-resolution data. In mature neural cultures, spatially localized bursts become common as groups of neurons generate spikes in clusters, stimulating other clusters to trigger in response. We decided to use spatial auditory patterns to represent the propagation of the neural patterns in space, aiming to create both aesthetic and functional merits. We believe that sonification can be more effective than visualization in such a spatial context due to the human auditory system's ability to perceive synchronous spatial stimuli from every point within the space, while visual perception is limited to the physical range of sight. In addition, we decided to provide players with tangible interaction with the data using newly designed percussive controllers. By allowing players to trigger neural spike propagations based on their natural occurrence, we hoped to encourage an active learning process of the interconnections and patterns in the culture. We, therefore, developed a system that allowed a group of players to collaborate in composing a musical piece using our new controllers. As part of the performance, neural spikes and propagation clusters were used to influence and prompt players to respond with actions of their own, facilitating the creation of a unique interactive experience. As players interacted by sending sound waves to each other in a manner that simulated spike propagation in the culture, they became a part of the system, reacting and interacting as the neurons do. The encoded neural information functioned, in a sense, as an equal-right participant in this process. In order to reinforce the functional goals of the system, we also complemented the auditory display with a video display (see Figure 3), which helped represent the information to players and viewers in a multimodal manner.

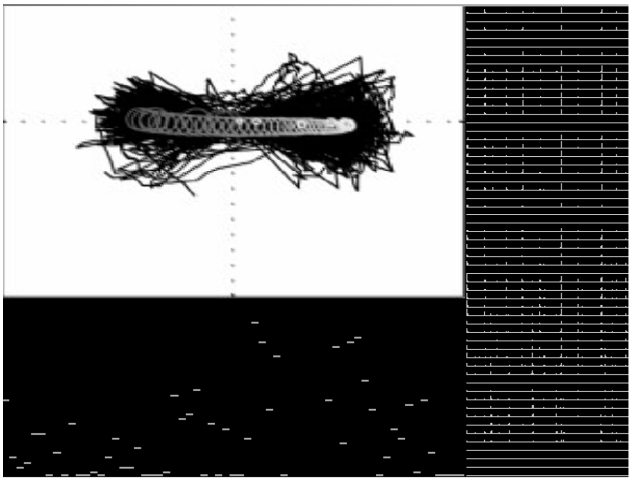


Figure 3. Real-time visualization of the data as projected to audience in "BrainWaves" performance. Top left - currently active spike propagation pattern. Bottom left - real-time data values from 60 electrodes. Right - sensor history

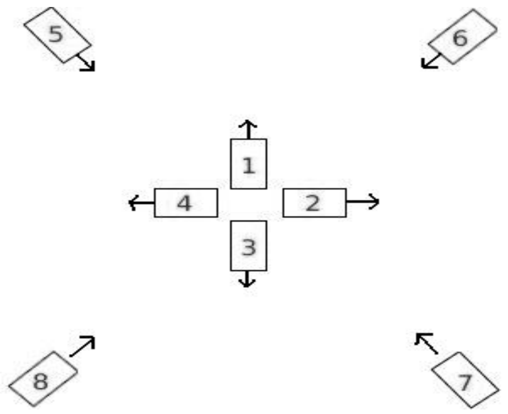


Figure 4. Speaker placement and sound projection in "BrainWaves"

The neural data used in "BrainWaves" was recorded and processed by a research group at the Laboratory for Neuro-Engineering at the Georgia Institute of Technology, directed by Steve Potter [11]. The data was captured from neuron cultures of a mouse cortex, grown over multi-electrode sensor arrays arranged in an 8 X 8 grid [15]. Two different sets of data were used for the project. The first was the raw recorded information from sixty electrodes, measuring the electrical activity in different areas of the culture. The second dataset was based on pattern recognition methods used to study the spatial propagation of spikes in the culture. Our system utilized nine of the most commonly occurring burst propagation patterns in the culture, simulating their trajectories as sound propagation in space. In order to provide effective spatial resolution for the representation of a sixty-electrode grid, we decided to use an eight speaker sound system. More speakers could have presented difficulties in interaction conveyance and sonic cancellation, while fewer speakers might not have provided high enough spatial resolution. Our goal was to represent the propagation in the culture as accurately as possible in a manner that would fit well with the performance space. We, therefore, divided the sensor arrays into eight different zones as depicted in Figure 4. Four outer zones extended from the corners of the grid towards the center, and four inner square zones were projecting sound from the center outwards. In mapping the data to the speakers, we used each sensor value per time sample to calculate an average of simultaneous values for each zone, testing it against a predetermined threshold. If the value exceeded the threshold, a spike was determined for the appropriate zone. We determined which spatial propagation pattern had to be triggered by using statistical data on each spike pattern probability given its spatial origination within the culture. The audio was then sent through the appropriate speakers, approximating the path of the propagation. When idle, the system played a soft drone sound-scape using long tones and pads of low frequency noise. When a spike was triggered by the encoded data, high frequency sounds were used to represent the propagation patterns. We also allowed users to trigger their own spikes, using a set of newly developed controllers (see Figure 5.). For these user-triggered spikes, we chose a set of loud and distinctive high frequency sounds with a noisier content. This timbre differentiation helped portray the interaction to users and to the audience by separating between the recorded data, the analyzed spike propagation, and the user-generated spikes.

Each of the eight 3 feet percussive controllers was installed next to a speaker around the performance space. Each controller contained a piezoelectric sensor to detect hits and an electric circuit to drive two sets of LEDs, offering spatial visual representation of sound propagation in space. An elastic chord connected the hitting surface to the controller's frame, providing haptic and expressive hit response. The system was designed to provide tactile familiarization with the network topology and the high-level structures in the culture. In order to engage players and provide a long lasting musical experience, we designed the interaction as a game. Players could trigger a propagation pattern by hitting their respective controllers. A spatial wave of sound then propagated from the nearest speaker to other speakers in the space, simulating the neural propagation in the culture. When the pattern ended, the player positioned at that receiving station was prompted to respond with his or her own spike trigger, and so on. By allowing players to send waves of sound to a stochastically chosen destination (the propagation cluster was chosen based on its statistic occurrence in the culture) we added an element of surprise to the system which encouraged players to follow the propagation cluster more closely and to try and surprise their co-players. Free-play interaction schemes were also experimented with, which often led to shorter, less engaged play sessions. In January 22, 2005 the "BrainWaves" system was presented in a performance at the Eyedrum Gallery in Atlanta, GA. The performance began with a brief explanation of the project, followed by a demonstration of the system in idle mode. We then played the recorded neural data accompanied by a visual representation of sensor history, real-time sensor activity, and a visual representation of spike propagation each time a spike occurred (see Figure 1). The system ran the recorded data for several minutes, giving the audience the opportunity to become familiarized with the information and its representation. After several minutes, a group of six performers started triggering spikes and propagating sound waves according to the game rules as described above. The performance proceeded in this manner, eventually involving all performers up to a point where the activity lessened and the system returned to play in an autonomous manner, fading out slowly. When the performance ended, audience members were invited to interact with the system in free play mode (see Figure 5).

Figure 5. A player interacts with the "BrainWaves" controller, triggering sound patterns in space based on probabilistic occurrence of neural spike in the brain

4. DISCUSSION AND FUTURE WORK

The "BrainWaves" performance was well received, leading to positive responses from audience and participants who found the tactile interaction and detailed spatial information useful and aesthetically interesting. This constituted a notable improvement in comparison to user reactions in "Fish-and-Chips". The observations and discussions with participants and audiences also led to a number of suggestions for further improvement and future work. While the environment was intriguing and immersive to most users, the relationship between the data and the sounds, particularly in the interactive sections, were not apparent to everyone. While the projected visualization did help the audience to better understand the activity in the culture, for some players the graphics became the main focus of attention. For the next version of the project we plan to experiment with less distracting visualization schemes that would complement and augment the auditory artistic experience, rather than dominate it. Another manner in which we plan to improve the auditory display is to conduct further experimentation with speaker placement and mappings. The sounds we chose for the performance worked well and created the aesthetics that we aimed for, but the system can benefit from experimentation with other timbres sets that are further separated from each other perceptually. This would hopefully improve group interaction, since at times, players found it difficult to follow the sound propagation. The game interactions were well received and encouraged participants to follow the sounds that they created in space, adding an element of tension and surprise. Some players, however, were prompted to interact more than others due to the fact that many patterns in the data ended in the same few stations. To address this problem we plan to explore other spatial datasets that may show a larger variety of propagation patterns. We will also investigate and explore other interaction schemes using non-spatial patterns in the data. This may involve implementing new pattern recognition methods, using tools such as neural networks to discover new and potentially meaningful patterns in the culture. Finally, we aim to improve the musical mappings by focusing on elements such as rhythm and harmony, utilizing perceptual concepts such as rhythmic stability and harmonic tension [4, 12], which may lead to richer and more interesting musical results.

5. ACKNOWLEDGMENTS

We would like to thank Steve Potter, Doug Bakkum, Clint Cope, Mark Godfrey, Jason Orlosky, Tony Gordon, Scott Driscoll, and the Eyedrum Gallery in Atlanta for their help and support in realizing this project.

6. REFERENCES

[1] Ben-Tal, O. and Berger, J. Creative aspects of sonification, *Leonardo*, 20 (3), MIT Press 2004, 229-232.

[2] Catts, O., Zurr, I. Growing Semi-Living Sculptures *Leonardo* 35(4), MIT Press 2002

[3] Derthick, M., Kolejejchick, J., Roth, S. F. An interactive visualization environment for data exploration. *Proc. Of Knowledge Discovery in Databases*. AAAI Press, August 1997, 2-9.

[4] Dibben, N. The Perception of Structural Stability in Atonal Music: The Influence of Salience, Stability, Horizontal Motion, Pitch Commonality, and Dissonance. *Music Perception*, 16(3), 1999, 265-294.

[5] Fry, B. (2002). Genome Valence. access at: http://acg.media.mit.edu/people/fry/valence/.

[6] Heer, J., Card, S. K., Landay, J. A. Prefuse: A toolkit for interactive information visualization. *Proc. of the International Conference for Computer-Human Interaction (CHI 2005)*, ACM Press (2005), 421-430.

[7] Hunt, A. and Hermann, T. The importance of interaction in sonification. *Proc. of the International Conference on Auditory Display (ICAD 2004*) 61-64.

[8] Lodha, S. K., Beahan, J., Heppe, T., Joseph, A. J., and Zane-Ulman, B. Muse: A musical data sonification toolkit. *Proc. of the International Conference on Auditory Display, ICAD 1997*, 41-54.

[9] Paine, G. (2004). Reeds – a responsive sound installation. *Proc. of the International Conference on Auditory Display (ICAD 2004*)

[10] Paine, G., Ramakrishnan, C., Tydeman, J. (2005). Meterosonics. Access at: http://www.meterosonics.com

[11] Potter's Neuroengineering Laboratory at Georgia Tech - http://www.neuro.gatech.edu/groups/potter/index.html.

[12] Schmuckler, M. A. Testing Models of Melodic Contour Similarity. *Music Perception*, 16(3), 1999, 295-326.

[13] Sturm, B. L. Surf Music: Sonification of Ocean Buoy SpectralData. Proc. of the 2002 *International Conference on Auditory Display, (ICAD 2002*) Kyoto, Japan, 2002.

[14] Symbiotica Research Group - http://www.symbiotica.uwa.edu.au/

[15] Wagenaar, D. A., Madhavan, R., Pine, J., and Potter, S. M. Controlling bursting in cortical cultures with closed-loop multi-electrode stimulation. *Journal of Neuroscience*, 25(3), January 19, 2005, 680-688.

[16] Walker, B. and Cothran, J. Sonification sandbox: a graphical toolkit for auditory graphs. *Proc. of the International Conference on Auditory Display (ICAD 2003).*

[17] Yeo, W. S., Berger, J., and Zune Lee. SonART: A framework for data sonification, visualization and networked multimedia applications. *Proc. of the International Computer Music Conference (ICMC 2004).*

Non haptic control of music by video analysis of hand movements : 14 years of experience with the «Caméra Musicale»

Jacques Rémus
Ipotam Mécamusique
19 rue des Frigos 75014 Paris France
+33 1 45 82 874 40
jr@directmail.org

ABSTRACT
The "Caméra Musicale" is an interface which allows a musical practice based on the movement of hands and fingers under a camera. Born from technologies and methods developed in the early eighties to create music from choreographic movements, the "Camera Musicale" has evolved during the nineties to become a user friendly device since 2003. This article describes its fundamental principles of operation, which have remained relatively unchanged. It then focuses on the importance of the choices necessary to be made while analyzing the video image.

Keywords
Musical Camera, camera musicale, non haptic instrument interface, musical hand's mappings, Jacques Rémus'sound sculptures and mechanical musical machines.

1. INTRODUCTION
The "Caméra Musicale" is an instrumental interface which allows musical practice based on the position and movements of hands in space, in the scope of a video camera. After the first prototype was elaborated, its development was insured by constant experimentations. The first version (1992) had an audience interact by moving their hands in mid-air, controlling large and mechanical musical devices. The appeal of this version lied in the contrast between the immaterial nature of the movements and the materiality of the music produced by the machines.

The system was immediately embraced by audiences of all kinds : Festival audiences, school kids, the handicapped, and of course, musicians. As years went by, thousands of people have played "Caméra Musicale" in various contexts. Consequentially, its development was in close relation with the audience and the "Caméra Musicale" evolved on grounds of experimentation - a very pragmatic, rather than scientific approach.

These experiments also led to the development of another technology: an interface controlling audio or MIDI systems.

2. ORIGINS AND DEVELOPMENTS
The "Caméra Musicale" was born from research and works by Sylvain Aubin, with the "Manorine.

A first prototype was developed in 1982 " (patent n° 82695, december 1982). The point of this work was to combine a logical interface between a video camera and analogical synthesizers. The main goal of such a machine was to create music for choreographic performances (Stéphanie Aubin, performances, Paris, Rennes 1986).

The principle was based on the analysis of a clear surface (the body, or a part of the body) being detached in hyper-contrast on a black screen. This analysis brought up many characteristics of the moving spot:

X and Y, global parameters giving the position of the object

dX and dY, local parameters describing the size of the object

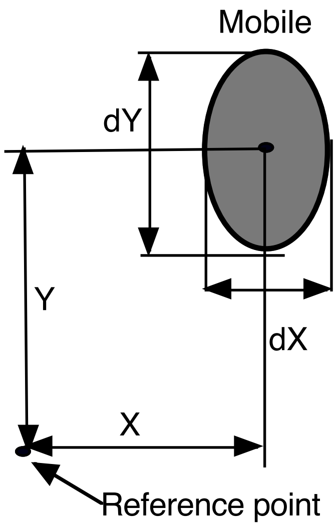


Figure 1. Global and local parameters.

After these first experiments, focusing on dance movements generating music, it was clear that the prototype was a relevant tool for artistic creation. However, its potential was restrained by the fact that it was hard to operate by non-technicians.

In 1992 Jacques Rémus, Sylvain Aubin and Gwek Bure Soh, artist, associated to undertake the project of the "Caméra Musicale".

A Manorine with a MIDI interface coupled with developments on the Max [1] software was then created.

David Rokeby's VNS ("Very Nervous System") [2] initially developed for dance as well, was connected in parallel to the Manorine-MIDI, hence completing the first version of the "Caméra Musicale".

The analysis of the Manorine was the core of the system, while the VNS was used to enhance the movement/sound reactions.

3. GENERAL PRINCIPLES

The "Caméra Musicale" is assembled in part from mainstream elements, available in the market. Those elements are combined to relatively easy developments, considering one is familiar with live electronics software (Max-MSP)

Here are its main components:

- A black and white camera. A standard IR photo filter, positioned on the lens, allows exclusive infrared detection. The camera is aimed towards the floor and stands 1 to 3 meters above the player's hands.

- An infrared projector - the type used for night surveillance (invisible in darkness). Next to the camera, it also aims towards the ground. This system allows the "Caméra Musicale" to remain unaffected by the various lightings, often problematic in live performances.

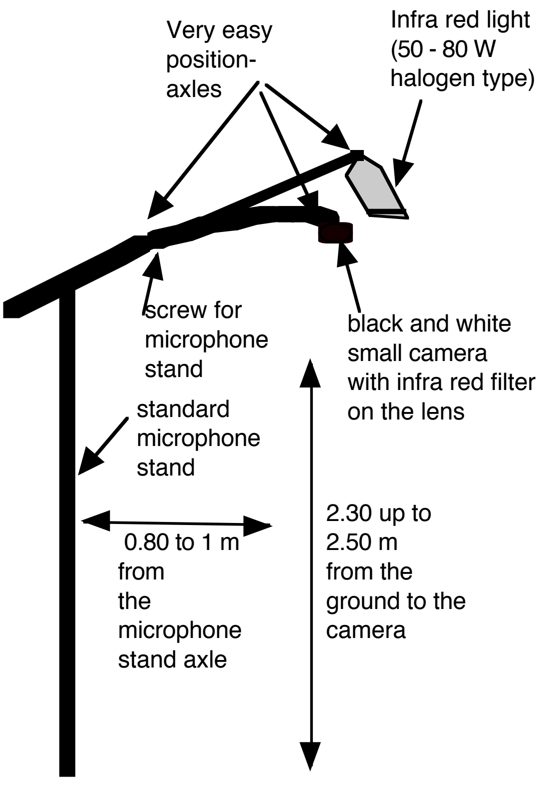


Figure 2. Structure of the "Camera Musicale".

- An interface or a standard video computer card (e.g. a TV card for computer can very well do the job)

- A software development on Max-MSP-Jitter [1], allowing to control and chose, live, various parameters of video analysis and musical patches. Cyclops [1] or VNS [2] plug-ins or Jitter [1] developments are used to filter simple information from the very complex original image. The filtered data then enters in Max, in the form of a continuous flow (25 or 30 images per second). It's basically a software "Manorine" with important improvements from the plug-ins.

- Musical instruments: synthesizers (expanders), musical machines, sound sculptures or computer generated sound (Max-MSP)

- One or many screens, showing the hands, allowing visual interaction for the player and audience.

4. DESCRIPTION OF THE VIDEO/MIDI GENERATED INFORMATION

The information resulting from the video analysis remain the center of the development. As described above (§2), there are four:

- X and Y measure the position of the mobile

- dX measures the widest part of the mobile's image.

- dY measures its highest part.

The dynamic of the mobile's movement (hands or fingers) is essentially rendered by the speed in which X', Y', dX' and dY' evolve. Their calculation in Max creates four new data flows. The calculation of the second derivation of the first four parameters (accelerations X", Y", dX", dY") creates once again four new data flows.

The hands and fingers can be individualized and independent (by definitions of areas or mobile objects' individualization).

On this basis, all the data that can be rendered by pixel variations in defined areas (principle of the VNS) and the indications of the grids from Cyclops complete and enrich the possible video analysis parameters

5. NUMBER OF INDEPENDENT PARAMETERS A MUSICIAN CAN CONTROL

The essential choice while operating the "Caméra Musicale" is the number of independent parameters the musician can control with the camera.

In theory, the "Caméra Musicale" allows to play with a number of parameters, dependent or independent, in various predefined areas in space (drawn on the screen). (cf. §4) However, if the complexity can produce gratifying results with lots of feedback and automatisms, they are seldom relevant.

The number of independent parameters a classical instrumentalist can control is relatively low: pitch, intensity, articulation, timbre and rhythm (succession of events in time). It hardly exceeds four or five.

The same goes for the "Caméra Musicale", with three or four independent parameters. The time parameter, rhythm, is considered independently from the others.

The "Caméra Musicale" is developed to read a limited number (from one to four) of data flows from the incoming information.

Only very simple patches with two to four independent parameters could allow a good control of the outcome. This applies to the beginner as well as to the accomplished musician.

On the other hand, the chosen parameters can be very different from one try to another : X, X', X", Y, Y,' Y", dX, dX,' dX", dY, dY', dY", surfaces, number of pixels and their variations, etc.

Example 1 (basic) : X specifies the pitch of the note or sound, Y specifies the intensity and dY" a change in the attack or timbre.

Example 2: dots or small areas defined in X, Y are drawn on the screen and the parameters in which they accelerate to reach one another are used (variations of pixels combined with dX" and dY" ensembles). For example: the faster the movement is, the longer (or shorter, or harmonically transformed...) the note at the point of release (X, Y) will be. The hands move in space at different speeds and "touch" predetermined points (X, Y) which trigger the sounds, modulated by the speed of the movements.

Example 3: a 12 note loop is automatically played: Its speed is increased in Y, its velocity in X, the length of its notes in dX and their attack in dX'. The mapping then takes into account the width of the hands, or the distance between both hands (dX). However, if we add a change of timbre for dY', the control of yet another parameter would become too difficult.

Once the choices are made, the possibilities of the mapping remain very wide, even if in practice (see § 7) the movements chosen by the players tend to imitate those of a pianist, harpist or drummer !

6. NON "HAPTIC" BODY LANGUAGE.

These gestures are performed in mid-air and the player has counterbalance every movement he makes with its opposite, in order to keep balance. It is hard to master at the beginning, like all non return (non haptic) interfaces, but this doesn't take away the fact that the mapping is precise and pleasant. Moreover, the body being entirely engaged in these gestures, the musician's body language can become an object of interest for the audience.

Practice has proved that the changes between playing modes should be done by a different system that the gestures and the camera. Simple midi-pedals now allow such changes, with no risk of errors due to bad control of the gestures.

7. EXPERIENCE WITH PUBLIC AND MUSICIANS

The "Caméra Musicale" in use since 1992 by large audiences with midi robotized acoustic machines and since 2003 by musicians on electro-acoustic sounds

The "Caméra Musicale" was first used by the "Concertomatique N°2" ensemble [3] [9] made up of several pipe organs, a string quartet, percussion and machines with ringed pipes [3] [5]. Then, it was coupled with the "Carillons des Zic-phones" [3] [6]. the "Carillon Concertomatique N°3" (Festival Résonnances, Ircam, 2002) [3] [7]. and recently, with the "Pic-Verts" (Green Woodpeckers) [3], the "Ensemble des machines à laver musicales" (The Musical Washing Machine Ensemble) [3] [7] [8] and the "Orgabulles" (Bubleorgan) [3].

In concerts, expositions, installations, animations and festivals, thousands of people played with such installations and the mappings were progressively developed and improved. All these experiments show the strength of cultural perceptions related to musical gestures of the instrumentalists or conductor. Surprisingly enough, such gestures were particularly present with musicians.

The "Caméra Musicale" has always proposed innovative mappings (examples 2 and 3 from §5 only being a sample), but users will usually seek three primary types of gestures:

1) The movement related to percussion - with the tip of the finger or pretending to hold a drumstick or mallet (triggering related to a neutral zone at the edge, below the screen, which takes into account the entry of the finger, hand or arm in the image (area drawn in X Y)

2) The movement referring to the organization of pitches on a keyboard (low pitches at the left, high pitches at the right: X parameter).

3) Reference to the intensity, or strength they put in their movements (Y', Y" or more oftenly, dY' and dY" or the variations of pixels in the VNS).

If some mappings do not respect one of those three primary approaches, they are considered to be difficult, beyond understanding or unnatural.

The "Caméra Musicale" is not adapted to the type of movements an instrumentalist would do, in order to imitate the basic technique of his instrument: moving the fingers, trying to find the notes on an invisible instrument.

Research has to take account of such requests, but usually focuses on exploring other possibilities.

It is actually with pre-written modules in the patches that the "Caméra Musicale" takes form of an original instrument: automatic arpeggios, prerecorded loops, prewritten sequence modulations (as the Max Mathews sticks) erase the note interpretation difficulty, and opens to modulations related to a conductor's work.

In addition, the relationship with the Theremin is acquired by pitch-bends on the midi system and more easily again with sounds generated by calculation (Max-MSP).

Finally, the last developments ("Signa" duo concerts with theremin player Rolf Sudman (Berlin)[3]) focused on the live treatment and samples (scratching) of the sung voice, spoken word and noise. The deformation of sound was, in all cases, controlled by movements based on the three primary criteria.. Table 1 indicates the use of the parameters in the plays.

Mappings type		X	Y	X'	Y'	X"	Y"	dX	dY	dX'	dY'	dX"	dY"
INSTR.	impuls	5	5										
INSTR	pitch	5	2						3				
INSTR	velo		4		5				4		4	2	2
INSTR	modul		2					3	2	1	4	2	3
ORCH.	impuls												
ORCH.	pitch	5	1										
ORCH.	velo		4										
ORCH.	modu							3	4				
THER.	impuls												
THER.	pitch	3	3										
THER.	velo	3	3										
THER.	modul							1	2			2	2
D.AUD	impuls	5	5										
D.AUD	pitch	4	1		3			2	1				
D.AUD	velo		4		2				4			2	3
D.AUD	modu	2								1	1		

Table 1. Frequency of the parameters in the mappings (classified from 1 to 5)

Legend :
NSTR. = INSTRUMENTAL, ORCH. = ORCHESTAL
THER. = THEREMIN LIKE, D.AUD = DIRECT AUDIO
impuls = impulsion, velo =velocity, modul=modulations.

This table is not a statistical study, but merely a representation of the users' tendencies. The numbers (0 to 5) represent about fifty mappings: 1 = seldom use to 5= always or almost. We notice that instrumental mappings and mappings using the audio are related to the instrumentalist's basic movements whereas the theremin or conductor mappings allow more fantasy and imagination in the choice of movements.

8. CONCLUSION

The "Caméra Musicale" is an interface for musical performance and also a tool for musical creation. Other research has been developed in the same direction. In addition to the works of David Rokeby, already mentionned [2], the works of Tom Demeyer with the Big Eye software [10] have influenced this type of work - particularly with the residence at the Steim foundation.

The "Caméra Musicale" opens new paths for future musical practices. Its originality and reliability has been tested and approved by a wide audience. Still far away from its full potential, it should however always consider the natural movements of instrumentalists.

9. REFERENCES

[1] Cycling 74 home page : http://cycling74.com

[2] David Rokeby home page : http://homepage.mac.com/davidrokeby/

[3] Jacques Rémus and Ipotam Mécamusique home pages : http://jacquesremus.com http://mecamusique.com

[4] Yamawaki, K., Jacques Rémus: the musical garden with informatics, robotics and musical instruments In *Artec'97, The 5th International Biennale in Nagoya 1997*, official catalogue, (June 1997), The council for the international Biennale in Nagoya, Naganae printing Co, Japan, p.42-49.

[5] Rémus, J. The music of ringed pipes In *Proceedings of the International Symposium on Musical Acoustics* (July 2-6, 1995), SFA Jouve, Paris, France, p.69-73.

[6] Rhobiny, S., Because of the wonderful things he does, In *Kilometer zero N°3*, The Kilometer zero project, 2002, ISSN: 1629-3274, p.50-56

[7] L'oeil de la nuit 2003 Parcours Paris Rive Gauche, official catalogue of Nuit blanche 2003, Paris Musées, Les Musées de la ville de Paris, Paris, ISBN 20042-87900-831-X, p.136-137.

[8] Installation sonore, Lavage de tympans In Nova Magazine, Connexions pirates, N°101, Paris, Mai 2003, p.25.

[9] Official Catalogue of Musique en Scène Festival, Grame, Lyon, France, april 1995, p.8-9.

[10] Steim foundation, (Amsterdam) homepage : http://steim.org

[11] Mécamusique, In *La marionnette et les autres arts, Musiques en mouvement*, Puc n°6 1993 Editions Institut International de la Marionnette, France, ISSN 0993-0701, p.56-57.

[12] Jacques Rémus,Double Quatuor à cordes, De la Motte-Haber, H., *In Klangkunst,Tönende Objekte und klingende,* Räume, Laaber-Verlag, 1999,ISBN 3-89007-432-4, p.160-161.

[13] Jacques Rémus, Musica ex machina, Nicollet, G., Brunot V., In *Les Chercheurs de sons, Instruments inventés, machines musicales, sculptures et installations*, Ed Alternatives, Athènes, 2004, ISBN: 2-86277434-8, p.120-125.

[14] Arts Ménagers, arts sonores , Le Graverend, F., In *Area Revues, Délirantes ou célibataires, les machines dans l'art*, N°11, Area corp, Paris, march 2006, ISBN: 2-35276-002-X, p.160.

[15] Les nouveaux gestes de la musique,Genevois, H., de Vivo, R., Parenthèses , collection eupalinos, cultures musicales may 1999, ISBN :2-86364-616-8.

[16] La Sculpture sonore, pratique artistique en recherche de définition, Rémus, J. In *Musiques Arts Technologies, pour une approche critique, Music arts tecnologies, toward a critical approach*", (Barbanti R., Lynch E, Pardo, C., Solomos, M), L'Harmattan, France, june 2004, ISBN : 2-7475-6691-9 , p.61-75.

MICON A Music Stand for Interactive Conducting

Jan Borchers
RWTH Aachen University
Media Computing Group
52056 Aachen, Germany
+49 (241) 80-21050
borchers@cs.rwth-aachen.de

Aristotelis Hadjakos
TU Darmstadt
Telecooperation Division
64289 Darmstadt, Germany
+49 (6196) 902520
thadjakos@gmx.de

Max Mühlhäuser
TU Darmstadt
Telecooperation Division
64289 Darmstadt, Germany
+49 (6151) 16-3709
max@informatik.tu-darmstadt.de

ABSTRACT
The MICON is an electronic music stand extending *Maestro!,* the latest in a series of interactive conducting exhibits that use real orchestral audio and video recordings. The MICON uses OpenGL-based rendering to display and animate score pages with a high degree of realism. It offers three different score display formats to match the user's level of expertise. A real-time animated visual cueing system helps users with their conducting. The MICON has been evaluated with music students.

Keywords
Music stand, score display, exhibit, conducting.

1. INTRODUCTION
Conducting is a well-established and rich metaphor when interacting with a musical body such as an orchestra. Some people even enjoy conducting alongside a classical recording at home. *Personal Orchestra* [3], an interactive exhibit for public spaces, was the first system to let museum visitors actually control tempo and volume of an audiovisual recording via conducting gestures.

Visitors can also emphasize an instrument section by conducting towards it. This gives visitors, albeit limited, opportunity to change the musical expression and play a more active part. *Personal Orchestra* has been an exhibit at the HOUSE OF MUSIC VIENNA since 2000 [3]. Two follow-up systems with new gesture recognition and audio time-stretching algorithms have been developed since: *You're The Conductor* opened at the Children's Museum Boston in 2003 [6], and the latest, *Maestro!,* at the Betty Brinn Children's Museum in Milwaukee in 2006 [7,8].

These exhibits provide a previously unavailable interaction to people who are not professional conductors: the experience of conducting an orchestra. To make this experience more realistic, we wanted to provide the visitor with a music stand as an extension to *Maestro!*. To this end, the MICON (**M**usic Stand for **I**nteractive **Con**ducting Systems) was created.

Figure 1. A visitor conducting the Vienna Philharmonic in the *Personal Orchestra* exhibit at the HOUSE OF MUSIC VIENNA (http://www.hdm.at). The music stands were still merely decorative here.

2. DESIGN CONSIDERATIONS
The functional goals of the MICON were:

- To display the musical score to *Maestro!* visitors while they are conducting, and
- To indicate the current position in the score and automatically advance the pages following the music, to help visitors with their conducting.

The key constraints in creating the MICON were:

- **Production quality.** Since the *Maestro!* system featured professional orchestras, its look and feel had to fulfill high standards to be accepted by the museum and orchestra in question. The MICON had to look and behave as professional as the rest of the exhibit, which required excellent visual quality of the score display and fluid, non-distracting interaction with it.
- **Visitor profile.** Typical visitors using the system would be one-time users with a short dwelling time. The system had to provide for this through a particularly simple, self-explanatory, and obvious interface that required little or no interaction apart from the conducting itself.
- **Musical knowledge.** Some visitors might be amateur or even professional conductors, but most would have no prior experience in conducting. The MICON had to provide these beginners with alternatives to the complexity of a full orchestral score document. Hopefully, by interacting with the system, visitors would learn a little

more about conducting and experience some of its challenges and rewards.

- **Listener architecture.** The MICON had to extend the *Maestro!* system; to minimize dependencies between these two co-evolving projects, their communication interface had to be kept as narrow as possible, with the MICON essentially listening to the timing information that was already being generated by the *Maestro!* gesture recognition engine.

Additional guidelines for interactive music exhibits that we followed in this design can be found in [2].

3. RELATED WORK

There has been a wealth of research aimed at recognizing conducting gestures, which we will not cover here as it is not the focus of this paper. Similarly, a variety of systems have been developed previously that offer a more or less "complete" conducting experience—often limited to audio or video only, and frequently using synthetic (MIDI or VRML, for example) data as opposed to real audio and video material for playback. We refer the interested reader to [5] and [7] for a more detailed overview.

MOODS (Music Object Oriented Distributed System) is a synchronous real-time cooperative editor for music scores [1]: Every change is immediately made visible to all users. It is intended for orchestra musicians during rehearsal. MOODS consists of different types of lecterns for the instrumentalists, the conductor, and the orchestra's archivist. Editing in MOODS is based on different permissions for these groups. MOODS supports semiautomatic page-turning. The score on the instrumentalists' lecterns is separated horizontally; the page that is currently being played is shown below a separator, the following page appears above it. As the music advances, the separator moves downwards. The score on the conductor's and archivist's lectern is separated vertically between the current and next page. MOODS assumes a constant tempo for a piece; tempo variations have to be adjusted manually by a human operator during the performance, making it unsuitable for our purposes.

muse is a digital music stand for symphony orchestras [4]. It consists mainly of a portable display and a matching stand with integrated metronome and a pitch-generating tuner. It allows on-screen annotation and intrasymphonic wireless communication. Pages are turned automatically or manually. In automatic mode, an attached microphone captures the incoming sound, and the *muse* compares it against the score to turn pages at the appropriate moment. Instrumentalists load pieces from an archive over an encrypted connection to prevent copyright violation. *muse* was presented to the Pittsburgh Symphony Orchestra. However, it does not highlight the current score position, and its page-turning requires per-instrument microphones.

The commercial *eStand* (www.estandmusic.com) consists of a tablet PC and footswitch for manual page-turning. The system is optimized for low noise. The software displays the score, which has to be downloaded from the Internet or created with note-setting software. The musician can annotate the score with the provided pen. The eStand has a built-in metronome, tuning, music library management, and networked annotation sharing for ensembles. eStand does not highlight the current position in the score and does not advance the score automatically.

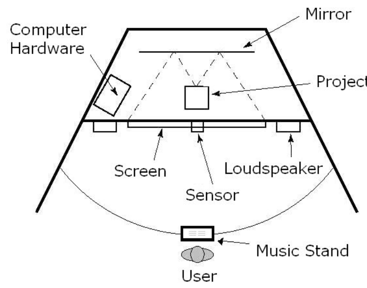


Figure 2. Typical conducting exhibit setup.

4. DESIGNING THE MICON

We will start with our envisioned usage scenario. Fig. 2 shows the layout of a typical conducting exhibit installation. The user stands in front of a large screen. Two loudspeakers are directed towards her. The computer hardware is hidden. In front of the user stands the music stand. A sensor tracks the baton and sends its position to the computer.

First, the user sees a list of music pieces on the large screen, points the baton towards the desired item and pushes the button on the baton. In the same way she selects the representation of the musical material on the music stand. She has three choices: full score, piano part score, or piano roll. In either case, the MICON shows the pulse notation at the top of the selected representation. After the selection, the orchestra appears, the user begins to conduct, and the orchestra starts playing, with the score display advancing automatically on the MICON.

A full score is an assembly of all the instruments' voices and the standard format that conductors use. There is a lot of information present in a full score: the notes for every instrument, dynamic markings, etc., and it requires very good music-reading skills. An extract of the full score for the piano is easier to read, as there are only two note systems, left hand and right hand. The piano-roll and the pulse notation do not require any score-reading abilities.

4.1 Score Animation and Highlighting

Fig. 3 shows the first page of the "Blue Danube" by Johann Strauss as it is presented on the MICON. The score is enhanced with additional information: An orange bar cursor marks the position of the music. While music plays, the cursor moves to the right. Above the cursor is a ball jumping up and down. This ball marks the beats in the music. The beats occur when the ball hits the "ground" (Fig. 4). The user can emphasize an instrument section by conducting towards it. This section is then highlighted on the music stand by coloring the corresponding lines red. The pages on the MICON do not have a flat appearance but are rendered more like naturally flexible paper. When the cursor reaches the last beat of the right page, the page lifts up (Fig. 5), turns (Fig. 6), and the next two pages become visible (Fig. 7).

When the cursor moves to another note system, it fades out smoothly. After it vanishes from the previous note system, it smoothly fades in at the new note system. The same happens to the jumping ball and the instrument group highlight during transition. When the jumping ball fades out completely at the end of the note system's last bar, it is located at the highest point of its trajectory. In the next note system the ball reappears at the highest point and moves downward while fading in (Fig. 8).

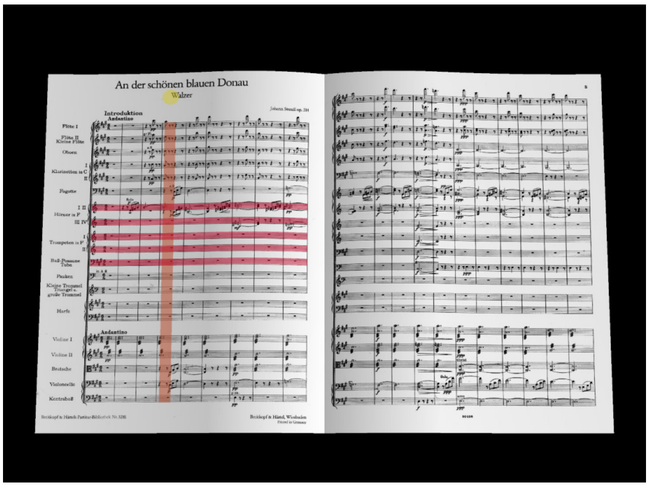


Figure 3. The "Blue Danube" on the music stand.

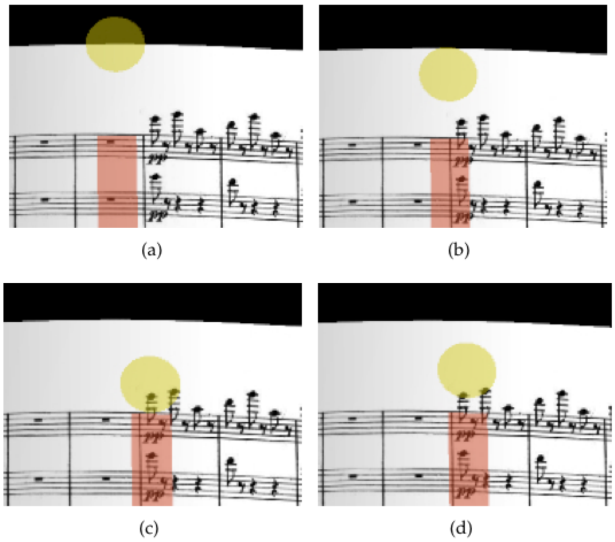


Figure 4. The jumping ball: (a) and (b) before the beat, (c) at the beat, (d) after the beat.

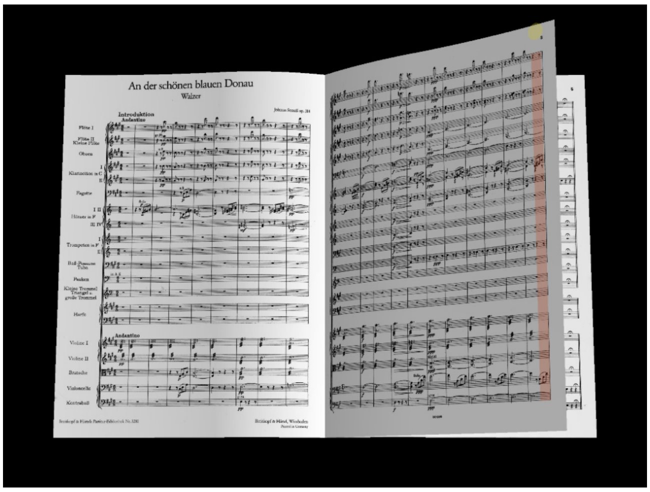


Figure 5. Page turning begins.

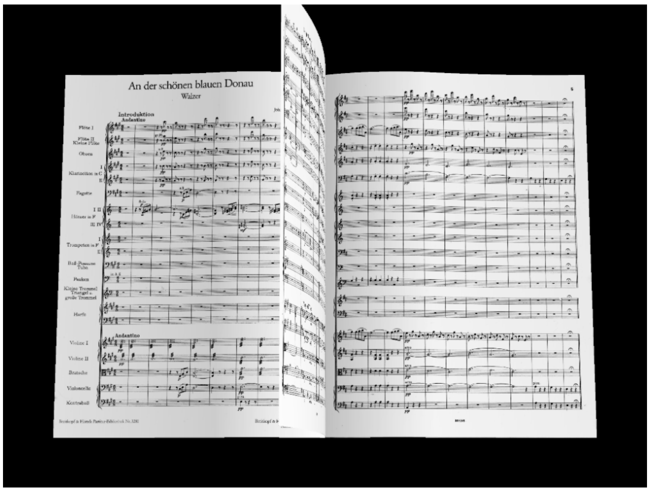


Figure 6. Page is halfway turned.

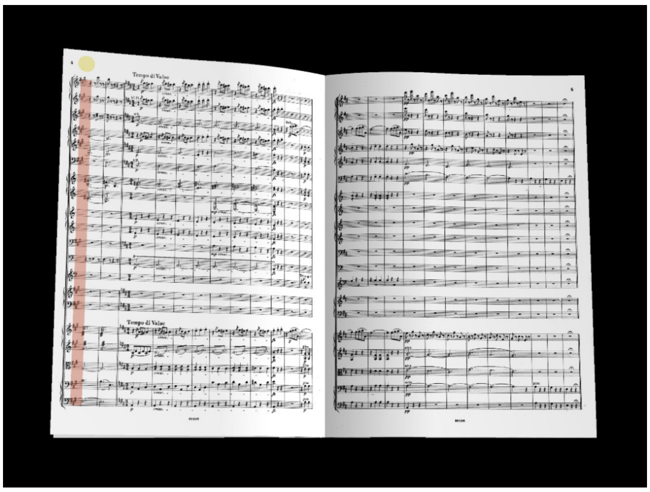
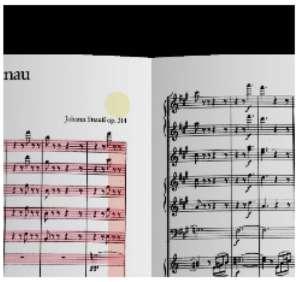
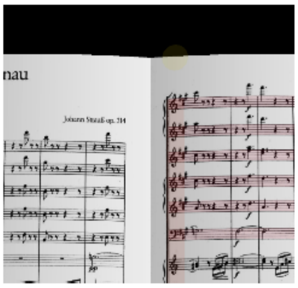


Figure 7 Page turning ends. The bending behavior of the page in the above panels aims to closely resemble physical paper.

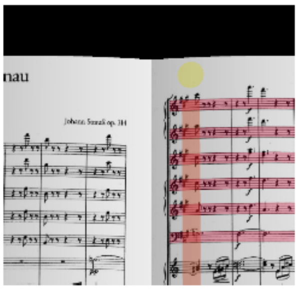
(a) Fading out begins.



(b) The ball reaches the highest point.



(c) The cursor reappears on the next page.



(d) The process is nearly finished.

Figure 8. The cursor moves to another note system.

4.2 Piano Roll Notation

The piano roll notation represents notes as boxes of uniform height. Box width represents the length of a note, the vertical position represents its pitch, and the horizontal position its onset time. Music software such as MIDI sequencers frequently uses piano roll notation to represent musical material, so that some users of the MICON may already be familiar with this notation. Fig. 9 shows the piano roll representation of bars 45–49 of the "Blue Danube" as it is presented on the MICON.

The notes that are currently played by the orchestra are in the center of the screen. In addition, they are highlighted in a brighter color. As time proceeds, the boxes move to the left, and the highlight moves to the following notes. The highlight moves rhythmically, mimicking the rhythm of the music. A user without prior experience with the piano roll notation can thus figure out the connection between this notation and the music.

4.3 Pulse

The pulse notation consists of circles representing the beats of the music. They are horizontally aligned. The more time there is between two beats, the longer is the distance between the two corresponding circles. Tempo changes lead to differing distances between the circles: the pulse notation outlines the tempo changes of the piece: It shows the metrical landscape of the music around the current beat, giving the user an indication of the recently passed and the upcoming tempo. Fig. 10 shows a metric landscape for a steady tempo, Fig. 11 shows a tempo decrease (ritardando).

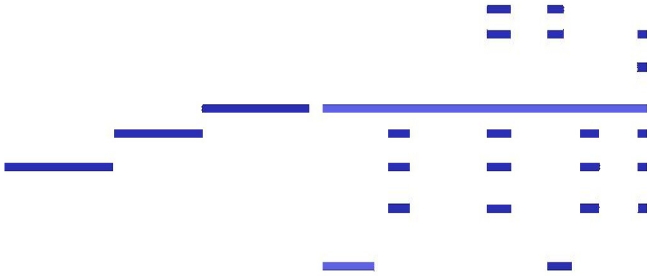


Figure 9. Bars 45–49 of the "Blue Danube" in piano roll notation. The beginning of the theme is still recognizable.

Figure 10. Steady tempo.

Figure 11. Tempo change (in a ritardando).

The circles move to the left as time proceeds. Coming from the right and approaching the center of the screen, they grow in size. Far from the center they grow only slowly, but as they approach the center, they grow at a higher rate. They reach their maximum size in the center of the screen, which marks the beat. Afterwards, they shrink again, first rapidly, then slowly, to their original size (Fig. 12, 13, and 14). Mathematically speaking, circle size is computed as the absolute value of a modified $1/x$ function, with x=0 at the center, and clipping the size to a maximum size at that point.

Figure 12. The circle approaches from the left (still original size).

Figure 13. The circle is in the center (maximal size). The beat occurs.

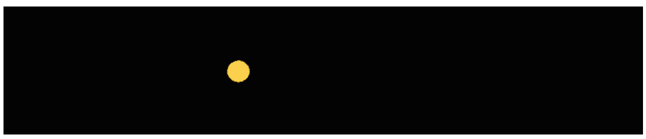


Figure 18. Circle in the middle barely visible. The other circles begins to move.

These animations are hard to describe on paper; therefore, we have created a screen capture that shows the system in use. See: http://media.informatik.rwth-aachen.de/micon.html.

5. IMPLEMENTATION

5.1 Preparing Music For The MICON

To add a music piece to the MICON requires to create four types of files:

- A MIDI file,
- A beats file,
- Page scans of the full score and the piano part, and
- ScoreInfo files for the full score and the piano part.

The piano roll and the recorded (digital audio) music have to be synchronized. Therefore, MIDI files were produced that closely resemble the tempo changes of the original recording. We created these files by playing alongside the orchestral recording on a MIDI keyboard and recording this performance to a MIDI file. Manual post-processing was done to correct minor errors and to achieve a better synchronization. In a second manual post-processing step, this synchronized MIDI file was refined under aesthetic considerations: The notes were quantized to a fixed raster and aligned so that notes belonging to a logical compound begin and end in the correct position. When, for example, a chord is played, the notes constituting the chord should all start and end at the same horizontal position. For *legato* passages, neighbouring notes should end respectively start at exactly the same horizontal position.

A beats file contains a list of time stamps indicating when each beat occurs in the orchestral recording. This allows the MICON to compute a mapping between the elapsed time (in seconds) and the position in the music (in beats). This mapping is required for the score and pulse representations. The beats file for a new piece can be created with *Midi2Beats*, a program we developed. *Midi2Beats* extracts the rhythm of a MIDI-file: Every *Note-On* in the MIDI file is stored as a beat event in the beats file. Therefore, a MIDI file has to be created that has exactly one *Note-On* on every beat. By deleting notes from and, sometimes, adding notes to the previously created MIDI file, it can be transformed into such a file. Using this file, the beats file is created using *MIDI2Beats.*

The ScoreInfo file provides graphical information about the score layout on the page. This is mainly:

- The *x/y* position of every beat,
- The position and extent of each note system and
- The position and extent of each instrument group on the score page (full score only).

We developed the *ScoreMarker* application (Fig. 19) to easily specify this information. First, the developer loads the graphical page scans into ScoreMarker. Using the mouse, he defines the note systems, beats and instrument groups in the score. *ScoreMarker* then creates the corresponding ScoreInfo file.

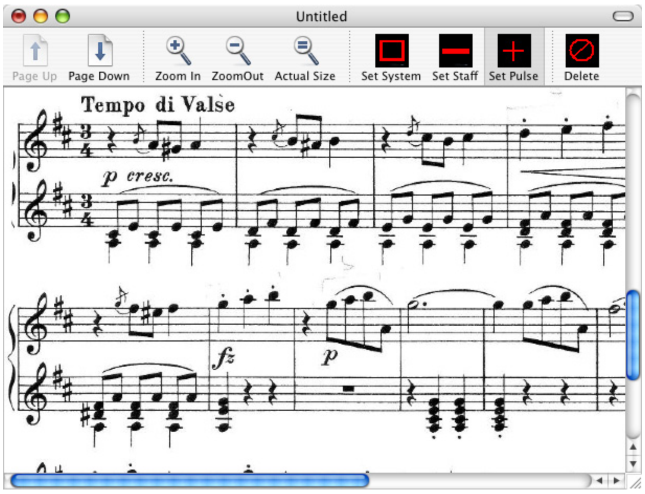


Figure 19. The *ScoreMarker* application.

5.2 Communication

Maestro! and the MICON communicate via a UDP-based protocol. Communication is one-way: *Maestro!* sends messages to the MICON which then updates its internal state. During the initial user selections, *Maestro!* informs the MICON about what piece and score representation the user chooses. During the piece, *Maestro!* continuously sends the current position in the piece to the MICON: every few seconds *Maestro!* sends the current instrument emphasis.

5.3 Rendering

MICON uses OpenGL for its graphical output and creates realistic-looking, curved score pages resembling physical paper more closely than a normal “flat” 2-D rendering would. Fig. 19 shows a curved score page. To create these pages, each page is sliced vertically into 100 equidistant pieces, which are then reconnected with their two neighbors to form the curved page. The angles between these neighboring slices were defined beforehand for several key frames of the page-turning animation. We experimentally chose values for these 99 inter-slice angles at each key frame. Rendering the slices for a given key frame is then straightforward: Starting at the origin, a slice is drawn along the x axis (the coordinate system is shown in Fig. 20). The origin is then translated along the x axis so that the origin now points to the right end of the drawn slice. Now, the origin is rotated around the y axis with the predefined angle, and the rendering of the next slice begins.

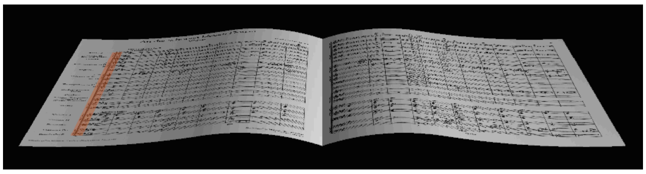


Figure 19. Score from the side.

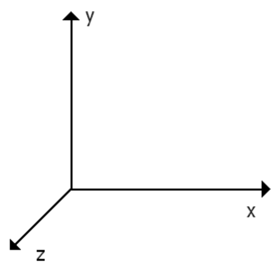


Figure 20. Reference coordinate system for rendering the above page.

As the page turns, it changes its form: At the beginning (0°) and end (180°) of the page turn, the page has the "normal" shape shown in Fig. 19. In addition to these two key-frames, a configuration of the 99 angles was specified for a page turned 90 degrees. These angles were again chosen experimentally. For all other frames, the 99 inter-slice angles smoothly transition between these key frame configurations using linear interpolation.

6. EVALUATION

We tested advanced prototypes of the MICON and *Maestro!* with music students at the University for Music and Performing Arts ("Hochschule für Musik und Darstellende Kunst") in Frankfurt, Germany. The goal was to get feedback about the *Maestro!* system in general, and the MICON in particular. Ten students participated in the test: four pianists, four other instrumentalists (violinist, cellist, flautist, guitarist), and one composer.

Page turning speed originally only depended on the tempo of the music. It would start when the last beat of the right page was reached, and end with the first beat on the new page. When users conducted very slowly, this made page turning unnaturally slow, as two students criticized. MICON now completes page turning in two seconds maximum—or less if the tempo is above 30 bpm.

Many students were uncertain how to start conducting, for example, whether to conduct quarter or half notes. To help with this, the pulsating circle described in section 4.3 was added.

The students liked the appearance of MICON. Especially the curved pages and the animated page turning were appreciated.

7. SUMMARY

We presented the MICON, a music stand for *Maestro!,* an interactive conducting exhibit. The MICON displays musical material in various formats: full score, piano extract score, piano roll, and pulse notations. This enables a broad public to use the MICON, while still allowing expert music score readers to benefit from their abilities. Several carefully designed animation techniques help users with their conducting. The MICON renders its output via OpenGL to create natural-looking curved pages and a realistic-looking page-turning mechanism. The MICON was tested successfully with music students, and their feedback incorporated into the final design.

8. FUTURE WORK

We intend to conduct more formal user studies to evaluate the MICON, and to make content creation less tedious. In particular, a (semi-) automatic way to create the piano roll MIDI files would greatly simplify this very time-consuming manual process.

9. ACKNOWLEDGEMENTS

The authors would like to thank Henning Kiel and Eric Lee for their help with integrating the MICON with *Maestro!*.

REFERENCES

[1] Bellini, P., Nesi, P., and Spinu, M. B. Cooperative visual manipulation of music notation. *ACM Trans. Comput.-Hum. Interact.,* 9, 3 (2002), 194–237.

[2] Borchers, J. *A Pattern Approach to Interaction Design.* John Wiley & Sons, Chichester, UK, 2001.

[3] Borchers, J. O,, Samminger, W., and Mühlhäuser, M. Engineering a realistic real-time conducting system for the audio/video rendering of a real orchestra. *IEEE Fourth International Symposium on Multimedia Software Engineering* (Newport Beach, CA, Dec. 11–13, 2002).

[4] Graefe, C., Wahila, D., Maguire, J., and Dasna, O. muse: a digital music stand for symphony musicians. *Interactions,* 3, 3 (1996), 26–35.

[5] Kolesnik, P. *Conducting Gesture Recognition, Analysis and Performance System.* M.Sc. Thesis, McGill University, 2004.

[6] Lee, E., Nakra, T. M., and Borchers, J. You're the Conductor: A Realistic Interactive Conducting System for Children. *NIME 2004 International Conference on New Interfaces for Musical Expression* (Hamamatsu, Japan, June 2004), 68–73.

[7] Lee, E., Karrer, T., and Borchers, J. Toward a Framework for Interactive Systems to Conduct Digital Audio and Video Streams. *Computer Music Journal,* 30,1 (Spring 2006), 21–36.

[8] Lee, E., Kiel, H., Dedenbach, S., Grüll, I., Karrer, T., Wolf, M., and Borchers, J. iSymphony: An Adaptive Interactive Orchestral Conducting System for Digital Audio and Video Streams. *CHI 2006 Extended Abstracts (Montréal, Apr. 22–27, 2006),* ACM Press (in print).

conga: A Framework for Adaptive Conducting Gesture Analysis

Eric Lee, Ingo Grüll, Henning Kiel, and Jan Borchers
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
{eric, gruell, kiel, borchers}@cs.rwth-aachen.de

ABSTRACT

Designing a conducting gesture analysis system for public spaces poses unique challenges. We present *conga*, a software framework that enables automatic recognition and interpretation of conducting gestures. *conga* is able to recognize multiple types of gestures with varying levels of difficulty for the user to perform, from a standard four-beat pattern, to simplified up-down conducting movements, to no pattern at all. *conga* provides an extendable library of *feature detectors* linked together into a directed acyclic graph; these graphs represent the various conducting patterns as gesture *profiles*. At run-time, *conga* searches for the best profile to match a user's gestures in real-time, and uses a beat prediction algorithm to provide results at the sub-beat level, in addition to output values such as tempo, gesture size, and the gesture's geometric center. Unlike some previous approaches, *conga* does not need to be trained with sample data before use. Our preliminary user tests show that *conga* has a beat recognition rate of over 90%. *conga* is deployed as the gesture recognition system for *Maestro!*, an interactive conducting exhibit that opened in the Betty Brinn Children's Museum in Milwaukee, USA in March 2006.

Keywords

gesture recognition, conducting, software gesture frameworks

1. INTRODUCTION

Orchestral conducting has a long history in music, with historical sources going back as far as the middle ages; it has also become an oft-explored area of computer music research. Conducting is fascinating as an interaction metaphor, because of the high "bandwidth" of information that flows between the conductor and the orchestra. A conductor's gestures communicate beat, tempo, dynamics, expression, and even entries/exits of specific instrument sections. Numerous researchers have examined computer interpretation of conducting gestures, and approaches ranging from basic threshold monitoring of a digital baton's vertical position, to more sophisticated approaches involving artificial neural networks and Hidden Markov Models, and even analyzing data from multiple sensors placed on the torso, have been proposed.

Figure 1: *Maestro!*, an interactive conducting exhibit for children that we developed, at the Betty Brinn Children's Museum in Milwaukee, USA. *Photo appears courtesy of the Betty Brinn Children's Museum in Milwaukee, WI, USA.*

Our work is motivated by research on novel computer music and multimedia interfaces for public spaces such as museums (see Figure 1), and *conga* builds on our prior experience with designing interactive orchestral conducting exhibits, including *Personal Orchestra*, an exhibit for the HOUSE OF MUSIC in Vienna (coordinated by Max Mühlhäuser, now at Darmstadt University) [1] and a collaboration with Teresa Marrin Nakra on *You're the Conductor*, a children's exhibit for the Boston Children's Museum [9]. Our systems allow the user control over tempo, by making faster or slower gestures; volume, by making larger or smaller gestures; and instrument emphasis, by directing the gestures towards specific areas of a video of the orchestra on a large display (instrument emphasis is not supported in *You're the Conductor*). Designing a gesture recognition system for a museum-type environment poses unique and interesting challenges, primarily because museum visitors have a wide range of experience with conducting. Moreover, there is little to no opportunity to either train a user to use the system, or to train the system to a specific user; a museum on a busy day may see over 1000 visitors, and so a visitor will spend, on average, less than one minute at an exhibit.

In this paper, we present *conga*, a system for **con**ducting **g**esture **a**nalysis. Unlike current systems, *conga* does not restrict the user to conduct in a specific way, nor does the system itself require training to tune itself to a user's specific movements; instead, it continuously evaluates user gestures against a set of gesture *profiles*, which are encoded with the characteristic features of particular types of gestures, and uses the best-matching profile to extract information such as beat, tempo, size, and center from the user's gestures.

We begin with a more detailed description of the scope and requirements for *conga*, followed by a quick survey of existing work in the area of conducting gesture recognition. Then, we describe our design of *conga*, and provide some implementation-specific details and challenges. We conclude with a discussion of some preliminary results obtained by testing *conga* with users.

2. REQUIREMENTS AND SCOPE

Our target user group for this work is museum visitors, and thus, one of our primary goals was to build a gesture recognition system that works for a wide spectrum of users. We also wanted to accommodate people with a wide variety of musical/conducting knowledge. This led to requirements that *conga* be able to:

- recognize gestures from a user without any prior training (either for the user or for the system).
- recognize a variety of gestures to accommodate different types of conducting styles.

One of our goals was also to design *conga* as a reusable component of a larger system that requires gesture recognition; thus, we also required *conga* to:

- integrate well with a computationally-expensive rendering engine for digital audio and video.
- not depend on the specific characteristics of any particular input device.

While conducting is an activity that typically involves the entire body [11], it is generally agreed that the most important information is communicated through the hands [6, 17]. Since we also intended *conga* for use in a public exhibit, we have thus far limited our gesture analysis with *conga* to input from the user's dominant hand. The output of the gesture analysis consists of four parameters: rate (tempo), position (beat), size (volume), and center (instrument emphasis). It is important to note, however, that the design of *conga* itself does not place any restrictions on the types of inputs or outputs, although we leave the implementation of such extensions for future work.

3. RELATED WORK

Gesture-controlled conducting systems have a long history in computer music research. Mathews' early work on the *Radio Baton* [12], which triggers a beat when the baton goes below a certain vertical position, has inspired a number of researchers to study conducting as an interface to computer music systems.

Ilmonen and Takala's *DIVA* system [5] features a conductor follower that is capable of classifying and predicting beats, and even sub-beats, to control tempo and dynamics. The system uses artificial neural networks, and needs to be trained with user data prior to use.

Usa and Mochida's *Multi-modal Conducting Simulator* [18] analyzes two-dimensional accelerometer data using Hidden Markov Models and fuzzy logic to detect beats in gestures. The system features beat recognition rates of 98.95–99.74%, although it also needs to be trained with sample data sets prior to use.

Murphy *et al.*'s work on conducting audio files uses computer vision techniques to track tempo and volume of conducting gestures [14]. Users' movements are fitted to one of several possible conducting *templates*, described in [13]. While the system does not require any training, the user must be familiar with the gesture templates. Murphy used a combination of C code and EyesWeb [3], a library for gesture processing.

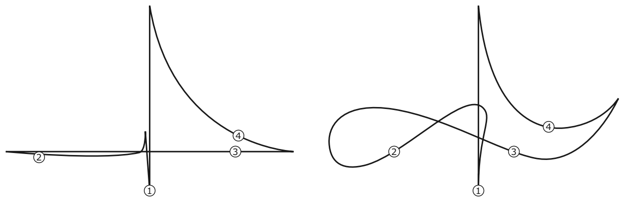


Figure 2: Beat patterns for the four-beat neutral-legato (left) and expressive-legato (right), as described by Rudolf. The numbers indicate where beats are marked in the gestures.

Marrin's *Conductor's Jacket* [11] collects data from sensors along the arms and upper torso, measuring parameters such as muscle tension and respiration. She was primarily interested in mapping expressive features to sections in the music score, rather than obtaining measurements on how movements map to rhythm and beats. In her later collaboration with us on *You're the Conductor* [9], she developed a gesture recognition system that mapped gesture velocity and size to music tempo and dynamics. Her systems were built using LabVIEW [15], a graphical development software for measurement and control systems.

Kolesnik's work also uses Hidden Markov Models for recognizing conducting gestures [6], although the focus of this work was on expressive gestures with the off-hand rather than beat recognition with the dominant hand. His conducting system was built using a combination of EyesWeb and Max/MSP [16].

Our system is thus unique in the following ways:

- the system does not need to be trained prior to use, unlike those that use artificial neural networks and Hidden Markov Models.
- users are not required to learn or be proficient with specific gestures before using the system.
- the system interprets multiple types of gestures, allowing it to respond to the precise gestures of a conductor's four-beat conducting pattern as well as the potentially erratic movements of a child.

4. DESIGN

The design of *conga* is inspired by Max Rudolf's work on the grammar of conducting [17]. In his book, he models conducting gestures as two-dimensional beat patterns traced by the tip of a baton held by the conductor's right hand (see Figure 2). Conducting, then, is composed of repeating cycles of these patterns (assuming the user keeps to the same beat pattern), with beats corresponding to specific points in a cycle. By analyzing certain features of the baton's trajectory over time, such as trajectory shape or baton movement direction, we can identify both the specific pattern, and the position inside the pattern, as it is traced by the user.

Unlike Murphy's work on interpreting conductors' beat patterns [13], we do not try to fit the user's baton trajectory to scaled and translated versions of the patterns shown in Figure 2; as a majority of our target user base are not proficient conductors, such a scheme would most probably not work very well for them; in fact, we have found in previous work that even after *explicitly* instructing users to conduct in an up-down fashion, the resulting gestures are still surprisingly diverse [10]. Murphy also makes use of the dynamics encoded in the music that the user is conducting to differentiate between unintentional deviation from the pattern and intentional deviation to change dynamics; the ability to make this distinction requires one to assume that the user is already familiar with the music (an assumption we are unable to make).

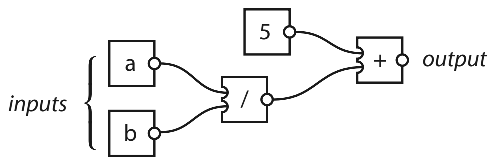


Figure 3: A basic *conga* graph to compute $5 + \frac{a}{b}$. The addition triggers a division, which then in turn pulls data from the inputs a and b.

Our general approach is to instead identify specific characteristics (*features*) in various types of gestures, such as turning points in the baton position with a certain direction or speed. These features are encoded into gesture *profiles*, and the features are triggered in sequence as the user moves the baton in a specific pattern. The advantage of this approach is that the system does not require the user to perform the gesture too exactly; as long as the specific features of the gesture can be identified in the resulting movements, the overall shape of the gesture is unimportant.

conga, as a software framework, allows a developer to work at several layers of abstraction; at the most basic level, it provides a library of feature detectors. These feature detectors can then be linked together into a more complex graph to identify specific gesture profiles, and to date, we have encoded three types of gesture profiles into *conga*, with increasing levels of complexity: wiggle (for erratic movements), up-down (for an inexperienced conductor, but one who moves predictably), and the four-beat neutral-legato (for the more experienced conductor). Finally, we have developed a profile selector that evaluates which of these profiles best matches the user's baton movements at any given time, and returns the results from that profile.

4.1 Feature Detectors

conga's library of feature detectors offers basic building blocks that provide a specific function; for example a bounce detector may detect a change in the baton's direction. Each feature detector *node* has one or more input ports and at least one output port. It takes, as input, a continuous stream of data (e.g., two-dimensional position of a baton). The output is a "trigger", a Boolean value that is true when the feature is detected, and false otherwise. There may be other outputs from the feature detector, so that any nodes that use the output from the feature detector can obtain more information regarding what caused the feature detector to trigger. Other types of nodes also exist to manipulate data, such as rotating the data about an axis, applying various types of filters, etc.

These nodes are connected into directed, acyclic graphs. The graph is evaluated using a pull model, where the output requests data which then pulls on its input nodes to perform the necessary computation (see Figure 3).

This graph-based approach has been used successfully in a number of existing frameworks, including LabVIEW, EyesWeb, and Max/MSP. While it would have been possible to build *conga* as a layer on top of one of these systems, we decided against such a solution after evaluating each of these three systems. *conga* was envisioned from the beginning as a component of a larger system for conducting digital audio and video streams running on Mac OS X [7]; of the three aforementioned frameworks, only Max/MSP runs on the Mac, but, unfortunately, Max/MSP does not provide all of the basic building blocks that we needed to implement *conga*. An alternative would be to use a two-machine solution, such as in [6, 1, 9], although we have learned from prior experience that such setups are awkward to maintain in a museum setting. Nevertheless, we feel these are implementation-specific details, and we emphasize that *conga*'s contribution to the research community is not just the feature detector framework, but our design of a system for conducting gesture analysis based on such a framework.

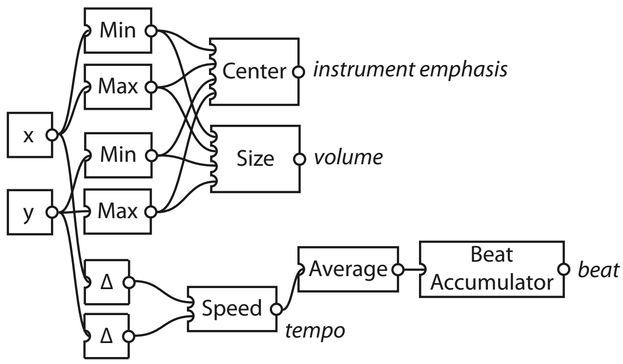


Figure 4: The *conga* graph for the Wiggle gesture profile. The gesture speed determines tempo, gesture size determines volume, and the gesture's geometric center determines instrument emphasis.

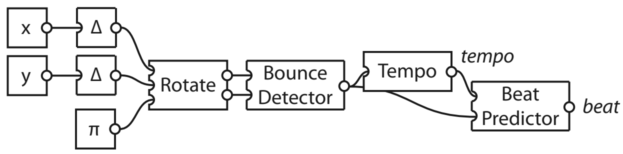


Figure 5: The *conga* graph for the Up-Down gesture profile. The downwards turning points of the gestures correspond to beats; a beat predictor beat values in between these values.

Further details of the feature detector framework and types of nodes it provides are given in [4]; we will discuss only the feature detectors of relevant interest to our discussion here.

The next three subsections describe the three profiles that we have built for *conga* using our feature detector library.

4.2 Wiggle Profile

Figure 4 shows the *conga* graph for the Wiggle profile, which is the most fundamental of the three gesture profiles that *conga* recognizes. Inspired by Teresa Marrin Nakra's work on *You're the Conductor* [9], gesture speed is mapped to tempo, gesture size is mapped to volume, and the geometric center of the gesture determines instrument emphasis (see Figure 4). *conga* falls back to this profile when it cannot use any other means to interpret the user's gestures.

The "x" and "y" nodes hold time-stamped positional data from the baton that has been preprocessed to remove noise. *conga* assumes the origin is at the lower left of the coordinate system. From there, "min" and "max" nodes store the most recent minimum and maximum values of the baton position; these are then used to determine the gesture size and center.

The gesture speed is computed by taking a numerical time derivative of the baton position, followed by a moving average of this derivative. Since the gestures themselves are not synchronized to the music beat, a numeric integral of the speed is used to arbitrarily derive beat information from the gesture speed.

4.3 Up-Down Profile

The Up-Down profile tracks the vertical movement of the user to determine beat and tempo (the method for deriving gesture size and geometric center remain the same as Wiggle, and will not be repeated here). Figure 5 shows the *conga* graph for the Up-Down profile.

The primary feature that is detected is the downwards turning point, using the "bounce detector" node. The bounce detector node takes, as input, the current velocity of the baton, and looks for a positive to negative zero crossing in the y component of the velocity (i.e., an upside-down "U" shape). Since such a detector would normally track the *upwards* turning point, the data from the baton must first be rotated by 180 degrees. To prevent false triggers, the bounce detector imposes a criterion that the magnitude of the vertical movement over the last few samples be some multiple of the magnitude of the horizontal movement (set as optional parameters in the bounce detector node).

The triggers sent by the bounce detector mark whole beats, and so the tempo can be derived by taking the numerical time derivative of these beat positions over time. This tempo is then used to predict the current fractional beat value until the next trigger occurs. If r is the current tempo in beats per minute, and t_0 is the time of beat b_0 in seconds, then our predicted fractional beat value b for time t is computed using $b = b_0 + \frac{r}{60}(t - t_0)$. We also impose the additional constraint that $b < b_0 + 1$ until the next trigger occurs, to ensure that beats are always monotonically increasing.

We found this simple beat prediction algorithm to work well for estimating the fractional beat values between beats in early prototypes of *conga*. While the beat prediction could be improved if we detected more features in the gesture (e.g., detecting the upper turning point to mark the halfway point into the beat, in addition to the lower turning point), doing so would also place more constraints on the types of movements that would fit the profile. For example, we found that many users naturally tend to conduct "pendulum-style", rather than in strictly vertical up-down movements.

4.4 Four-Beat Neutral-Legato Profile

The Four-Beat Neutral-Legato profile is the most complex, and unsurprisingly, the most challenging beat pattern to detect. Multiple features are detected in parallel, which then drive a probabilistic state machine to track where in the four-beat pattern the user currently is at (see Figure 6).

The features that are detected are: the downwards turning point at beat 1; the upper turning point just after beat 1; the change in horizontal direction just after beat 2; the change in horizontal direction just after beat 3; and the upper turning point after beat 4. Note that the features detected do not necessarily correspond to the beats themselves (see Figure 6).

The first and third features are very distinct sharp turns, and so the bounce detector is again used to track these features. The second feature tends to be more subtle, and thus we look only for a zero crossing in the baton's vertical velocity at that point, without the additional constraint that the bounce detector imposes, as described earlier. Finally, the fourth and fifth features also have a softer curvature, and are also tracked with a zero crossing node. Since zero crossing nodes trigger on both positive to negative, and negative to positive transitions, the undesired trigger is filtered out before sending it to the state node. The state machine node tracks the progress through the beat cycle; it also detects and compensates for missed or false beats using a probability estimation based on the current tempo and time in which the last trigger was received. For example, if we are currently in state 4, and the state machine receives a trigger for state 1, it checks to see how much time has elapsed, and together with the current tempo, guesses what the correct state should be. If it appears that the feature for state 5 was just simply not detected, the state machine will jump directly to state 1. Otherwise, it will assume the trigger for state 1 was simply a falsely detected trigger and ignore it.

The state machine node also acts as a beat predictor; however, unlike the beat predictor in the Up-Down profile, which receives

```
CONGANode *a, *b, *div, *five, *plus;

// Inputs nodes.  Their values will be set externally.
a = [[CONGAPassiveValueNode alloc] initWithTime:0.0f];
b = [[CONGAPassiveValueNode alloc] initWithTime:0.0f];

// Division node.
div = [[CONGADivisionNode alloc] initWithTime:0.0f];
[div setInputPorts:[NSArray arrayWithObjects:a, b]];

// A node with a constant value.
five = [[CONGAPassiveValueNode alloc] initWithTime:0.0f];
[five setValue:5.0f];

// Addition node.
plus = [[CONGAAdditionNode alloc] initWithTime:0.0f];
[plus setInptPorts[NSArray arrayWithObjects:five, div]];
```

Figure 7: The source code corresponding to the basic *conga* graph shown in Figure 3, which computes $5 + \frac{a}{b}$.

whole beat information and predicts beat values in between the whole beats, the state machine receives *fractional* beat information – this is to compensate for the phase shift between the beats and features in the gesture cycle. For the four-beat pattern, beats 1 to 4 are at 0, 0.25, 0.5 and 0.75 (percentage of one whole cycle), respectively, while the features occur at values of 0, 0.12, 0.31, and 0.63 (see Figure 6).

4.5 Profile Selection

The three gesture profiles described above run concurrently in *conga*, and the final step in interpreting the user's gestures is a profile selection scheme that decides which of the profiles is returning the most reasonable data. Our algorithm for performing this selection is based on the assumption that the user does not make erratic changes to the tempo; our informal observations of users using our prior systems have confirmed that users moving in an up-down gesture, or a four-beat neutral-legato pattern must exert considerable effort to make relatively sudden changes to the conducting pattern, and thus, the conducting pattern is usually quite regular.

At each regular update cycle, each of the profile graphs is evaluated to determine the current beat. A threshold value is computed based on the standard deviation of the last four calculated beat values, and a confidence value returned by the beat predictor for each of the profile graphs. If this value falls below a certain threshold, we conclude that the profile is returning a sensible result. Profiles are also given a precedence order, so that if more than one profile falls below the given threshold, the one with the highest precedence wins. Our order of precedence from highest to lowest is: four-beat neutral-legato, up-down, and wiggle.

5. IMPLEMENTATION

conga was implemented using the Objective-C programming language under Mac OS 10.4 with the support of Apple's AppKit libraries. Nodes and graphs are created programmatically rather than graphically, a departure from systems such as LabVIEW, EyesWeb, and Max/MSP (see Figure 7). While it is possible to build a graphical editor to create *conga* graphs, such an editor was beyond the scope of this work. *conga* builds into a standard Mac OS X framework, making it easy to include it as part of a larger system, as we have done with *Maestro!*, our latest interactive conducting system [8].

conga graphs are evaluated at 9 millisecond intervals; computation occurs on a separate, fixed-priority thread of execution.

We tested early prototypes of *conga* with a Wacom tablet. Our current implementation uses a Buchla Lightning II infrared baton system [2] as the input device. While the design of *conga* itself is device agnostic, the specific characteristics of the Buchla did

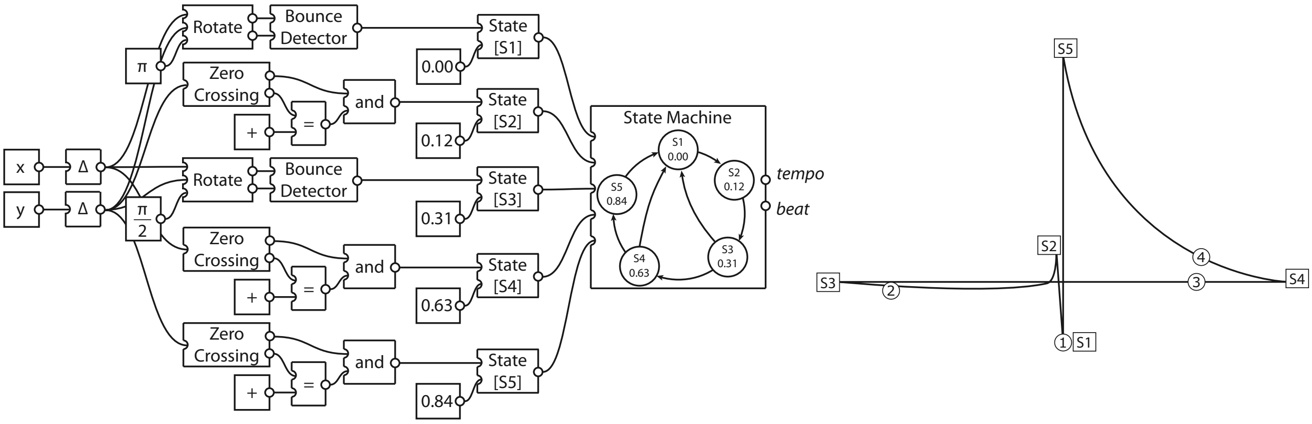


Figure 6: The left figure shows the *conga* graph for the Four-Beat Neutral-Legato gesture profile. Five features are detected, which are used to trigger the progress of a state machine that also acts as a beat predictor. The input to the state machine is the current progress (0 to 1) of the baton as it moves through one complete cycle of the gesture, starting at the first beat. The right figure shows the corresponding beat pattern that is tracked; numbered circles indicate beats, squared labels indicate the features that are tracked and the state that they correspond to.

Table 1: Summary of latency results for the Up-Down profile.

User	Avg Tempo [bpm]	Latency [ms]		
		Min	Max	Avg
1	54	36	117	96
2	58	63	117	86
3	96	81	144	113
4	114	81	144	118
5	130	108	135	122

present some challenges during implementation. For example, data from a Wacom tablet is relatively high resolution and noise-free, compared to the Buchla Lightning II, which has a resolution of only 128 in both width and height. Data from the Buchla can also be quite noisy, and we experimented with different types of filtering to compensate. Based on these experiments, we found a combination of hysteresis filtering and a 32 point Hanning filter to denoise the data gives the best results. Unfortunately, this Hanning filter also adds between 4 to 10 samples of latency to the overall system (36 to 90 ms), and we are looking into alternative methods to reduce this latency without compromising overall accuracy.

6. EVALUATION

We conducted some preliminary testing with users to evaluate *conga*'s accuracy and response. We asked five users (four male, one female) to conduct using up-down movements, and three users (all male) to conduct using the four-beat neutral-legato pattern. The users conducted for approximately 30 seconds each. The three users conducting the four-beat pattern were already somewhat proficient with the gesture prior to the experiment.

The system starts by default using the Wiggle gesture profile; for all five users, the system switched to the Up-Down profile within the first two beats. After that, *conga* did not falsely detect any beats, nor miss any beats, in the user's gestures (100% recognition rate). We also measured *conga*'s overall latency by measuring the time difference between when the user marks a beat, and when it is detected by *conga*; the results are summarized in Table 1.

Since the smoothing that we apply to the Buchla baton data introduces an average 63 ms delay, we estimate *conga*'s latency to be between 23 and 59 ms. There also appears to be a correlation between tempo and latency, although more data points would be required to make a conclusive statement.

Table 2: Summary of latency results for the Four-Beat Neutral-Legato profile.

User	Avg Tempo [bpm]	Latency [ms]		
		Min	Max	Avg
1	104	27	675	175
2	96	72	666	203
3	98	18	225	107

For the Four-Beat Neutral-Legato pattern, we found that for one user, *conga* fell back to the Up-Down profile 8% of the time, and failed to detect his beats 6% of the time. For the other two users, *conga* stayed in the Four-Beat profile 100% of the time, and did not fail to detect any of their beats. The latency results are summarized in Table 2.

The maximum latencies for users 1 and 2 were particularly high; a closer analysis of the data showed that these high latencies occurred consistently on beat 3, and sometimes for beat 4. One possible explanation is that the users' unfamiliarity with the four-beat gesture confused the beat predictor, resulting in *conga* behaving unpredictably. For user 3, who is the most familiar with the gesture, *conga* fared significantly better. Again, we believe that more data points and detailed analysis would be required to make a conclusive statement.

7. FUTURE WORK

We have identified a number of areas that we are actively pursuing to further the development of *conga*:

More gesture profiles: We have currently implemented three gesture profiles in *conga*, which already illustrate the capabilities and potential for the framework. However, only one of these is actually a real conducting gesture, and thus we would like to incorporate more professional conducting styles, such as the four-beat expressive-legato pattern shown in Figure 2.

Improved profile selector: As we incorporate more gesture profiles, we will naturally have to improve the profile selector as well. For example, the four-beat neutral-legato and the four-beat full-staccato have very similar shapes, but their placement of the beat is different, as is the way in which they are executed.

Lower latency: While the current latency introduced by *conga* is acceptable for non-professionals, professionals will find the latency much more disturbing. One way to reduce the latency is to implement a better beat predictor, especially for the four-beat profile. Another method to reduce latency is to realize that *conga* can only detect a feature *after* it has occurred; thus, by the time the trigger is sent, we are already at some future point in time. *conga* nonetheless reports the feature as having triggered "now", and by compensating for this time delay from when the feature actually happened to when it is detected in the beat predictor, we can reduce the perceived latency further. Again, this would require a more sophisticated beat predictor.

Graphical *conga* editor: *conga* graphs are currently created programmatically rather than graphically, like in Max/MSP. A graphical editor for creating *conga* graphs would make our framework more approachable to a wider range of potential developers.

8. CONCLUSIONS

We presented *conga*, an analysis framework for conducting gestures. *conga* distinguishes itself from current approaches to conducting gesture recognition in that it uses a feature detector approach, which allows the user's data to be fitted to multiple gesture profiles. These gesture profiles represent the key characteristics of a particular beating pattern. The advantage of this approach is that *conga* does not need to be trained with sample data sets, nor does it require users to conduct in specific patterns. As long as their movements trigger the features of a particular profile, the precision by which they are executed is unimportant. A profile selector decides which profile best matches the user's movements, in order to maximize the user's communication bandwidth with the virtual orchestra. We showed our design for three gesture profiles of varying quality and difficulty for the user: four-beat neutral-legato, up-down, and wiggle. Our preliminary evaluation of *conga* showed that it has a remarkably high beat recognition rate, although the latency can be quite high, making our current implementation a little premature for professional use.

Nonetheless, *conga* is both a significant improvement over our previous work, and a novel approach to a well-studied problem, and we hope to continue its development to further advance conducting as an interface to computer music.

9. ACKNOWLEDGEMENTS

The authors would like to thank Teresa Marrin Nakra for early pointers and advice on this project, and Saskia Dedenbach for her assistance with the preliminary user evaluation of *conga*. Work on *conga* was sponsored by the Betty Brinn Children's Museum in Milwaukee, USA as part of the *Maestro!* interactive conducting exhibit.

10. REFERENCES

[1] J. Borchers, E. Lee, W. Samminger, and M. Mühlhäuser. Personal Orchestra: A real-time audio/video system for interactive conducting. *ACM Multimedia Systems Journal Special Issue on Multimedia Software Engineering*, 9(5):458–465, March 2004. Errata published in ACM Multimedia Systems Journal 9(6):594.

[2] D. Buchla. Lightning II MIDI controller. http://www.buchla.com/.

[3] A. Camurri, B. Mazzarino, and G. Volpe. Analysis of expressive gesture: The EyesWeb expressive gesture processing library. In *Gesture Workshop 2003*, volume 2915 of *Lecture Notes in Computer Science*, Genova, 2003. Springer.

[4] I. Grüll. conga: A conducting gesture analysis framework. Diploma Thesis, University of Ulm, April 2005.

[5] T. Ilmonen and T. Takala. Conductor following with artificial neural networks. In *Proceedings of the ICMC 1999 International Computer Music Conference*, pages 367–370, Beijing, October 1999. ICMA.

[6] P. Kolesnik. Conducting gesture recognition, analysis and performance system. Master's thesis, McGill University, June 2004.

[7] E. Lee, T. Karrer, and J. Borchers. Toward a framework for interactive systems to conduct digital audio and video streams. *Computer Music Journal*, 30(1):21–36, 2006.

[8] E. Lee, H. Kiel, S. Dedenbach, I. Grüll, T. Karrer, M. Wolf, and J. Borchers. iSymphony: An adaptive interactive orchestral conducting system for conducting digital audio and video streams. In *Extended Abstracts of the CHI 2006 Conference on Human Factors in Computing Systems*, Montréal, Canada, April 2006. ACM Press.

[9] E. Lee, T. Marrin Nakra, and J. Borchers. You're the conductor: A realistic interactive conducting system for children. In *Proceedings of the NIME 2004 Conference on New Interfaces for Musical Expression*, pages 68–73, Hamamatsu, Japan, June 2004.

[10] E. Lee, M. Wolf, and J. Borchers. Improving orchestral conducting systems in public spaces: examining the temporal characteristics and conceptual models of conducting gestures. In *Proceedings of the CHI 2005 Conference on Human Factors in Computing Systems*, pages 731–740, Portland, USA, April 2005. ACM Press.

[11] T. Marrin Nakra. *Inside the Conductor's Jacket: Analysis, interpretation and musical synthesis of expressive gesture*. PhD thesis, Massachusetts Institute of Technology, 2000.

[12] M. V. Mathews. *Current Directions in Computer Music Research*, chapter The Conductor Program and Mechanical Baton, pages 263–282. MIT Press, Cambridge, 1991.

[13] D. Murphy. Live interpretation of conductors' beat patterns. In *13th Danish Conference on Pattern Recognition and Image Analysis*, pages 111–120, Copenhagen, 2004.

[14] D. Murphy, T. H. Andersen, and K. Jensen. Conducting audio files via computer vision. In *Gesture Workshop 2003*, volume 2915 of *Lecture Notes in Computer Science*, pages 529–540, Genova, 2003. Springer.

[15] National Instruments. LabVIEW. http://www.ni.com/labview/.

[16] M. Puckette. Max at seventeen. *Computer Music Journal*, 26(4):31–43, 2002.

[17] M. Rudolf. *The Grammar of Conducting: A Comprehensive Guide to Baton Technique and Interpretation*. Schirmer Books, 3rd edition, June 1995.

[18] S. Usa and Y. Mochida. A multi-modal conducting simulator. In *Proceedings of the ICMC 1998 International Computer Music Conference*, pages 25–32. ICMA, 1998.

Real-time CALM Synthesizer New Approaches in Hands-Controlled Voice Synthesis

N. D'Alessandro
TCTS Lab (FRIA Researcher)
Faculté Polytechnique de Mons
B-7000 Mons, Belgium
nicolas.dalessandro@fpms.ac.be

C. d'Alessandro, S. Le Beux, B. Doval
LIMSI - CNRS
Université Paris Sud XI
F-91403 Orsay, France
{cda, slebeux, boris.doval}@limsi.fr

ABSTRACT

In this paper, a new voice source model for real-time gesture–controlled voice synthesis is described. The synthesizer is based on a causal-anticausal model of the voice source, a new approach giving accurate control of voice source dimensions like tenseness and effort. Aperiodic components are also considered, resulting in an elaborate model suitable not only for lyrical singing but also for various musical styles playing with voice qualities. The model is also tested using different gestural control interfaces : data glove, keyboard, graphic tablet, pedal board. Depending on parameter-to-interface mappings, several instruments with different musical abilities are designed, taking advantage of the highly expressive possibilities of the synthesis model.

Keywords

Singing synthesis, voice source, voice quality, spectral model, formant synthesis, instrument, gestural control.

1. INTRODUCTION

Remarkable achievements have been recently reached in singing voice synthesis. A review of state of the art can be found in [1]. Technology seems mature enough for replacing vocals by synthetic singing, at least for backing vocals [2] [3]. However, existing singing synthesis systems suffer from two restrictions : they are aiming at mimicking singers rather than creating new instruments, and are generally limited to MIDI controllers.

We think it worthwhile to extend vocal possibilities of voice synthesizers and design new interfaces that will open new musical possibilities. On the one hand, a voice synthesizer should be able to reproduce several voice quality dimensions, resulting in a wide variety of sounds (e.g. quasi-sinusoidal voice, mixed periodic aperiodic voice, pressed voice, various degrees of vocal effort, etc.). On the other hand, vocal instrument being embodied in the singer, multidimensional control strategies should be devised for externalizing gestural controls of the instrument.

In this paper, a new elaborate voice source model able to produce various voice qualities is proposed. It is based on spectral modelling of voice source [4]. Links between spectral parameters and auditory effects are relatively straightforward. Then playing instruments based on spectral modelling seems very intuitive. Another key point is gesture-to-parameter mapping. Following the pioneering work by Fels [5], we found data glove particularly well suited to vocal expression. Recent work on hand-controlled vocal synthesis include series of instruments presented by Cook [6] and the Voicer by Kessous [7]. It must be pointed out that musical possibilities offered by an instrument strongly depend on mapping and interfaces. Then, depending on intended musical aims, different instruments are proposed. This paper is organized as follows. In section 2, the voice synthesis model is reviewed. In section 3, control devices and mapping of voice quality dimensions onto control parameters are discussed. Section 4 presents two musical instruments built on basis of synthesis model and vocal dimensions. Section 5 presents a discussion of results obtained and proposes directions for future works.

2. VOICE SYNTHESIS MODEL

In this section, we first give an overview of mechanisms involved in voice production. Then, we focus on the glottal source and present the causal-anticausal linear model developed by d'Alessandro/Doval/Henrich in [4]. We also explain the nature of non-periodical components we introduced in the model. Finally, we describe structure and possibilities of the real-time glottal flow synthesizer based on CALM (RT-CALM) we developed and integrated in following singing instruments.

2.1 Voice production

Voice organ is usually described as a "source/filter" system. Glottal source is a non-linear volume velocity generator where sound is produced by complex movements of vocal folds (larynx) under lungs pressure. A complete study of glottal source can be found in [8]. Sounds produced by the larynx are then propagated in oral and nasal cavities which can be seen as time-varying filtering. Finally, the volume velocity flow is converted into radiated pressure waves through lips and nose openings (cf. Figure 1).

In the context of signal processing applications, and particulary in singing synthesis, some simplifications are usually accepted. First, lips and nose openings effect can be seen as derivative of the volume velocity flow. It is generally processed by a time-invariant high-pass first order linear filter [9]. Vocal tract effect can be modelized by filtering of glottal signal with multiple (4 or 5) second order resonant linear filters (formants).

Contrary to this "standard" vocal tract implementation, plenty of models have been developed for represen-

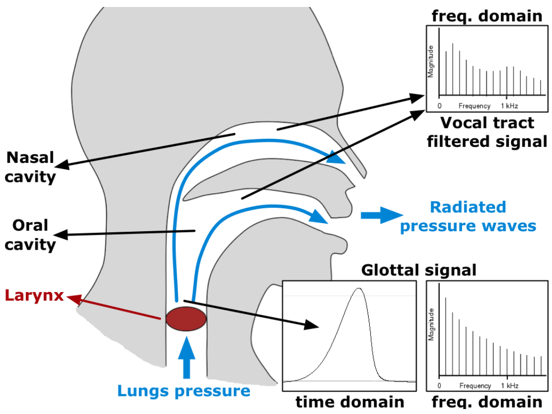


Figure 1: Voice production mechanisms : vocal folds vibrations, vocal tract filtering and lips/nose openings radiation.

tation of glottal flow, with differences in accuracy and flexibility. Usual models are KLGLOTT88 [10], R++ [11], Rosenberg-C [12] and LF [13] [14]. We present now the causal-anticausal linear model (CALM) [4], explain why we worked with this spectral approach and propose adaptations of the existing algorithm to ease real-time manipulation.

2.2 CALM : causal-anticausal linear model

We have seen that modelling vocal tract in spectral domain (with resonant filters central frequency, amplitude and bandwidth) is very powerful in term of manipulation because spectral description of sounds is close to auditory perception. Traditionally, glottal flow has been modelized in time domain. A spectral approach can be seen as equivalent only if both amplitude and phase spectra are considered in the model.

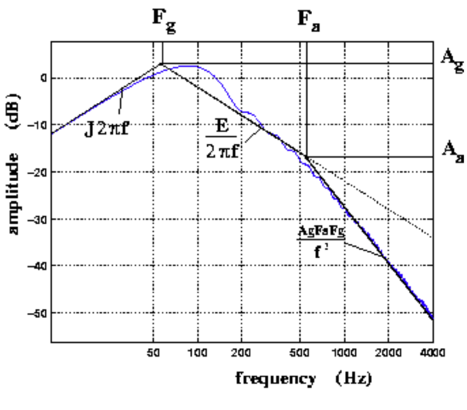


Figure 2: Amplitude spectrum of the glottal flow derivative : illustration of glottal formant (F_g, A_g) and spectral tilt (F_a, A_a).

For amplitude spectrum, two different effects can be isolated (cf. Figure 2). On the one hand, an amount of energy is concentrated in low frequencies (i.e. below 3 kHz). This peak is usually called "glottal formant". We can see that bandwidth, amplitude and position of the glottal formant change with voice quality variations. On the other hand, a variation of spectrum slope in higher frequencies (called "spectral tilt") is also related to voice quality modifications.

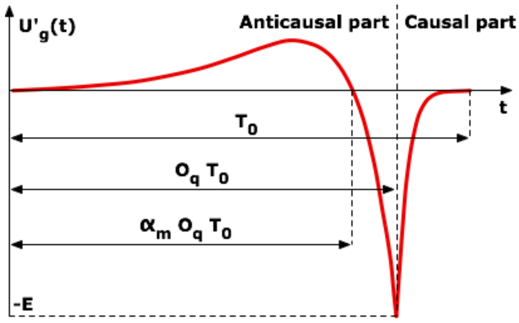


Figure 3: Time-domain representation of derivated glottal pulse : anticausal part and causal part.

Considering both "glottal formant" and "spectral tilt" effects, two cascading filters are implemented. A second order resonant low-pass filter (H_1) for glottal formant, and a first order low-pass filter (H_2) for spectral tilt. But phase information indicates us that this system is not completely causal. Indeed, as it is illustrated on Figure 3, glottal pulse is a combination of a "increasing" (or active) part and a "decreasing" (or passive) part. The decreasing part, called the return phase, mainly influences the spectral tilt and hence is causal. And we can also show that the second order low-pass filter has to be anticausal in order to provide a good phase representation.

A complete study of spectral features of glottal flow, detailed in [4], gives us equations linking relevant parameters of glottal pulse (F_0 : fundamental frequency, O_q : open quotient, α_m : asymetry coefficient and T_l : spectral tilt, in dB at 3000Hz) to H_1 and H_2 coefficients. Note that expression of b_1 has been corrected. [4] also contains equations linking this time-domain parameters with spectral-domain parameters.

Anticausal second order resonant filter :

$$H_1(z) = \frac{b_1 z}{1 + a_1 z + a_2 z^2}$$

where :

$$a_1 = -2e^{-a_p T_e} cos(b_p T_e),\ a_2 = e^{-2a_p T_e}$$

$$b_1 = \frac{E}{b_p} e^{-a_p T_e} sin(b_p T_e)$$

$$a_p = -\frac{\pi}{O_q T_0 tan(\pi \alpha_m)},\ b_p = \frac{\pi}{O_q T_0}$$

Causal first order filter :

$$H_2(z) = \frac{b_{T_L}}{1 - a_{T_L} z^{-1}}$$

where :

$$a_{T_L} = \nu - \sqrt{\nu^2 - 1},\ b_{T_L} = 1 - a_{T_L}$$

$$\nu = 1 - \frac{1}{\eta},\ \eta = \frac{\frac{1}{e^{-T_L/10ln(10)}} - 1}{cos(2\pi \frac{3000}{F_e}) - 1}$$

2.3 Non-periodical components

As described theoretically in [4], the glottal flow is a deterministic signal, completely driven by a set of parameters. Adding naturalness involves the use of some random components we propose to describe.

Jitter

Jitter is a natural unstability in the value of fundamental frequency. It can be modelized by a random value (gaussian distribution, around 0 with variance depending on the amount of jitter introduced), refreshed every period, added to the stable value of fundamental frequency.

Shimmer
Shimmer is a natural unstability in the value of the amplitude. It can be modelized by a random value (gaussian distribution, around 0 with variance depending on the amount of shimmer introduced), refreshed every period, added to the stable value of amplitude.

Turbulences
Turbulences are caused by additive air passing through vocal folds when glottal closure is not complete. It can be modelized by pink noise filtered by a large band-pass (tube noise), modulated in amplitude by glottal pulses.

We can note here that we kept a direct control on irregularities (based on *Jitter*, *Shimmer* and *Turbulences* rates). Other models were developed, involving granular synthesis coupled with self-organizing dynamic systems [15], and could be considered in further works.

2.4 RT-CALM framework

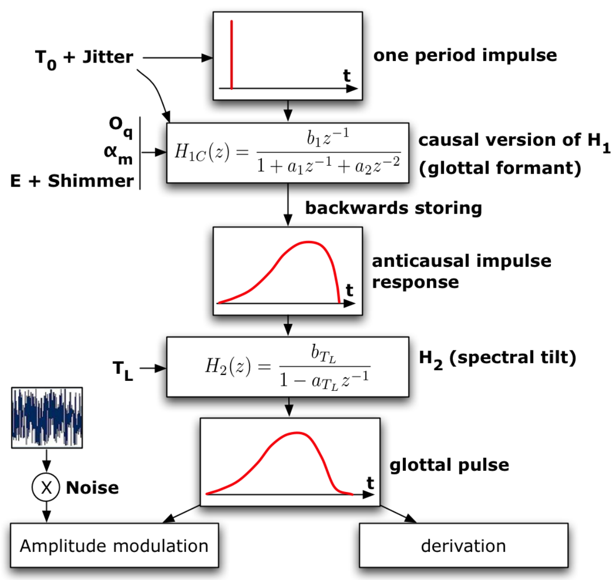


Figure 4: Framework of RT-CALM algorithm, allowing real-time synthesis of glottal pulses based on causal-anticausal linear model.

Full anticausal processing is only possible offline, by running algorithms backwards on data buffers. Anyway, in this context, we can take advantage of physical properties of glottis to propose a real-time algorithm. Indeed, glottal pulse corresponds to opening/closing movements of vocal folds. It means that impulse responses generated by H_1 and H_2 filters can't overlap. Thus, if ranges of parameters are correctly limited, impulse responses can be stored backwards and truncated period-synchronously without changing too much their spectral properties.

To achieve the requested waveform, impulse response of causal version of H_1 (glottal formant) is computed, but stored backwards in the buffer. This waveform is truncated at a length corresponding to instantaneous fundamental frequency (F_0+*Jitter*). Then the resulting period is filtered by H_2 (spectral tilt). Coefficients of H_1 and H_2 are calculated from equations described in subsection 2.2 and [4]. Thus, both time-domain and spectral-domain parameters can be sent. On the one hand, glottal pulses are derivated to produce pressure signal (cf. Figure 3). On the other hand, it is used to modulate the amount of additive noise. Complete RT-CALM algorithm is illustrated at Figure 4.

3. VOICE QUALITY DIMENSIONS

Voice synthesis model is driven by a set of low-level parameters. In order to use these parameters in singing, they must be organized according to musical dimensions. Mappings between parameters and dimensions, and between dimensions and controllers are essential parts of instrument design. In this section, we describe main musical dimensions for voice source (cf. Figure 5) and vocal tract (cf. Figure 6).

3.1 Glottal source

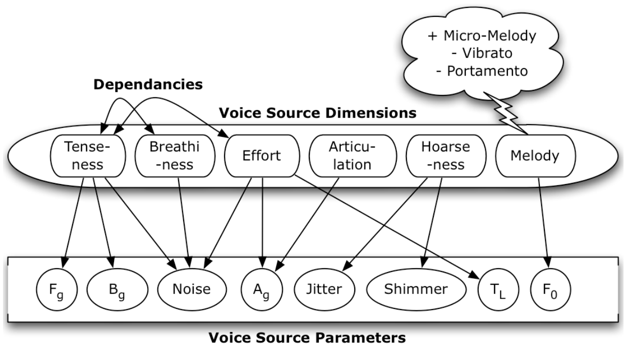


Figure 5: Mapping of the vocal source

Melodic dimension
For singing, this dimension can be decomposed into two parts. On the one hand, it seems important to sing in tune i.e. to make use of notes with well-defined pitches. On the other hand, micro-melodic variations are essential for expressive and natural singing (portamento, vibrato, etc.). Two different controls seem necessary for melodic dimension. This dimension mainly depends on parameter F_0. Anyway, a more precise vibrato synthesis should also involve amplitude variations.

Hoarseness dimension
This dimension is linked to structural aperiodicities in voice source, like *Jitter* and *Shimmer*.

Breathiness dimension
This dimension is linked to aspiration noise in voice source. It controls the relative amount of voicing vs. whispering, using the *Noise* parameter.

Pressed/lax dimension
This dimension is mainly linked to the position of the glottal formant F_g and its bandwidth B_g. It is often linked to breathiness and vocal effort. The pressed/lax dimension is used in some styles of singing e.g. Japanese noh theater or belt singing.

Vocal effort dimension
This dimension is linked to spectral tilt T_l and of course to gain parameter A_g.

3.2 Vocal tract

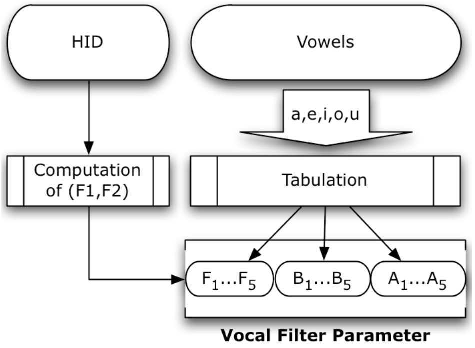


Figure 6: Mapping of the vocal tract

Vocalic space
This space is defining vocal genre (male/female/child), phonemes, and other expressive features (lips rounding, lips spreading, tongue position). This space can also be used for harmonic singing. The vocalic space is defined by formant parameters F_1, B_1, A_1, F_2, B_2, A_2,... F_N, B_N, A_N.

Articulation dimension
Finally, notes attacks and decays are controlled by an articulation dimension. "Articulation" is taken here in its musical meaning i.e. transitions between notes. It is essentially controlled by gain parameter A_g.

3.3 Musical control of vocal dimensions

Playing with melody
Melodic playing usually requires precise pitches. Then "selection" gestures are needed using e.g. a keyboard. However, natural vocal note transitions are generally slow, with more or less portamento and vibrato. Small and controlled pitch variations are therefore needed, and the "selection" gesture must be accompanied by a "modification" gesture, using e.g. hand position in one dimension of space. Another elegant solution offering accurate pitch control and smooth micro-melodic variation is using a graphic tablet. A virtual guitar board can be emulated this way. Well tuned pitches are not required in some singing styles imitating speech, like Sprechgesang (parlar cantando). Then only one control gesture is needed, that can be achieved by position of hand in one spatial dimension.

Playing with timbre : vocalic space
Playing with vocalic timbre is often used on slow moving melodies e.g. harmonic singing. The basic vocalic space needs two dimensions for contrasting vowels e.g. a joystick or a graphic tablet. One dimension is sufficient for harmonic singing (moving only second formant frequency), using a slider or position of hand in one spatial dimension. But a third dimension would be needed for signaling facial movements like lips spreading or rounding, using e.g. a data glove.

Playing with timbre : noise and tension
Some musical styles are also playing with noise and tension. These parameters are moving relatively slowly, on a limited scale, and gestures must not be extremely precise. They can be naturally associated to flexion of fingers in a data glove.

Playing with articulation and phrasing The data glove proved also useful for articulation (in the musical meaning of note attack and release) and phrasing. Hand movements in space are well suited to phrasing and finger flexions are well suited to articulation.

4. CALM-BASED INSTRUMENTS

This section describes two setups we realised. Main purpose of this work was to realize extensive real-time tests of our CALM synthesis model and voice quality dimensions mappings. No dedicated controllers were designed for this purpose. Only usual devices such as tablets, joyticks or keyboards were used.

4.1 Instrument 1

In this first instrument implementation, we use a keyboard to play MIDI notes in order to trigger the vowels at different tuned pitches. Thus, by using keyboard, we are able to set glove free for fine tuning of F_0 so as to achieve vibrato, portamento of other types of melodic ornaments. Accurate control of F_0 by glove position alone proved difficult because well tuned notes references were missing, due to approximative nature of hand gestures.

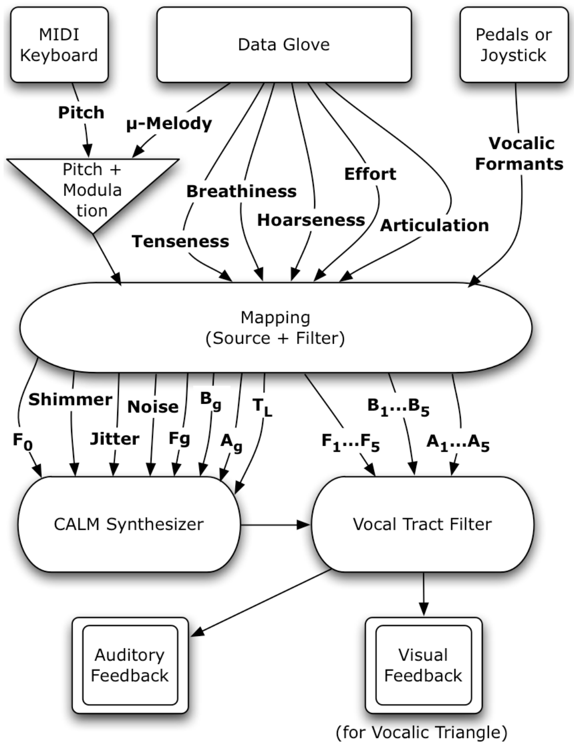


Figure 7: Structure of Instrument 1

Tempered notes (or other conventions) delivered by keyboard can be modified to a certain extent, thanks to tracking of glove position along a certain axis (transversal axis gives better ergonomics as one don't have to fold the elbow to achieve vibrato). General gain is mapped onto longitudinal axis of the glove. Then both vibrato and amplitude envelopes of sound can be produced by circular hand movements. Other vocal dimensions are controlled by flexion of data glove fingers. First finger controls vocal effort (spectral tilt), second finger controls breathiness (linked to additive noise), third finger control the pressed/lax dimension (linked to the glottal formant), fourth finger controls hoarseness (linked to jitter and shimmer). Voice quality modifications are achieved by closing/opening movements of whole hand or selected fingers. Preset vowels are associated to keys of computer keyboard. Vowel formants can also be modified by additional devices, like pedal board or joysticks.

In summary, for this first instrument :

1. left-hand controls the keyboard (tempered notes)
2. right-hand movements control both fine pitch modulation, and note phrasing.
3. right-hand fingers control tension, effort, hoarseness, and breathiness.

In this implementation, note phrasing results of relatively large hand movements. An alternative solution is to couple effort and note phrasing in fingers movements, and to keep one dimension of hand movement for controlling another vocal dimension (e.g. breathiness). Then, phrasing is controlled by smaller and quicker finger movements. Overall description of this instrument and its various components is illustrated on Figure 7.

4.2 Instrument 2

The key point of this second instrument is simplicity of learning and using. Different choices have been made to achieve that result. First, we decided to focus on voice quality. Vocal tract control would be limited to vowel switching. Then, we took advantage of our natural writing abilities to map all glottal flow features only on tree dimensions of a graphic tablet (x axis, y axis and pressure).

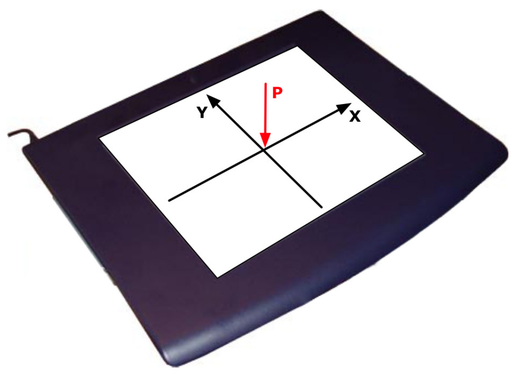


Figure 8: Mapping on the graphic tablet. X axis : fundamental frequency, Y axis : pressed/lax and vocal effort dimensions, Pressure (P) : general volume.

As described on Figure 8, horizontal axis is mapped to fundamental frequency. Tests have been made showing that, after a few training, 2 or 3 (even 4) octaves can be managed on a *Wacom Graphire* tablet. Anyway, transposition and surface scaling features have been implemented. Vertical axis control both pressed/lax and vocal effort dimensions. Mapping is made by using Y value as an interpolation factor between two different configurations of parameters O_q, α_m and T_L, from a "quiet" voice to a "tensed" voice (cf. Figure 9). Finally, pressure parameter is mapped to the gain (E).

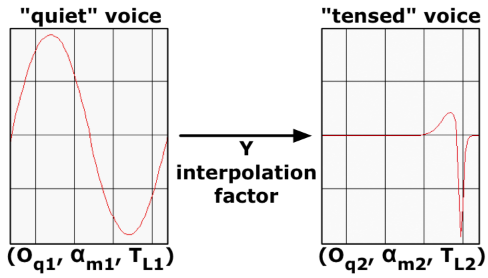


Figure 9: Interpolation between "quiet" voice and "tensed" voice made by Y axis of the graphic tablet.

Regression of voice quality control on an overall expressive axis makes main manipulations of voice source possible with simple "drawings" (i.e. bidimensional + pressure shapes). This compromise makes this instrument really intuitive. Indeed, as it can be done e.g. with a guitar, interpreter only needs graphic tablet to play. MIDI controller (e.g. pedal board) is just used for changing presets (cf. Figure 10).

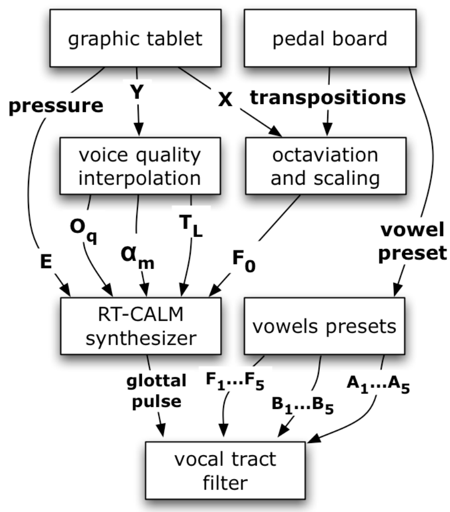


Figure 10: Structure of Instrument 2.

5. CONCLUSIONS AND FUTURE WORK

The two instruments implemented so far are suitable for musical use. Instruments have a truly human sound, and new possibilities offered by gestures to sound mapping enable intuitive playing. Compared to other voice synthesis systems, more emphasis is put on voice quality controls. It is then possible to play with expressive musical dimensions inherent to wind instruments, like effort, pressure and noise. These dimensions are exploited in acoustic instruments like saxophones and brass, and of course voice, but are generally ignored in singing synthesis. Hand movements in space and hand/fingers closures/openings are intuitively associated to such dimensions as effort or voice pressure.

Another challenging point for singing synthesis is accurate yet flexible F_0 control, like in fretless string instruments. This has been implemented in two ways in our instruments (graphic tablet and glove controlled F_0). This flexible F_0 control enables the player all possible types of intonation, from singing to speech. Melodic ornaments like e.g. vibrato or portamento are easily controlled.

Spectral processing of voice quality proved also useful for "spectral" singing styles. Overtone singing, formant melodies, various types of throat singing are easily produced and controlled in this framework.

Instruments can also be considered as tools for studying singing, because they produce very natural sounding and controlled signals. Then they can be used for investigating musical gestures involved in singing.

Apart from the two instruments presented here, we are also investigating other types of data gloves and elaborated 3D joysticks for refining control of the synthesizer. However, this will not change the nature and number of useful vocal dimensions, but improve precision and ergonomics.

Of course, singing is an instrument that mixes together music and language. Thus, our next challenge is to control the "speech" part of singing. This point has been only marginally considered in the present research and will be the object of future work. Addition of speech articulations would drive us to more accurate modelization of vocal tract, eventually based on existing databases. Considering interfaces, syntaxic abilities of controllers have to be determined in order to achieve syllables, words or sentences synthesis.

6. ACKNOWLEDGMENTS

This work originated from the Speech Conductor project, a part of the eNTERFACE'05 workshop organized by Prof. Thierry Dutoit (Faculté Polytechnique de Mons) within the SIMILAR Network of Excellence (European Union - FP6). We would like to thank all these institutions that provided excellent working conditions.

7. REFERENCES

[1] M. Kob, "Singing Voice Modelling As We Know It Today," *Acta Acustica United with Acustica*, Vol. 90, pp. 649–661, 2004.

[2] Virsyn Corporation, "The Cantor Singing Synthesis Software," 2005-present, url : http ://www.virsyn.de/

[3] Yamaha Corporation, "The Vocaloid Singing Synthesis Software," 2003-present, url : http ://www.vocaloid.com/

[4] B. Doval, C. d'Alessandro and N. Henrich, "The Voice Source as an Causal-Anticausal Linear Filter," *Proc. ISCA ITRW VOQUAL'03*, Geneva, Switzerland, August 2003, pp. 15–19.

[5] S. Fels and G. Hinton, "Glove-TalkII : A Neural Network Interface which Maps Gestures to Parallel Formant Speech Synthesizer Controls," *IEEE Transactions on Neural Networks*, Vol 9, No. 1, pp. 205–212, 1998.

[6] P. Cook, "Real-Time Performance Controllers for Synthesized Singing," *Proc. NIME 2005*, Vancouver, Canada, May 2005, pp. 236–237

[7] L. Kessous, "Gestural Control of Singing Voice, a Musical Instrument," *Proc. of Sound and Music computing 2004*, Paris, October 20-22, 2004.

[8] N. Henrich. *"Etude de la source glottique en voix parlée et chantée."* Thèse de doctorat à l'Université Paris VI, 2001.

[9] G. Fant. *"Acoustic Theory of Speech Production"*, Gravenhage, 1960.

[10] D. Klatt and L. Klatt, "Analysis, Synthesis, and Perception of Voice Quality Variations among Female and Male Talkers," *J. Acoust. Soc. Am.*, 87(2) :820–857, 1990.

[11] R. Veldhuis, "A Computationally Efficient Alternative for the Liljencrants-Fant Model and its Perceptual Evaluation," *J. Acoust. Soc. Am.*, 103 :566–571, 1998.

[12] A. E. Rosenberg, "Effect of Glottal Pulse Shape on the Quality of Natural Vowels," *J. Acoust. Soc. Am.*, 49 :583–590, 1971.

[13] G. Fant, J. Liljencrants, and Q. Lin, "A Four-Parameter Model of Glottal Flow," *STL–QPSR*, 85(2) :1–13, 1985.

[14] G. Fant, "The LF-Model Revisited. Transformations and Frequency Domain Analysis," *STL–QPSR*, 2–3, 119–56, 1995.

[15] E. R. Miranda, "Generating Source-Streams for Extralinguistic Utterances". *Journal of the Audio Engineering Society (AES)*, Vol. 50 N°3, March 2002.

GRASSP: Gesturally-Realized Audio, Speech and Song Performance

Bob Pritchard
School of Music
University of British Columbia
Vancouver, B.C. Canada
+1.604.822.3526
bob@interchange.ubc.ca

Sidney Fels
Dept. of Electrical Engineering
University of British Columbia
Vancouver, B.C. Canada
+1.604.822.5338
ssfels@ece.ubc.ca

ABSTRACT
We describe the implementation of an environment for Gesturally-Realized Audio, Speech and Song Performance (GRASSP), which includes a glove-based interface, a mapping/training interface, and a collection of Max/MSP/Jitter bpatchers that allow the user to improvise speech, song, sound synthesis, sound processing, sound localization, and video processing. The mapping/training interface provides a framework for performers to specify by example the mapping between gesture and sound or video controls. We demonstrate the effectiveness of the GRASSP environment for gestural control of musical expression by creating a gesture-to-voice system that is currently being used by performers.

Keywords
Speech synthesis, parallel formant speech synthesizer, gesture control, Max/MSP, Jitter, Cyberglove, Polhemus, sound diffusion, UBC Toolbox, Glove-Talk,

1. INTRODUCTION
The Gesturally-Realized Audio, Speech and Song Performance (GRASSP) environment is designed to synthesize speech and sound, and assist in real time processing of audio and video from real-time performer control through a mapping interface. For input, we created various input objects supporting a Cyberglove™, Polhemus Fastrak™ tracker, a custom-built left hand glove, and a footswitch, allowing the performer to accomplish all of this within a Max/MSP/Jitter environment.

The use of modified gloves, meta-gloves, and glove-like mechanisms for the control of synthesis and processing is not uncommon, as shown by such work as Waisvisz [1] and Sonami [2]. Indeed, gestural control is now common enough to warrant publications [3] dealing with the associated problems and solutions. Of particular interest in our project is the creation of a mapping and training component that quickly allows the user to provide examples for adapting the mapping as well as for learning the hand positions required for speech synthesis. This has implications for the expanded development and use of sophisticated controllers in the GRASSP environment.

2. CONTEXT
Speech synthesis is found across a wide spectrum of uses, ranging from children's games to sophisticated telecommunication applications. In general the implementations that are found in text-to-speech synthesis use either concatenative synthesis such as Festival[4], or else some form of text-to-speech converters such as DecTalk™ to drive the speech synthesizer. Text-to-speech is a powerful and successful method but it lacks the ability to improvise speech in real time, and it can be difficult to implement a variety of inflections for expressive purposes.

Glove-TalkII [5] is one example of a system that allows users to improvise speech and inflection, allowing for more natural sounding conversation with the user. Using a Cyberglove™, a Polhemus Fastrak™, a custom-made left hand glove, and a foot controller, the user is able to control the formants and noise components of speech through different hand and finger positions mapped to control parameters of a parallel formant speech synthesizer. Along similar lines, Cook [6] has created a vocal synthesizer that he controls with various bellows-based instruments including a concertina and an accordion.

One of the key features of Glove-TalkII is the ability of the user to provide examples to the system of hand gestures for different cardinal sounds. These examples were used to train the mapping between gesture and speech synthesis parameters using a neural network. We were motivated by this approach to create a similar, but expressively expanded environment for performers to control speech, sound and video parameters. In order to do this, we explicitly consider gestural control of sound and video parameters as a trainable mapping between inputs and outputs.

In Glove-TalkII, the training paradigm was done separately from the process of controlling speech. However, within Max/MSP, we are able to unify the objects that provide inputs and the objects that provide expressive output control through a generalized mapping interface that we call a *Trainer*. As Orio et al [7] point out, the mapping between inputs and outputs needs to be the focus of attention for understanding expressive control. Furthermore, Glove-TalkII illustrated that adaptive mappings can provide an effective way for a user to be able to train a mapping to suit their taste. Glove-TalkII started with an initial mapping that specified a reasonable starting point for producing speech from gesture based on a *hand-as-articulator* metaphor. The user's interpretation of that mapping determined the actual training data. In keeping with this approach, our training/mapping interface provides a mechanism to display the current mapping while also allowing the addition of new examples from the performer to adjust the mapping. Our current adaptation adjusts the centres of radial basis functions [8] using single examples. However, it is a simple matter to extend our interface to include other machine learning techniques such as found in Glove-TalkII.

3. THE ENVIRONMENT

The GRASSP environment can be viewed as having three distinct yet interrelated components. These three parts and their roles are summarized as follows:

The Controllers: these consist of a Cyberglove™, a Polhemus Fastrak™, a custom left-hand glove, and a footswitch. This collection allows the user to generate control signals.

The Trainer: this consists of a Dictionary, a Recorder, Tracker, and a Mapper. These four Max/MSP/Jitter interfaces a) provide the user with models of hand positions associated with specific speech sounds, b) record the control data resulting from the user's attempts to imitate the position, c) show the user the approximate relation of the hand location to the target positions and d) provide the relation between the Controllers and the UBC Toolbox. The Mapper also contains the adaptation algorithm that uses the data from the Recorder.

The UBC Toolbox: this is a collection of Max/MSP/Jitter bpatchers, created by Hamel and Pritchard [9]. Included in the UBC Toolbox is the parallel speech synthesis engine of Glove-TalkII that has been recoded in Max/MSP and implemented as a bpatcher. We included this object so that performers can sing or speak using gestures.

Table 1: UBC Toolbox bpatcher examples

Category	Name/Description
Players	boxcar, filePlayer, fmPlayer, granulator, ksPlayer, pafPlayer, sampler, sfPlayer, vibePlayer, vSynthDB, vSynthDBFem,
Effects	chorus, combFilter, crossSynth, delay, multifilter, flange, fShifter, harmonizer, reverb, ringMod, vocoder
I/O	multipan4, multipan8, NAInterface, netSender, netReceiver, pan4, pitchTracker, recorder
Controls	cMatrix, control collections, dispatch, matrix, mixer, randGen
Jitter	chromaKey, crossfade, floatWindow, mathOps, messenger, motionDetector, videoGrabber, videoPlayer, writeToDisk

4. RELATED WORK

Fels and Hinton implemented the Glove-TalkII system based on the parallel-formant speech synthesizer of Rye and Holmes [10] which was developed at the Joint Speech Research Unit (JSRU) in the UK. Fels and Hinton created a gesture-controlled interface for their version of the JSRU synthesizer, and added an adaptive interface component to refine the user training and speech synthesis. The resulting system converts hand gestures to speech, using a gesture-to-formant model. The right hand location in 2D horizontal space controls the creation of vowel formants by mapping hand position to the amplitudes of the first two formants. Specific hand locations are identified as targets for specific cardinal vowels and their related formants. Normalized radial basis functions [8] provide a vowel landscape that produces cardinal vowel sounds as the hand approaches the target position and maintains it as the hand extends to extreme positions outside the main vowel space. The vertical position of the right hand controls pitch, and the right hand finger positions control the creation of all other speech sounds with the exception of eight stopped consonants. These eight consonants are triggered by contact switches on a left hand glove, and the overall amplitude of the system is controlled by a foot switch.

Hamel and Pritchard created the UBC Toolbox as a way of providing easily useable yet sophisticated modules. The collection contains over forty bpatchers, and all of the audio related ones use the same data and messaging protocol, and can be chained together or controlled via a bpatcher matrix.

5. PROJECT DESCRIPTION

5.1 Project Goals

The main goals of our project included the implementation of the JSRU synthesizer in the Max/MSP environment, the expansion of the speech synthesizer's capabilities to include the control of sound synthesis, sound processing, sound localization, and video processing, and the creation of a training system to ease the task of learning to synthesize speech. The Max/MSP/Jitter environment running on OS X was attractive for us since we have a great deal of experience with the environment, and almost all of the software development in the School of Music (synthesis methods, notation systems, teaching materials) is executed in OS X.

Secondary goals include using the gesture data to control kinetic sculptures and lighting systems, creating a direct correlation between gesture, sound, motion, and illumination. An additional goal is to develop a fully portable version of the environment by making the system self-contained.

5.2 Implementation

As our first demonstration of the utility of GRASSP, we have implemented a gesture-to-voice system based on Glove-TalkII. We use this example to illustrate the different components of GRASSP.

We divide the task of gesture-to-voice control into three areas: vowels, consonants, and stopped plosives. Vowels are considered to be those sounds made with the open mouth. Consonants are utterances created by constraining or restricting the air flow through the vocal passage, and stopped plosives are consonants in which the air flow is interrupted by the lips, tongue, teeth, palette, or some combination of those. In this example for Controllers we use data gloves, a tracking sensor, and a foot pedal for capturing gestures. The Trainer is set up to provide a mechanism for performers to see and add to the dictionary of gestures to vowel and consonant sounds. The Tracker allows performers to set the space of vowel control as well as to visually monitor the current location of their hand in that vowel space. The Trainer also provides the Mapper that generates the 11 control parameters of the JSRU speech synthesizer from the Controllers. Finally, we use a new version of the JSRU parallel formant synthesizer implemented in Max/MSP. This implementation forms a new member of our UBC Toolbox.

5.2.1 The Controllers

For the generation of control data we adopted the approach of Glove-TalkII, using a Cyberglove™, custom left-hand glove, a foot switch, and a Polhemus FastTrak™. It was necessary to write new Max/MSP objects to connect to the Cyberglove™ and FasTrak™, so glove_ini, glove, polhemus_ini, and polhemus objects were created. The _ini objects translate Cyberglove™ and Fastrak™ control messages and pass them on to the standard Max/MSP **serial** objects to configure the respective hardware devices. The **serial** objects access USB (and RS232 ports) and pass the incoming data to the glove or polhemus object. The glove object supplies the 18 sensor readings required for synthesis, while the polhemus object supplies the X, Y, Z, Yaw, Pitch, and Roll coordinates of up to 4 trackers. We currently use only one tracker located on the back of the right hand wrist for our gesture-to-voice implementation.

Additional control data is generated by a foot switch and by a custom-made left-hand glove. The footswitch is a simple USB model that is intended to control the overall system volume as in Glove-TalkII. The left hand glove triggers are intended for the production of stopped plosives as in Glove-TalkII. It uses a qwerty keyboard chip for data control, and has nine touch points – two on each finger and one on the thumb. When the thumb contacts a touch point it triggers a bang message that can be used to produce one of eight stop consonant routines such a making a 'B' sound.

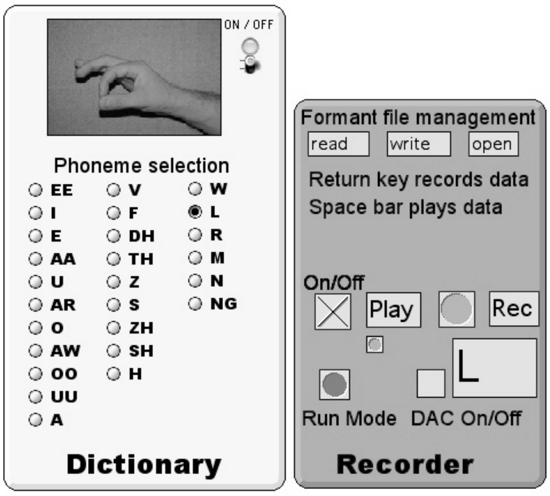


Figure 1: Dictionary and Recorder

5.2.2 *The Trainer*

The Trainer consists of a Dictionary, a Recorder, Tracker and an embedded Mapper. Figure 1 shows the Dictionary displaying the hand position for the phoneme 'L' for our implementation of a gesture-to-voice mapping. The Recorder is ready to accept training data being generated by the performer and write it into the data base for use by the Mapper. The Dictionary's illustrations are of a gloveless hand since this provides the trainee with greater detail of how to configure the hand. The Dictionary images provide a graphical view of the initial gesture vocabulary, and the images can be changed easily to accommodate different representations of the initial vocabulary. In our gesture-to-voice implementation, vowels do not require finger positions since they are determined by the horizontal location of the wrist, and the Tracker assists in creating the data necessary for their production.

The Tracker is shown in Figure 2. The Tracker displays a vowel quadrilateral based on the work of Petersen and Barney[11]. The small circle in the lower left of the quadrilateral is a cursor displaying the current position of the gloved right hand in relation to the Fastrak receiver. (The Tracker's coordinates are based on the user's frame of reference, with X increasing horizontally left-to-right, and Y increasing horizontally close-to-far.) The user is able to set the X, Y, and Z boundaries of the synthesis space by activating the appropriate button to record the location of the hand.

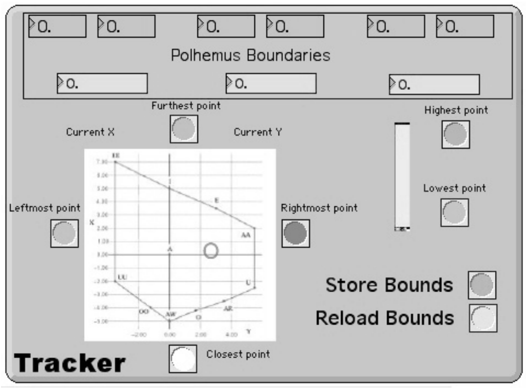


Figure 2: Tracker

Using the Trainer in our gesture-to-voice implementation, the performer sets the locations of eleven target vowels indicated in the quadrilateral. To do this, she selects a vowel from the left hand column in the Dictionary, moves her hand so that the cursor in the Tracker is over the chosen vowel, and enters that location using the Recorder. She also uses the Tracker to set her movement bounds of her vowel space. Likewise, the performer uses the Dictionary interface to store the 15 consonant gestures. The entire collection of data (consisting of the bounds of the space to be used in relation to the Fastrak receiver, the vowel locations, and the hand positions for each consonant) is known as an "accent". Each user develops and modifies his or her own accent, based on their personal preferences and their physical characteristics. Each vowel or consonant in an accent contains over 40 different floats and integers, as each contains all the sensor readings from the Cyberglove™ and FastTrak, eleven control values for the synthesis of that sound, and placeholders for future use.

The Mapper uses the accent to adjust the parameters of the relation between the Controllers and parameters of the output controls which in our case is the speech syntehsizer and other bpatchers in the UBC Toolbox. We currently adjust the centres of the normalized RBF functions using the single examples for each sound in the accent. However, different machine learning techniques may be used as well as a different training data collection scheme for this.

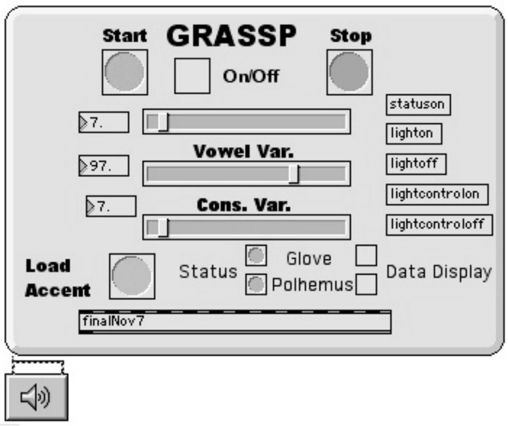


Figure 3: GRASSP bpatcher

5.2.3 *The UBC Toolbox*

As mentioned, the Toolbox currently contains over forty bpatchers. (See Table 1.) The bpatchers are capable of a variety of synthesis and processing duties in the audio and video domains, and are easily controlled by external inputs and/or and internal messaging, Our coding of the speech synthesis routines in a bpatcher allows it to be integrated seamlessly into the processing and synthesis chain of the GRASSP environment. Figure 3 shows the GRASSP bpatcher. We focus on the implementation of the speech synthesizer here as it required significant effort and its implementation is not obvious.

5.3 The Speech Synthesizer

We implemented the JSRU parallel formant speech synthesizer [10] in Max/MSP as a bpatcher in the UBC Toolkit. The synthesizer accepts eleven control signals: FN, F1, F2, F3 -- the centre frequencies and amplitudes of vowel formants; AHF, A1, A2, A3 – the amplitudes of those frequencies; IV – the degree of voicing between voiced and unvoiced sounds; and F0 – the current fundamental frequency of the sound being synthesized. Ten of the control signals are low-pass filtered to eliminate artifacts caused by fast changes before being passed to the synthesizer. This is especially important since the control parameters are coming from the Mapper and may be quite discontinuous.

Following [10], inside the synthesizer there are six areas of control and synthesis: voiced excitation, unvoiced excitation, a mixer, gain units, formant resonators, and spectrum weighting filters. The relation of these areas is shown in Figure 4.

Voiced excitation is provided by digital representations of glottal waveforms, which are played using the cycle~ object. Unvoiced excitation uses the noise~ object.

The six channel mixer blends the voiced and unvoiced excitations according to the degree of voicing indicated by the IV control signal. The resulting signal is then sent to the six different formant resonators. At the same time, the centre frequency and bandwidth are calculated for each formant based on the input parameters and the data are passed to the formant resonators.

Each formant resonator contains a biquad~ object, and that object's output is phase matched before being passed on to the spectrum filter. All six channels are then combined and passed through the spectrum weighting filters which consist of a bank of five biquad~ objects. Additional heuristic adjustments to take into account spectral balancing, glottal energy equalization and special parameter couplings according to [10] are also implemented.

Each set of 11 input parameters are updated at 100Hz. The processing is handled effectively on a MacMini with spare capacity for managing the I/O. When voiced excitation is used, the system uses prerecorded glottal waveforms to drive the resonators. Collectively, the system synthesizes speech from the formant parameters supplied to the bpatcher.

5.4 Training of Gesture-to-Voice

Unlike the original Glove-TalkII, our gesture-to-voice implementation in GRASSP does not make use of neural networks to assist in teaching and refining the system. Instead, we use a simple method of setting vowel and consonant centres using a single gesture example for each sound. It is expected that the user will adapt to the interface: in much the same way that a performer must learn to adapt to a traditional instrument, GRASSP users must learn to adapt to this new instrument.

Training begins with the user creating a complete accent by recording the eleven target positions for vowels, the fifteen hand positions for consonants, and the spatial coordinates of the bounds.

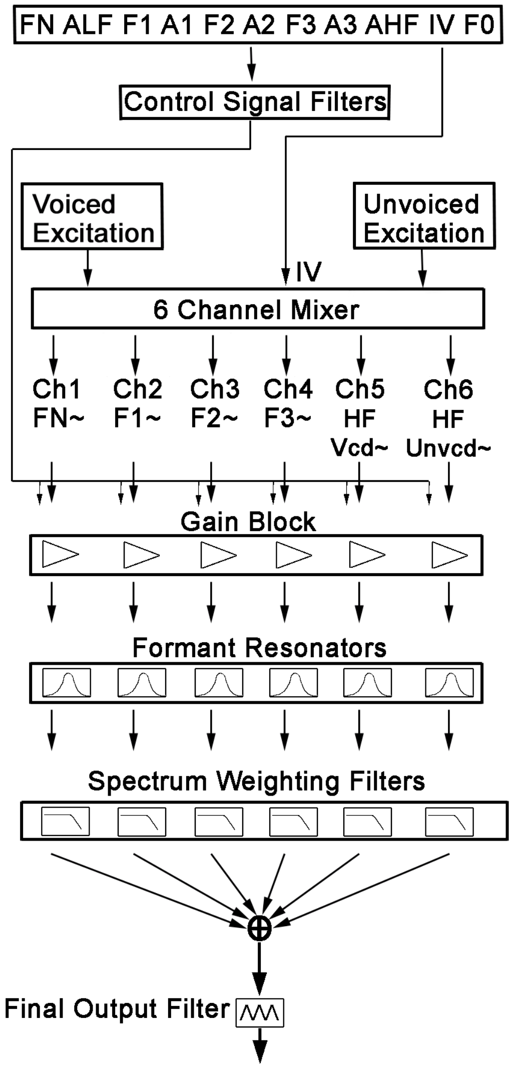


Figure 4: Block diagram of speech synthesizer

Once the accent has been established, training continues with the exploration of the vowel space, and then progresses to combining vowels with a variety of consonants. All users have expressed surprise at the discoveries they make about their own natural speech production, and to date it has not been an onerous task to spend time learning how to speak. Indeed, the enjoyment is shown as one test phrase currently used for comparing accents between users is "I need a beer!"

6. PERFORMANCE

In performance the audio output of the gesture-to-voice system in GRASSP can be fed directly to a sound system for diffusion as in a regular electroacoustic performance. However, we find it musically more interesting to add modules from the UBC Toolbox to the audio chain, thereby expanding the capabilities of the basic gesture-to-voice system.

For instance, using the horizontal coordinates of the different vowels to control one of the Toolbox's multipan4 or multiPan8 modules results in a one-to-one correlation between the vowel being synthesized and its localization within the concert hall. In effect, the audience is placed within the Petersen and Barney vowel quadrilateral. Or, from an articulatory phonetics point of view, it is as if the audience is inside the mouth at the quadrilateral approximates of the tongue position during vowel production. Because of the open architecture of the code and the flexibility of Max/MSP, control signals can be pulled from any point in the synthesis chain and mapped onto other parameters. Thus, hand height can also control reverberation amount, different consonants can control different processes, and the left hand contact switches can be used to select different configurations via the matrix presets, or trigger samples or other processes.

Additionally, a very interesting performance configuration is possible by replacing the digitized glottal waveform with the sound of another live performer or with prerecorded samples. The result can be similar vocoder processing, with the strength of the effect dependent upon the richness of the substituted waveform.

7. FUTURE DIRECTIONS

Future work with GRASSP will include the recoding and refinement of the Max/MSP patches to overcome the limitations of sensitivity caused by the lack of precision in floats. Additionally, we may recode the entire synthesizer as a single Max/MSP object, making it easier for other users to include the object in their compositions. We will also be expanding the use of the interface via the UBC Toolbox to include graphic processing and the control of kinetic sculpture. The Essential Reality P5™ glove might also be of some interest once its yaw, pitch, and roll data is updated more frequently.

8. CONCLUSION

We have created an environment called GRASSP in Max/MSP that supports three main components for creating adaptive mappings between controllers and expressive sound and video. The components include objects for the Controllers, a Trainer and output control from the UBC Toolbox. To illustrate the effectiveness of GRASSP we have implemented a gesture-to-voice system based on Glove-TalkII since it requires all the components to be used. Our implementation required us to build controller objects for a Cyberglove™, a Polhemus Fastrak™, a custom left-handed glove and a foot pedal. We also ported the JSRU parallel formant speech synthesizer for our speech output. We created a unique training method to allow performers to adjust the mapping between gestures and voice. Using our gesture-to-voice system and the additional components in GRASSP, performers have begun to make new performance pieces. Thus, GRASSP has proven to be an effective environment for new forms of gestural control of expressive content.

9. ACKNOWLEDGMENTS

The GRASSP project is made possible by an Artist-Researcher Grant from the Social Sciences and Humanities Research Council of Canada, and is generously supported by the following University of British Columbia research units: Music, Sound, and Electroacoustic Technology (MuSET), the Media And Graphics Interdisciplinary Centre (MAGIC), and the Institute for Computing, Information, and Cognitive Science (ICICS).

10. REFERENCES

[1] Michel Waisvisz. The hands. In Proceedings International Computer Music Conference, pages 313-318, 1985.

[2] Sonami, L., http://www.sonami.net/lady_glove2.htm.

[3] Wanderly, Marcelo and Battier, Marc, eds. Trends in Gestural Control of Music. IRCAM – Centre Pompidou – 2000.

[4] http://festvox.org/festival/

[5] Fels, S. and Hinton, G., Glove-TalkII: A neural network interface which maps gestures to parallel formant speech synthesizer controls, IEEE Trans on Neural Networks, Vol 9, No. 1, pp. 205-212, 1998.

[6] P. Cook, "SPASM: a Real-Time Vocal Tract Physical Model Editor/Controller and Singer: the Companion Software Synthesis System," CMJ (Computer Music Journal), 17: 1, pp 30-44, 1992.

[7] Orio, N., Schnell, N., and Wanderley, M., Input Devices for Musical Expression: Borrowing Tools from HCI, New Interfaces for Musical Expression Workshop (NIME)'01 at ACM CHI 2001, Seattle, USA, April 2001.

[8] Fels, S., Using Radial Basis Functions to Map Hand Gestures to Speech, in Radial Basis Function Networks 2, New Advances in Design, ed. Howlett, R. J. and Jain, L. C., Physica-Verlag, pp. 59-101, 2001.

[9] http://www.opusonemusic.net/muset

[10] Rye, J.M. and Holmes, J.N. A Versatile Software Parallel-formant Speech Synthesizer. Joint Speech Research Unit Report No. 1016, 1982.

[11] Peterson, B. and Barney, H. "Control Methods Used In a Study of the Vowels," America, 24, 1952.

The 'E' in NIME: Musical Expression with New Computer Interfaces

Christopher Dobrian
University of California, Irvine
303 Music and Media Bldg., UCI
Irvine CA 92697-2775 USA
(1) 949-824-7288
dobrian@uci.edu

Daniel Koppelman
Music Department, Furman University
3300 Poinsett Highway
Greenville SC 29613 USA
(1) 864-294-2094
daniel.koppelman@furman.edu

ABSTRACT
Is there a distinction between New Interfaces for Musical Expression and New Interfaces for Controlling Sound? This article begins with a brief overview of expression in musical performance, and examines some of the characteristics of effective "expressive" computer music instruments. It becomes apparent that sophisticated musical expression requires not only a good control interface but also virtuosic mastery of the instrument it controls. By studying effective acoustic instruments, choosing intuitive but complex gesture-sound mappings that take advantage of established instrumental skills, designing intelligent characterizations of performance gestures, and promoting long-term dedicated practice on a new interface, computer music instrument designers can enhance the expressive quality of computer music performance.

Keywords
Expression, instrument design, performance, virtuosity.

1. INTRODUCTION
The title and popularity of this conference, *New Interfaces for Musical Expression*, now in its sixth year, demonstrates the international interest in the design of new methods by which to enhance the expressive power of computer music. In this article we examine some generally accepted assumptions about computer music expression, and some practices associated with new interfaces, in order to draw attention to the question of whether musical expression in performance is being adequately addressed in much current research on realtime computer music interfaces.

2. WHAT IS EXPRESSION?
2.1 Common definition
expression:
felicitous or vivid indication or depiction of mood or sentiment; the quality or fact of being expressive [15]

expressive: effectively conveying meaning or feeling [16]

2.2 Composition vs. performance
Music can indicate mood or sentiment, and can convey meaning or feeling simply by its organization (composition) of sound elements [3], and the performer of music—often called the "interpreter" in the case of composed music—provides expression by evincing that organization, by adding shape and nuance to the given materials. So it is important to distinguish whether we are talking about expression in composition—expressive characteristics of musical materials and their organization by a composer—or expression in performance—expressive gestural nuance in real time. For the purpose of this article, we are referring to the latter—the nuance that a live performer adds to the available materials. The *New Grove Dictionary of Music and Musicians* notes that this form of expression encompasses "those elements of a musical performance that depend on personal response and that vary between different interpretations." [4][1] In the case of "programmable instruments" and live control of compositional computer music algorithms, the distinction between compositional expression and performative expression may be blurred somewhat; the performer may be shaping primary characteristics of the composed/improvised musical materials themselves. What we are specifically concerned with in this discussion, however, are those characteristics of the live performance that enhance expressive communication beyond that which is contained in the materials on a notated page or a pre-programmed algorithm.

2.3 Performers bring expression to music
Poepel [22] described a mechanism by which "performers code expressive intentions using expressive-related cues" (including "tempo, sound level, timing, intonation, articulation, timbre, vibrato, tone attacks, tone decays and pauses") and listeners "receive musical expression by decoding" these cues. This implies that performer expression, like language, depends on a set of conventional signifiers and an understanding of those signifiers shared by both performer and listener. These cues are generally at a different logical level than that of each individual parameter of a sound, and thus are not easily emulated with simple one-to-one mappings of gesture to sound parameter. "From the perspective of a musician, live performances allow him or her to make real-time choices that affect the interpretive variation in the music. These choices give the performance a trajectory and a range of variation that define the expressiveness and power of the performance. Techniques for creating this variation involve subtle control over

[1] The *New Grove* article on "expression" treats primarily the organizational aspects of expression in composed music.

aspects such as timing, volume, timbre, accents, and articulation—sometimes implemented on many levels simultaneously.” [20] Thus, what we think of as musical expression in performance usually involves the performer’s contribution of culturally understood variations of specific sonic attributes at the note level—e.g. intonation, timbre, vibrato speed and depth, etc.—and attributes of the musical structure at the phrase level—rubato, crescendo, etc.

Even the physical gestures made by the performer affect the listener’s perception of the music. In the visual arts, a viewer can respond to specific traits of a pencil line, such as its smoothness, which can be “a tactile quality, a pleasurable one. [One’s] reaction to the line is not just to its formal qualities (continuity, for instance, or darkness or lightness), but also a kinesthetic sympathy (the Italians call it syntony) with the hand that drew it—the pressure, the weight, the gestural control, etc. Is the line, then, expressive or does it seem expressive because it is a trace of what I perceive/read as an expressive human action?” [Murata, M., personal correspondence]. Similarly, when one hears a sound, one can imagine and empathize with the physical gesture that might have created the sound; in this way, sounds imply gesture, even choreography. Conversely, viewing the physical gesture that a performer makes can influence the listener/viewer’s perception of the sonic expression. Computer interfaces can dissociate gesture from result to varying degrees by the way that software intermediates the relationship between gesture and resulting sound. (A one-to-one correspondence such as a mallet striking a marimba is an example of a simple gesture-result relationship, while a finger pushing the play button on a CD player exemplifies the opposite extreme in which a simple neutral gesture produces a complex musical result.) Jordà [19] evaluates this relationship as the “efficiency” of the interface, defined as the ratio of “musical output complexity” to “control input complexity”, but acknowledges that these are “quite fuzzy terms”, and that while computer-mediated controllers can provide more “efficiency” than most acoustic instruments, they often lack the “expressiveness” (flexibility, diversity of micro-control, etc.) of traditional instruments.

2.4 Can a machine be expressive?

Just as philosophers and computer scientists have debated the question of machine intelligence for decades [26, 25], there continues to be debate as to whether true musical expression (conveying meaning or feeling) can be produced by a computer. The fundamental question in both cases is whether the *appearance* of intelligence or expression is sufficient to believe that intelligence or expression exists. Emulations of expressive performance have been attempted by means of rule-based programs (e.g., [7, 10, 14]) and by machine learning, notably case-based reasoning (e.g. [1, 2, 3]). However, it is questionable whether systematic emulation of performer expression derived from other musical contexts is the same as what human performers do each time they interpret a composition. Computer musicians working with realtime performance systems have expressed the cautionary view that “just as the –ivity suffix in the word ‘interactivity’ connotes ‘a quality of’ interaction that can only be artificial in a machine, ‘expressivity’ for a computer can only be a demonstration of an artificial or simulated quality of being expressive in the sense that we apply it to human music making: the conveyance of meaning or feeling.” [11] “Be aware though: music instruments, being machines,...cannot be expressive…since machines do not have anything to express. However, they can be used to transmit human expressiveness, and they can do that in many ways and with different degrees of success. Our instruments will achieve this for better or worse, in the measure they permit the performer to transform a mental musical representation into musical gesture and sound.” [19] These musicians feel that the expression comes from the performer, and the instrument enables—and ideally facilitates and amplifies—that human expression. Although we may speak of an “expressive instrument” for the sake of brevity, it is important to recognize that we usually mean “an instrument that affords expression”, that is, “an instrument that enables the player to be expressive”.

3. CONTROL AND EXPRESSION

3.1 Control ≠ Expression

The mere presence of a finely calibrated instrument does not guarantee that it will be put to an expressive use. It might be said that the ability to control a sound generator, and the means of that control, are the *tool* or the *medium* by which expression is made possible. But it is important to note that one should not therefore equate control with expression. The performer’s expression is the significant content that is made possible to convey by the ability to control the sound generator in real time.[2]

The most basic need for a controller is that it accurately capture the data provided to it by the human interface. Another basic need is that the software provide correspondences between input data and output sound that are sufficiently intuitive for both performer and audience. It has been suggested that “the expressivity of an instrument is dependent on the transparency of the mapping for both the player and the audience.” [13], be it through direct mapping schemes, or more sophisticated gesture analysis (e.g. [8, 23]). Transparent or not, the correspondences must be learnable, repeatable, and sufficiently refined to enable control of the sound that is both intimate (finely detailed) and complex (diverse, and not overly simplistic).

Thus, control is a precondition for enabling expression, but is not in and of itself sufficient. Expressive control requires at a minimum that the interface provide accurate capture of gesture, and that the mapping of input to sonic result be situated at the appropriate level of structural detail (microscopic, mid-level, or macroscopic). Expressivity can be enhanced by intelligent recognition of gesture in order to characterize the gesture and make the appropriate mapping.

3.2 Simple and complex mapping schemes

In trying to design an instrument that will enable expression, it is necessary to consider how the performer will provide musical expression, notably how the performer’s gesture will affect the sound. Simple one-to-one mapping of input control data to a particular sound parameter is essential in many cases in order for the performer to have precise control, but such control is not equal to expression. Expressive control relies on more sophisticated use of the control input information, such as through one-to-many mapping of control data to a combination of parameters, recognition of complex characteristic gestures, or other methods that enable the

[2] The English philosopher R.G. Collingwood formulated “his celebrated distinction between art and craft, according to which craft is a means to an end and must therefore be conducted according to the rules laid down by that end, whereas art is not a means but and end in itself, governed by no external purpose.” [4] The building of instruments/controllers is the craft of enabling expressive control, whereas the expressive use of the instrument is an art.

simultaneous and multi-dimensional shaping of combinations of parameters. Good performers use this complex multi-parametric shaping to encode a meaningful variation from a norm, be it by adding nuance not specified in the score of a composition or by varying established standards of consistency (steady tempo, discrete scale steps of pitch). For example, vibrato in a flute or a violin sound, which is rarely notated but which is a generally accepted method of playing *espressivo* in Romantic and contemporary music, is a simultaneous modulation of pitch, loudness, and timbre (i.e., frequency, amplitude, and spectrum), with these multiple modulations themselves being shaped (modulated) by the performer.

It is one thing to create a controller with simple mappings that even a novice can use with satisfying results without training (e.g., [8]), but it is quite another to develop an instrument that provides maximal control, diversity, and efficiency, in order to best enable expression by a skilled user. For an instrument to be considered potentially expressive by a trained musician, it must necessarily have a certain degree of complexity in the relationship between input control data and sonic result. So it is reasonable to expect that such an instrument will have a certain learning curve (c.f. [19], p. 176 ff.); a performer will require a certain amount of training and practice to achieve good control of it. For high-quality musical expression, an instrument should be mastered; the performer should achieve a level of virtuosity.

4. WHITHER VIRTUOSITY?

4.1 Common Definition

virtuosity: great technical skill [17]

First we wish to clarify that our use of the term "virtuosity" refers to a person having complete mastery of an instrument, such that s/he can call upon all of the capabilities of that instrument at will with relative ease; we are not referring simply to extravagant displays of extreme speed or dexterity. (Indeed, a computer is capable of playing at speeds much greater than humans, so playing fast notes on a computer instrument is no longer necessarily a display of virtuosity by the performer.)

4.2 Virtuosity Facilitates Expression

A primary value of virtuosity is that it transfers much of the knowledge of how to control the music to the subconscious level for the performer; basic functionality of the interface is no longer necessarily foremost in the performer's thoughts. "The human operator, once familiar with the system, is free to perform other cognitive activities whilst operating the system." [18] When control of the instrument has been mastered to the point where it is mostly subconscious, the mind has more freedom to concentrate consciously on listening and expression.

4.3 Lack of Virtuosity Inhibits Expression

How much time is needed to develop mastery or virtuosity on a musical instrument? Of course there is no definitive answer to this question, but it is probably safe to say that virtuosi have almost invariably spent years of highly focused practice and experience on their instrument. It is also safe to say that almost all sophisticated musical instruments have evolved over many years or even many centuries of technological refinement, focused development of technique over a long period by numerous different musicians (often competing, but also usually learning from each other), and a production of a large body of repertoire that contributes to both the technological and the technical advances of the instrument.[3] Is it then naïve to think that masterful performance will occur on an instrument that was designed, developed, built, composed for, and rehearsed only within the last year or even the last few months or weeks?

The vast majority of performances of computer music that involve new interfaces, new instruments, alternative controllers, etc. are more experimental than they are refined and virtuosic. They are generally performed by someone who has only recently encountered the instrument, has had relatively little time to explore and understand the subtleties of expressive capabilities it affords, may be dealing with an interface the mappings of which have only recently been programmed (let's be honest, in some cases as recently as the dress rehearsal), and who does not truly have a performance-level mastery of the instrument as it is configured. In many cases, the performer is someone who is a composer or technician more than a professional instrumentalist or stage performer.

There is nothing wrong with this experimentation. Indeed, it is vital to the progress of this field. And in fact there is nothing so very wrong with putting this experimentation onstage in a less-than-refined form at demonstrations, workshops, and conferences. But it would be a mistake to pretend that such an onstage experiment is a good representation of the *expressive* capability of that instrument, or that it can—except in a few fortunate instances—be legitimately compared to a high-caliber professional virtuosic music performance.

Schloss [24] has remarked that "some pieces performed nowadays claim to be interactive, but in reality they are simply not finished yet. So the performance involves the 'baby-sitting' and 'knob-twiddling' we might see on stage that is so unsatisfying to watch." The lack of virtuosity on new musical interfaces is apparently another case of the "elephant in the corner"—a big bothersome issue that everyone knows is there but is hesitant to discuss.

4.4 New Instruments Modeled on Old Ones

One approach to improving virtuosity and expressivity in live computer music has been to design instruments modeled on existing acoustic instruments. Indeed, this is still an attractive approach to many in the field, as demonstrated by the NIME 2006 "special paper session" on *Digital Interfaces for the Violin Family*. Early designers of synthesizers, and designers of the MIDI protocol, recognized the value of taking advantage of the years of skill developed by large numbers of keyboardists. Designers of other commercial computer-enabled acoustic instruments (computer-captor-enhanced violins, saxophones, etc.) and instrumental controllers (Zeta Strados violin pickup and Synthony II MIDI processor, Yamaha G1D guitar pickup and G50 guitar MIDI converter, Yamaha WX5 wind controller, etc.) have also attempted to make controllers that will allow capable performers of those instruments' acoustic counterparts to bring their expressive skills to computer music.

Computer interfaces that are closely modeled on existing acoustic instruments can reduce the learning curve for those performers who are experienced on the acoustic counterpart,

[3] A particularly musically satisfying fusion of 'technical' and 'artistic' issues occurs in works such as Chopin's Etudes for piano, which address both in a meaningful way.

tap into the existing resource of performers' virtuosic skill, and become readily usable by a larger pool of performers compared to more novel interfaces. But interfaces based on existing instruments present some challenges as well.

Although there is a growing number of expert instrumentalists who are interested in performing interactive computer music, many acoustic instrumentalists remain reluctant to use new interfaces, perhaps because they feel intimidated by their own lack of computer music knowledge, and/or because they have had experience with poor computerized models of their instrument in the past.

Indeed for the designer of such an instrument, there are many challenges in trying to make the instrument seem natural and intuitive for a player. First of all there is the problem of knowing exactly what best to capture in the player's gesture. (For example, to capture vibrato on a violin, should the computer monitor the pitch of the bowed note, or the length of the bowed string, or is it necessary to know also the movement of the hand and finger in order to monitor the spectrum-altering and amplitude-damping effect of different finger angles?) There may also be a need to recognize and categorize certain second- or third- order aspects of the gesture (e.g., recognize that a vibrato is taking place, take note of its rate and depth, the rate of change, etc.) And crucially, there is the question of how to map data from the interface onto specific changes in the sound, i.e., to map the relationship of interface to sound generator.

As noted earlier, one-to-one mapping of a single input control parameter to a specific parameter in sound production is effective for transparent and repeatable control (e.g., selecting a specific pitch), but the subtle details of performer *expression* usually require more complex mappings. Instrument designers benefit from examining the gesture-sound relationships that exist in acoustic instruments, if only in order to design instruments that are more intuitive for virtuoso players, and that provide a rewarding complexity that encourages practice to achieve mastery. Thus, mapping plays a significant role in the success or lack of success of an instrument in both the short term (enabling expression) and the long term (encouraging development of virtuosity).

5. POSSIBLE DIRECTIONS TOWARD MORE EXPRESSIVE INTERFACES

5.1 More Participation by Virtuosi

If we accept the premise that virtuosity facilitates expression and lack of virtuosity inhibits expression, then it stands to reason that computer music can be made more expressive by more virtuosic performers. One approach is to use sensor-equipped acoustic instruments or an interface modeled on an acoustic instrument to take advantage of the virtuosity already developed by experienced players. Another approach is for experienced performers to dedicate the time necessary to develop virtuosic mastery of a new interface. This often requires years of dedication to a particular interface, but the rewards of such dedication are demonstrated by performers such as Laetitia Sonami and Michel Waisvisz (and of course, in the pre-computer age, Theremin virtuosa Clara Rockmore). Experimental performances by inexperienced musicians or by performers who have incompletely mastered a new interface are often acceptable as a proof-of-concept demonstration of a new design (particularly in a technical conference), but when done on the concert stage are subject to rigorous musical and aesthetic critique.

5.2 Still Better Mapping Ideas

Defining correspondences between gesture and sound—i.e., mapping control data to sonic parameter(s)—has been the focus of an enormous amount of research (e.g., [28]). Some basic problems have been recognized, yet many still have not been satisfactorily solved. "Strategies to design and perform these new instruments need to be devised in order to provide the same level of control subtlety available in acoustic instruments." [27] One problem is the need to have intuitive yet detailed control of a computer music instrument that might itself be vastly multi-dimensional. However, with "controllers that output more than 3 continuous streams from the same gesture, it can be exceedingly difficult to reliably reproduce a gesture in performance." [21] Indeed, with a motion capture system, "a single performer wearing a standard set of thirty markers, with three coordinates per marker, produces a stream of 90 simultaneous continuous parameters available for musical control.…This profusion of control data presents…a challenge of the limitations of awareness for the performer." [12] A "fly-by-wire" strategy (i.e., a divergent one-to-many, mapping), whereby a small amount of control data provides the necessary guidance to a complex system, is implied for intuitive-yet-complex control of sound. Such a system requires some time to master.

5.3 Feedback

Instrumentalists rely on tactile and visual information as well as sonic information. A pianist can see and locate a specific key before playing it, can use the resistance of the key-action mechanism to help know how hard to press the key, and can use the feeling of adjacent keys to keep track of hand position. Similar examples can be found for almost any acoustic instrument. Thus, visual information (telling the player what is possible, and where controls are) and visual feedback (telling the player what happened) are very helpful in a new interface. Likewise, haptic (tactile) feedback is useful for gauging one's progress on a continuum. Some new interfaces lack sufficient visual and haptic feedback: for example, video motion tracking software allows the performer unfettered movement, but provides only sonic feedback. Sonic feedback, while necessary and valuable for musical performers, is always retrospective; the sound has already occurred by the time the feedback has been received. One can learn to play such a "virtual" instrument virtuosically, but the learning curve is decidedly high.

5.4 Gesture Recognition

While accurate tracking of gestural information is crucial for good control and expression, the software that interprets that data can be made even more "intelligent" by analyzing characteristics of that gesture. This includes second- and third-order analyses of the input data—recognizing not only the input value, but also the speed and direction of change, acceleration of change, etc. Some implementation of pattern analysis, recognition, and categorization can also lead to more intelligent software. For example, in addition to tracking gesture, it is useful to know what kind of gesture it is, thus making it possible to associate meaning with that gesture. Techniques of pattern matching and gesture recognition have a long and well-developed history in the field of artificial intelligence, and much of that knowledge can be fruitfully applied to musical gesture.

5.5 Critical Discourse

Critique is a vital aspect of intellectual and artistic life, which obliges analysis, evaluation, and discussion, which in turn leads to improvement. Conferences such as NIME, ICMC, SEAMUS, and SMC focus predominantly on technical presentations and music concerts for an audience of like colleagues; however, critical discourse regarding the quality of the music, aesthetic values, and effectiveness of new interfaces takes place mostly in private conversations over an after-concert beer. In addition to being the sites for exchange of technical information, these conferences can and should serve as ongoing public forums for evaluative aesthetic discourse, encouraging increased public critique and debate.

5.6 Repeat Performances

In order for listeners to appreciate and evaluate the expressive qualities of a performance, access to multiple interpretations—whether by the same artist or by different artists—is essential. However, since most of the leading computer music conferences focus their selection criteria for performances on either the composition or the technology, placing a premium on originality, repeat performances are rarely available. This is the opposite of the situation in classical music, where the most visible performances are usually pieces from a standard repertoire rather than newly composed works. Performance interpretation and expression are highly valued in the classical genre, while in live computer music circles the quality of the individual's performance is usually a secondary consideration[4]—or may even be impossible to evaluate because the piece is only heard once and the interface is so novel. If expression is truly a valued component of this new art form combining humans and machines, then time could be allocated at major public events for the display, critique, and contemplation of the unique qualities brought to life in a particular realtime performance. One result of such concert programming might be the emergence of a number of "classic" pieces in the genre, in which different performers' interpretations could be critically compared, thus focusing attention on the expressive use of the interface, rather than its design characteristics.

6. CONCLUSION

If musical expression with new computer interfaces is to reach the level of sophistication achieved by major artists in other specialties (jazz, classical, etc), it will be necessary to encourage further development in the following areas: continued focused research on strategies for better mapping, gesture recognition, and feedback; dedicated participation by virtuosi (utilizing existing virtuosity and developing new virtuosity); repertoire development for—and multiple performances with—a given instrument as a way to further its development; and more opportunities for critical discourse, both within the community of practitioners and among non-practitioners. The future is rich with possibilities for involvement by a wide array of interested and talented artists and artisans.

7. ACKNOWLEDGMENTS

Thanks to music historian Dr. Margaret Murata for contributing her observations on historical and philosophical views of expression in music.

[4] At a recent electronic music conference, all the composers were called to the stage for a group photo, leaving the few performers in attendance to ponder the implications of their exclusion.

8. REFERENCES

[1] Arcos, J. and Lopez de Mántaras, R. "AI and music from composition to expressive performance". *AI Magazine* 23:3, pp. 43-57, Menlo Park, CA: American Association for Artificial Intelligence, 2002.

[2] Arcos, J. and Lopez de Mántaras, R. "An Interactive Case-Based Reasoning Approach for Generating Expressive Music". *Applied Intelligence* 14:1, pp. 115-129. Boston: Kluwer, 2001.

[3] Arcos, J., Lopez de Mantaras, R., and Serra, X. "SAXEX: A Case-Based Reasoning System for Generating Expressive Musical Performances". *Journal of New Music Research* 27:3, pp. 194-210. New York: Routledge, 1998.

[4] Baker, N., Paddison, M., and Scruton, R. "Expression", in Sadie, Stanley and Tyrell, John, eds. *The New Grove Dictionary of Music and Musicians*, second edition. London: Macmillan Publishers Limited, 2001.

[5] Bevilacqua, F., J.Ridenour, and D. Cuccia, "Mapping Music to Gesture: A study using 3D motion capture data", Proceedings of the Workshop/Symposium on Sensing and Input for Media-centric Systems, Santa Barbara CA, 2002.

[6] Cadoz, C. and Wanderley, M. "Gesture-Music". *Trends in Gestural Control of Music*. Paris: IRCAM, 2000.

[7] Canazza, S., De Poli, G., Roda, A., and Vidolin, A. "Analysis and Synthesis of Expressive Intention in a Clarinet Performance". *Proceedings of the 1997 International Computer Music Conference*, pp. 113-120. San Francisco: International Computer Music Association, 1997.

[8] Chew, E., François, A., Liu, J., and Yang, A. "ESP: A Driving Interface for Expression Synthesis". *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada, 2005.

[9] Cooke, D. *The Language of Music*. Oxford: Oxford University Press, 1959.

[10] Dannenberg, R. B., and Derenyi, I. "Combining Instrument and Performance Models for High-Quality Music Synthesis". *Journal of New Music Research* 27:3, pp. 211-238. New York: Routledge, 1998.

[11] Dobrian, C. "Strategies for Continuous Pitch and Amplitude Tracking in Realtime Interactive Software". *Proceedings of the 2004 conference on Sound and Music Computing* (SMC04), Paris: IRCAM, 2004.

[12] Dobrian, C. and Bevilacqua, F. "Gestural Control of Music Using the Vicon 8 Motion Capture System". *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03)*, Montréal, Québec, Canada, 2003.

[13] Fels, S., Gadd, A., and Mulder, A. "Mapping transparency through metaphor: towards more expressive musical instruments". *Organised Sound* 7:2, 109-126. Cambridge: Cambridge University Press, 2002.

[14] Friberg, A. *A Quantitative Rule System for Musical Performance*. Ph.D. dissertation, Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, Sweden. 1995.

[15] *http://www.webster.com/dictionary/expression*

[16] *http://www.webster.com/dictionary/expressive*

[17] *http://www.webster.com/dictionary/virtuosity*

[18] Hunt, A. and Kirk, R. “Mapping Strategies for Musical Performance”. *Trends in Gestural Control of Music*. Paris: IRCAM, 2000.

[19] Jordà Puig, S. *Digital Lutherie: Crafting musical computers for new musics’ performance and improvisation*. Ph.D. dissertation, Barcelona: Departament de Tecnologia, Universitat Pompeu Fabra, 2005.

[20] Marrin, T. *Inside the Conductor’s Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture*. Ph.D thesis, Cambridge, MA: Massachusetts Institute of Technology, 2000.

[21] Momeni, A. and Wessel, D. “Characterizing and Controlling Musical Materials Intuitively with Geometric Models”. *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03)*, Montreal, Québec, Canada, 2003.

[22] Poepel, C. “On Interface Expressivity: A Player-Based Study”. *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada, 2005.

[23] Rovan, J., Wanderley, M., Dubnov, S., and Depalle, P. “Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance”. Paris: IRCAM, 1997.

[24] Schloss, W. A. Using Contemporary Technology in Live Performance: The Dilemma of the Performer”. *Journal of New Music Research*, 32:3, pp. 239–242. New York: Routledge, 2003.

[25] Searle, John. “Minds, Brains, and Programs”. *The Behavioral Brain Sciences*, vol 3., pp. 417-424. Cambridge: Cambridge University Press, 1980.

[26] Turing, Alan M. “Computing Machinery and Intelligence”. *Mind*, 59:236, 1950.

[27] Wanderley, M. “Gestural Control of Music” *Proceedings of the International Workshop on Human Supervision and Control in Engineering and Music*. pp. 101-130. Kassel, Germany, 2001.

[28] Wanderley, M. and Battier, M. (eds). *Trends in Gestural Control of Music*. Paris: IRCAM, 2000.

32kg: Performance Systems for a Post-Digital Age

John Richards
Music, Technology and Innovation Research Centre
De Montfort University, Leicester
LE1 9BH, UK
+44 (0)116 207 8269
jrich@dmu.ac.uk

ABSTRACT
Why is a seemingly mundane issue such as airline baggage allowance of great significance in regards to the performance practice of electronic music? This paper discusses how a performance practice has evolved that seeks to question the binary and corporate digital world. New 'instruments' and approaches have emerged that explore 'dirty electronics' and 'punktronics': DIY electronic instruments made from junk. These instruments are not instruments in the traditional sense, defined by physical dimensions or by a set number of parameters, but modular systems, constantly evolving, never complete, infinitely variable and designed to be portable. A combination of lo- and hi-fi, analogue and digital, synchronous and asynchronous devices offer new modes of expression. The development of these new interfaces for musical expression run side-by-side with an emerging post-digital aesthetic.

Keywords
Post-digital, modular, dirty electronics, bastardisation, punktronics, portability, DIY, eBay, performance, live electronics.

1. INTRODUCTION
The scope of this paper is to look at the artistic, cultural and social impact of new interfaces for musical expression in the field of live electronics. The term live electronics is used here to define a performance practice of electronic music that can loosely described as experimental, often improvised and of an interactive nature. The economics of live electronics are also considered. It is safe to say that budgets for most concerts in this field are relatively small and that the venues for such performances are also small. These concerts are often run and promoted by dedicated enthusiasts not for profit.

2. FROM ON-LINE TO OFF-LINE
In a world where there is an ever-increasing miniaturisation of technology, why are so many musicians working in the broad field of live electronics having to consider the weight and size of their performance set-ups? In some respects, international travel has never been easier with frequent and relative cheap flights from continent to continent. Passenger air traffic continues to grow. There was a 6.5% rise in 2005 compared to the previous year [1]. The need to travel has in part been due to the development of global networks and new communities created by the Internet. The ideas expressed by radical publications such as *Mondo 2000* [2] on virtual reality - cruise the datascape, ride the electronic range, vacation in virtual reality - have only been part of the consequence of the digital age. Conversely, the Internet, despite its virtual nature and the potential for hidden and false identities, has physically brought people together. As Nicholas Negroponte [3] states in his seminal text *Being Digital*, the Internet is "all about people not places." Nevertheless, in many cases it is inevitable that through the power of communication there will be the desire to meet in a 'real' place. For example, there are current blogs that are also initiating blogwalks [4] with "face-to-face" meetings and where "minds can meet and inspire each other in direct conversation." Musicians working in specialised areas of music have benefited greatly from the Internet. In the case of live electronics, an international circuit has developed with musicians travelling on regular basis to perform in different venues. This has become the norm rather than the exception. To quote Negroponte [3] once more: "The global nature of the digital world will increasingly erode former and smaller demarcations." The concept of a travelling musician is by no means new, but it is the distance and means, air travel, and what is being carried that are relevant to this paper.

3. BASTARDISATION AND THE POST-DIGITAL
The interest in what could be loosely described as 'digital music' is beginning to wane. Negroponte [5] in *Beyond Digital* highlights changing attitudes towards the digital: "… the technology, is already beginning to be taken for granted, and its connotation will become tomorrow's commercial and cultural compost for new ideas." The term post-digital is becoming increasing used, although there is still relatively little in print on the definition of the term in relation to music or sound. Kim Cascone [6] in "The Aesthetics of Failure: 'Post-Digital' Tendencies in Contemporary Computer Music" has undoubtedly begun the lengthy process whereby there will be many subsequent discussions on the definition of the term post-digital. The notion of 'failure' is central to Cascone's [6] argument in outlining a post-digital aesthetic: "… it is from the 'failure' of digital technology that this new work has emerged: glitches, bugs, application errors, system crashes, clipping, aliasing, distortion, quantization noise, and even the noise floor of computer sound cards are the raw materials composers seek to incorporate into their music." What would seem more central to a post-digital aesthetic is the bastardisation of technology. This certainly goes beyond the digital (this will be discussed in more detail later). Bastardisation implies forcing a system in to a state in which it was never intended, or appropriating something for a use other than what it was initially designed for. For example, in analogue terms, this may involve circuit bending or hacking a sound generating device, or forcing a circuit to oscillate through a feedback loop. The process of

bastardisation is as much about success as it is failure, where the musician is able to find a latent 'music' inside the technology. Cascone's paper is more about defining the musical genre of glitch. The glitch as a digital artifact would seem to purport a purer, more uniquely digital aesthetic, with its own distinctive voice - some form of higher expression of digital music.

4. ONES, ZEROES AND CONTINUUMS

In *The Postdigital Membrane: Imagination, Technology and Desire* Robert Pepperell and Michael Punt [7] question the logical states of 'yes' and 'no' of a digital world. They argue that human experience is all about a continuum between different states: "… identifiable parts are not reduced to oppositions and analysed as dialectical constellations but considered as continuous with each other." They also suggest that: "For the anthropological notion of the 'binary' to be compatible with the computational notion of 'binary' the two terms must be exclusive. '1' or '0' can only be '1' or '0' - there can be no ambiguity or irrationality." A frustration with logic and a desire for the irrational, are key issues concerning a post-digital aesthetic. It is not surprising that many musicians are fascinated with the analogue in its various guises from virtual software synthesisers to 'antique' hardware equipment. This would appear not just a case of nostalgia or retro, but an active stance to expand the range of musical expression currently offered by the use of digital technology. The musician has returned to the analogue with rigour. It is not a question of discarding the digital, but having greater choice. Peter Lunenfeld [8] presents the idea of the analogue and digital representing the 'curved' and the 'crisp' respectively: "We have, in fact, come to expect a certain crispness from digital media precisely because of this stepping, leading some to categorize the analog as somehow natural, less polarized, more curved." Analogue and digital systems also present asynchronous and synchronous ways of thinking and doing. A digital system will always be asynchronous: an analogue source is converted into a binary code, possibly analysed and modified in some way, then converted back to the analogue. Many digital tools for the creation of music are designed to behave in a pseudo synchronous way giving the illusion of working in real-time. This is being blind to the real potential of digital technology. It is also not to say that an asynchronous way of working is any better or worse than the synchronous. There are some benefits musically by being asynchronous. For example, the written score allows a composer to work with sound in a more formal way. In the post-digital, the asynchronous and the synchronous can be themselves. This new musical crossbreed, where different modes of thought and resulting practices co-exist side-by-side, offers greater scope for musical expression.

5. MODULARITY AND OBJECTS

Nic Collins [9], in his introduction to a *Leonardo Music Journal* edition on David Tudor remarks: "The introduction of the microcomputer and MIDI at the start of the 1980s prompted many musicians to swap the soldering iron for software and increasingly affordable commercial music synthesizers". However, current trends in circuiting bending and hardware hacking have helped rekindle an interest in the work of David Tudor, and musicians are returning to their soldering irons. Tudor developed an approach that would seem to have obvious links with currently emerging post-digital aesthetic. Again, it is not just a case of Tudor working in the analogue domain. In an interview with Joel Chadabe, Tudor [10] highlights his interest in DIY electronics and 'small instruments' due to their flexibility, portability, and cost. This was in addition to these instruments providing him with the sound he wanted to hear. Tudor also sought to explore the sonic and musical potential of a circuit, whether this was from a discarded electronic appliance or a circuit made from scratch, and 'composing' inside electronics (this was also the name given to one of Tudor's ensembles). What is of greater significance is that Tudor combined these small devices to make-up larger modular performance environments.

With the advancement of digital technology, Tudor's modular approach towards working in the field of live electronics has not been supplanted. On the contrary, software developments such as Max/MSP [11] and PD [12] have helped reinforce an analogue model that is modular. The object in Max/MSP is synonymous with the module: the user patches components together to construct an 'instrument'. As Negroponte [3] argues being digital is all about doing: "Don't Dissect a Frog, Build One". In a digital world nothing is ever finished. The Internet presents dynamic texts that can be constantly altered, operating systems and software are constantly upgradeable, and computers themselves comprise of interchangeable modules. Doing is easy. In the virtual world when mistakes happen there is always the undo key. It stands to reason, therefore, that this spirit of 'doing' is shaping a post-digital practice. Hack, build, modularise, re-configure, upgrade, discard - these techniques and ways of thinking have spread across different domains to suggest a new form of musical instrument.

6. DIRTY ELECTRONICS AND LIVELY INSTRUMENTS

"… miniaturisation has changed our experience of mechanism … Our best machines are made of sunshine; they are all light and clean because they are nothing but signals, electro-magnetic waves, a section of a spectrum." [13] Our new digital technology is indeed clean, bright and weightless. It is also elliptical, intangible and sterile. To quote Nic Collins [9] again: "The electronic future, as envisioned for the past 80 years or so, has usually taken one of two forms: the streamlined, antiseptic, utopian vision in which technology allows us ever more control (the iPod future) and the messy, chaotic, dystopian vision in which electronics multiply and decay, leaving us at their mercy …". In terms of the appearance of the home computer, we have moved from beige plastic casing to silver and white finishes and minimalist design looks. Although there may be some novel software, and ways of personalising computer settings, for example, by having a background image of your skiing holiday or pet, these machines are faceless and characterless. It is natural that a musician has a unique relationship with the instrument with which they work, and that these instruments assume great importance. In a post-digital age electronic instruments are 'lively' instruments with character and unique voices. This is due to the 'doing' in the post-digital where self-made instruments take on personalities. These lively instruments are exemplified by the instruments of Reed Ghazala [14, 15] with their futuristic names and imaginative appearance, the infra-instruments of Phil Archer [16], the analogue cast-offs of Merzbow [17], Tom Bugs' Bits and Bobs Modular [18] and the modified toys of Brian Duffy [19]. This is only a snapshot of instruments and interfaces that are being used that reflect a much broader current trend towards the post-digital.

There would appear a backlash or reaction against the corporate hi-tech multi-nationals. The post-digital is characterised by a desire to subvert and challenge, what has become in many quarters, an acceptance of a digital hegemony. The influences of cyberpunk and post-cyberpunk seem to be spilling-over in to music; or perhaps just the essence of punk is re-emerging within certain music circles. A development and interest in what could be described as 'dirty electronics' has taken root. These are electronic instruments and working methods that are directly opposed to those of a mass produced digital culture and may include some of the following characteristics: designer trash (deliberately made to look beaten-up or broken), ugly, cheap, heavy, hand-made, designed to be handled or to come in contact with the body, ready-mades, hacked, bent, fedback and kitsch. The battered and bubble gum glued instruments of John Olson and Nathan Young from the group Wolf Eyes [20], or the bow-stab infra-violin of Phil Archer [16] are but two examples of this development. This practice could be further defined as 'punktronics'.

The electronic circuits for the instruments discussed above are often crude and built by modifying existing sound generating devices or by constructing circuits with stripboard or dead bugging. These instruments are also a result of a profligate attitude towards technology that is prevalent in the West. This point is well put by Norbert Möslang [21] in his article "How Does a Bicycle Light Sound?: Cracked Everyday Electronics": "For the last 20 years, various small electronic tools have been mass produced and thrown onto the market ... just waiting to be cracked! This is the wreckage of Western civilization, as it were, and the musician is the ethnologist who collects and cracks this wreckage."

The re-assemblage of this 'wreckage' has become a fixation mainly due to global access to jumble and junk through the Internet and companies such as eBay [22]. Access to the most obscure electronic component or musical artifact is now possible as obsolete technology is sought on-line with the intention of appropriating it for a new use. The full affect of eBay on musical culture is still being felt. It is not a question these days of getting hold of something, but rather what is to be done with it. There is a kind of scavenging mentality where the whole course of recent music history can be picked over and elements extracted to build a new musical vision of the future. The pickings are rich.

7. DISEMBODY - RE-EMBODY

'Physically' reconnecting to an instrument or interface would seem part of a post-digital aesthetic. This may not necessarily mean playing an instrument in a 'traditional' way. The bio art of Eduardo Kac, where technology may penetrate the skin, for example, a microchip is implanted in Kac's ankle in his piece *Time Capsule*, demonstrates how the physical and technological can co-exist in a radical hybrid [23]. The concept of body contacts expressed by Reed Ghazala also re-embodies the performer in a very direct and tangible way in regards to a musical interface. A body contact is where the human body, being able to conduct electricity, becomes physically part of an electronic circuit. Ghazala [14] termed this way of working as BEAsape: "BEAsape is an acronym for BioElectronicAudiosapian. Instrument/animal, mutant or hybrid, both musically and zoologically the BEAsape pushes boundaries". Ghazala [14] also writes: "I felt that a new, albeit temporary, creature was created when a musician played a body-contact instrument - in this moment when the electricity of both bodies intertwines, the same essential electricity that if interrupted would cause each body to die. I was changed and the circuit was changed, and I had trouble deciding where each of us began and ended. I simply concluded that we were something new, and we were one." The idea of using the human body to conduct electricity for artistic purposes in not new. As early as 1744, Georg Mathias Bose created the 'salon' performance piece *Venus Electrificata* (also known as the Electric Kiss) [24].

For some performers, there is something appealing about physical exertion in creating a sound or playing an instrument. Micro-gestures suitable for many digital interfaces only allow for a fraction of the gesture range of the human body. The interface in the post-digital provides 'something to hold on to': larger knobs or controls (the bigger the better), heavy metal body contacts, and connectivity through cables and wires.[1] Masami Akita (Merzbow) [17], refers to 'noisehands' in an interview with Arthur Potter: "Most Japanese noise artists never use computers or very high-tech equipment. We tend to be very low-tech and analogue, so our actions show the effects of expanded noisehands, muscles … the body's movement." Ergonomics and biological condition have again become central to the idea of musical interface.

This ties back in with the idea presented earlier in this paper of lively instruments, and in some respects, cultural references to machines of a bygone era. Erkki Huhtamo [26] has presented the idea of "familiar aliens", where machines that have been portrayed and personalised in the media and everyday life have become part of our cultural tradition. So, the use of a large Bakelite knob, for example, as part of an interface for a musical instrument, aside of the ergonomics, is loaded with cultural references. In the post-digital, the significance, or our relationship towards this interface (the Bakelite knob), is different in relation to its original context. The post-digital musician seeks to find the new in the old.

8. EXCESS BAGGAGE

There is a synergy between developing a post-digital aesthetic in music and the need to perform and communicate on a global scale. The modular approach towards creating new musical interfaces discussed in this paper offers a convenient and practical solution to transportation. As already mentioned, David Tudor considered the benefits of small modular devices, and some examples can be found where issues of portability influenced his performance set-up. In an interview with Joel Chadabe, Tudor [10] in discussion on the piece *Untitled* comments: "The number '60' came about because there were sixty components involved, and I was not about to travel with sixty components to create the source material, so I recorded it and then subjected it to the feedback loop which was under my control, and that was one of my most enjoyable experiences." This demonstrates that Tudor had to think very carefully about his performance set-up as well as its transportation. The following quote from Lowell Cross [27], whom Tudor collaborated with on many occasions, provides enlightening information on some of the problems Tudor faced when transporting his equipment in 1973. The five bags Cross [27] refers to contained Tudor's "electronic gadgetry and cables": "When the time came for David Tudor to leave, I was dreading the prospect of helping him to carry his heavy footlocker-style cases and other baggage up the basement stairs so that I could take him and his belongings to the Cedar Rapids Airport, a half-hour drive from Iowa City. However, he announced that he was departing not from Cedar Rapids, but from the Des Moines Airport, 120 miles/200 km to the west. David Tudor was not by disposition an early riser, and when Nora and I heard his

[1] This is not the same definition of "something to hold on to" given by Leigh Landy [25] that refers to a musical parameter or reference.

travel plans, he was running precipitously late to catch a flight out of Des Moines. We hastily loaded the car and took Karen with us. We arrived in time for his flight, fortunately, and I began the chore of unloading his cases as David Tudor engaged a skycap. The man was taken aback by what he saw, and he was even more taken aback when he began to lift the first case onto his cart. We left David Tudor with that bewildered man and took a leisurely drive back to Iowa City."

Tudor had a major influence on the group of composers known as the Sonic Arts Union (Robert Ashley, David Behrman, Alvin Lucier, and Gordon Mumma) who also adopted a DIY modular approach to live electronics. Robert Ashely [28] remarked that: "A Sonic Arts Union concert was about 1,000 miles of wire and all these little boxes that plugged into each other". The economics of organising and performing concerts, of which transportation of the equipment must have been an issue, resulted in Robert Ashley taking the radical stance to stop performing and composing.

Air transport in general is a fundamental problem for musicians given that there are weight and size restrictions for baggage and there are growing security concerns over any baggage anomalies. Kaffe Matthews [29] in an account of her concert in Bologna in 2001 announces: "watch out, BOLOGNA (Forli) airport is 70 km away from Bologna and Ryan Air only allow 15kg [of] luggage." The current economic trends of the airline industry suggest there could be some serious financial implications for musicians travelling with the type of performance set-up discussed in this paper. As Toshimaru Nakamura points-out, the baggage allowance presently offered by many airline companies is also as much that can be physically carried by one person [Nakamura, personal communication]. It is not just the flight, but also transporting the equipment in general that is an issue. This idea will be discussed in more detail later. Nevertheless, some musicians tend to push the restrictions of baggage allowances to the limit, cramming every nock and cranny of their luggage with equipment. Some facts and forecasts regarding air transport and excess baggage follow.

The International Air Transport Association (IATA) - the governing body of international air transport - reported that in 2005 the air transport industry lost six billion pounds. This was mainly due to the rise in crude oil prices that rose by approximately a third from the previous year. Furthermore, fuel bills in the industry have doubled since 2003 and the industry has made a financial loss consistently since 2001. These statistics have forced the IATA to come up with a Fuel Action Campaign [1]. Reduction of fuel consumption, which is connected to the weight of the aircraft, is the main priority of the campaign. The rise in the cost of fuel is gradually beginning to reach the customer in the form of baggage restrictions and surcharges. In 2004 British Airways [30] introduced a fuel surcharge to its customers that has continued to rise. Whilst in January 2006, Ryanair [31] announced a charge for handling baggage. The pay for what you have/use approach adopted by many budget airlines, such as easyJet, will undoubtedly be applied to baggage in the future. The cost of a flight will depend on the number of kilograms to be transported.

If the cost of excess baggage increases or the weight and size allowance for baggage decreases, there will be an impact on musical interfaces that are currently used for live electronic performances. This in turn will indirectly affect which musicians can travel and the type of music that is performed. This might seem an exaggerated point, but these practical issues also are key in shaping broader musical aesthetics. Changes in excess baggage rules will have cultural repercussions.

9. DO NOT REMOVE COVER FOR RISK OF ELECTRIC SHOCK

Due to the necessity to travel light, 'scavenged' hardware and electronic sound generating devices have taken on a deconstructed look. The housing or casing of these devices is the first thing to go to reduce weight, leaving components and wires exposed within an exoskeletal frame. Function prevails over form. Redeployed hardware is customised and re-constructed in aluminum and fiberglass enclosures. The appearance of these interfaces resembles, for example, the architectural work of Richard Rogers [32]. Rogers, whilst discussing the Lloyds Building, London, states: "Whereas the frame of the building has a long life expectancy, the servant areas, filled with mechanical equipment have a relatively short life, especially in this energy-critical period. The servant equipment, mechanical services, lifts, toilets, kitchens, fire stairs, and lobbies, sit loosely in the tower framework, easily accessible for maintenance, and replaceable in the case of obsolescence." The modular system of electronic devices discussed here, part exposed, part insulated, takes on a futuristic look. There is an analogy to the body and its outer surface, the skin, being removed revealing the 'cyborg/android' inside: for example, these types of images are common in popular science fiction films such as *Terminator 2* [33]. In the case a laptop, the danger of electricity is disguised beneath a smooth external casing. With the aforementioned de-housed interfaces, the performer is taking serious safety risks. Further weight reducing techniques include replacing power transformers, particularly in older devices, with a lighter switch mode power supply. As already referred to, these interfaces extend beyond their housings to include wires and cables. Many cables also undergo surgery and are shorn and stripped down to exact sizes. Plastic cased plugs and 'direct wiring' are also employed.

10. CONCLUSION

Although air transport and excess baggage problems are a real concern for a musician, '32kg' can also be used as a metaphor. Performances of live electronics are often concerned with the broader subject of chamber music. These include, for example, small group work, improvisation, intimate performance settings, a focus on instrumental nuance and 'discrete' voices, and the idiosyncrasies of a performer or their instrument. This approach is dependent on a close relationship between performer and their instrument. Transportation of an instrument is therefore a primary concern. The modular system outlined in this paper allows for an instrument/interface to be on a 'human scale': something which can be held in the hand as well as being portable. This paper highlights how theory and practice are inextricably bound. An anything goes, hybridisation, approach is gathering momentum, questioning the very nature of musical interface.

11. ACKNOWLEDGEMENTS

The author would like to thank the support of the Music, Technology and Innovation Research Centre at De Montfort University, Sue Thomas for information on Blogwalks, and Toshimaru Nakamura for answering questions concerning excess baggage.

12. REFERENCES

[1] International Air Transport Association (IATA) industry facts. Available at: http://www.iata.org/

[2] Rucker, R., Sirius, R.U. and Mu, Q, ed. *Mondo 2000: A User's Guide to the New Edge*. Thames and Hudson, London, 1993.

[3] Negroponte, N. *Being Digital*. Hodder & Stoughton, London, 1996.

[4] Information on blogwalks. Available at: http://blogwalk.mediapedagogy.com/AboutBlogWalk

[5] Negroponte, N. Beyond Digital. *Wired, 6*, 12, 1998. Available at: http://www.wired.com/wired/archive/6.12/negroponte.html

[6] Cascone, K. The Aesthetics of Failure: 'Post-Digital' Tendencies in Contemporary Computer Music. *Computer Music Journal 24*, 3, 2000, 12-18. Available at: http://mitpress.mit.edu/catalog/item/default.asp?ttype=6&tid=2289

[7] Pepperell, R. and Punt, M. *The Postdigital Membrane: Imagination, Technology and Desire*. Intellect, Bristol, 2000.

[8] Lunenfeld, P, ed. *The Digital Dialectic: New Essays on New Media*. MIT Press, Cambridge Mass., c1999.

[9] Collins, N. Composers Inside Electronics: Music after David Tudor. *Leonardo Music Journal, 14*, 2004, 1-3.

[10] David Tudor interviewed by Joel Chadabe, Tomkin's Cove, 9/8/93. Available at http://www.emf.org/tudor/Articles/chadabe.html

[11] Max/MSP software. Available at: http://www.cycling74.com

[12] PD software. Available at: http://www-crca.ucsd.edu/~msp/software.html

[13] Haraway, D. A Cyborg Manifesto. In Trend, D, ed. *Reading Digital Culture*. Malden, Mass.: Blackwell, Oxford, 2001.

[14] Ghazala, R. Q. The Folk Music of Chance Electronics: Circuit-Bending the Modern Coconut. *Leonardo Music Journal, 14*, 2004, 96-104.

[15] Ghazala, R. Q. *Circuit-Bending: Build Your Own Alien Instruments*. Wiley Publishing, Inc., Indianapolis, 2005. Reed Ghazala website at: http://www.anti-theory.com/

[16] Bowers, John. and Archer, Phil. Not Hyper, Not Meta, Not Cyber but Infra-Instruments. *Proceedings of NIME 05*, Vancouver, BC, Canada, 2005, 5-10.

[17] Merzbow (Masami Akita) selection of edited interviews. In *Expanded Noisehands: The Noise Music of Merzbow*. Ed. Carlos M. Pozo. Available at: http://noiseweb.com/merzbow/all.html

[18] Tom Bugs website at: http://www.bugbrand.co.uk/

[19] Brian Duffy website at: http://www.warmcircuit.com/

[20] Wolf Eyes interviewed by Sam Frank and Ben Tausig. Available at: http://www.dustedmagazine.com/features/238

[21] Möslang, N. How Does a Bicycle Light Sound?: Cracked Everyday Electronics. *Leonardo Music Journal, 14*, 2004, 83-84.

[22] eBay website at: http://www.ebay.com/

[23] Machado, A. A Microchip Inside the Body. Originally published in English on the Nettime list, February 22, 1998. Available at: http://wearcam.org/kac.html

[24] Elsenaar, A. and Scha, R. Electric Body Manipulation as Performance Art: A Historical Perspective. *Leonardo Music Journal, 12*, 2003, 17-28.

[25] Landy, L. 'The Something to Hold on to Factor' in Electroacoustic Timbral Composition. *Contemporary Music Review, 10*, 2. 1994, 49-60.

[26] Huhtamo , E. Cybernation to Interaction: A Contribution to an Archaeology of Interactivity. In Lunenfeld, P, ed. *The Digital Dialectic: New Essays on New Media*. MIT Press, Cambridge Mass., c1999, 96-110.

[27] Cross, L. Remembering David Tudor: A 75th Anniversary Memoir. Available at: http://www.lowellcross.com/articles/tudor/

[28] Ashely, R. In Holmes, T. *Electronic and Experimental Music,* Routledge, London, 2002.

[29] Kaffe Matthews website at: http://annetteworks.com/

[30] British Airways fuel surcharge. Information at: http://news.bbc.co.uk/1/hi/business/4372439.stms

[31] Ryanair to introduce baggage charge. Information at: http://www.guardian.co.uk/uklatest/story/0,,-5571142,00.html

[32] Cole, B. and Rogers, R. E, ed. *Richard Rogers and Partners. Architectural Monographs*, Academy Editions, London, 1985.

[33] *Terminator 2: Judgment Day*, James Cameron, TriStar Pictures, 1991.

Meta-Instrument 3: a look over 17 years of practice

Serge de Laubier
Studios Puce Muse
2 rue des Pyrénées
Wissous 91320 RUNGIS Cedex, France
(+33) 1 45 12 04 53
sdl@pucemuse.com

Vincent Goudard
Studios Puce Muse
2 rue des Pyrénées
Wissous 91320 RUNGIS Cedex, France
(+33) 1 45 12 04 53
programmeur@pucemuse.com

ABSTRACT
The Meta-Instrument (M-I) is a new instrument for computer music. Since 1989, three backward compatible prototypes were designed involving custom developments in electronics, mechanics, computer science and music. This paper presents the third generation of M-I, and a few ideas that emerged from the past 17 years of playing it.

Keywords
Audio-graphic portable instrument, WiFi, Ethernet, Répertoire

1. GENESIS OF THE PROJECT
From 1983 to 1988, we worked at the Puce Muse studios , on the simulation of sound movement in a 3 dimensional space. This research led to the conception of the Octophonic Spatial Processor. This machine compute the sound level distribution over 8 or 16 loud-speakers, from cartesian or polar sound coordinates.

The loud speaker positionning can fit into various geometries: line, cube, circle...etc.

After several experiences, we use essentially, since 1988, a configuration with the loudspeakers placed at the corners of a cube. This cube fits into a space matching either the stage or the whole theater.

The various musical pieces composed with the system quickly highlit the link between the movement of sound in space, and its spectral movement. The nature of this « intern » Vs. « architectural » movement relation is a complex topic, since it covers several fields of knowledge: acoustics, music, and cognitive sciences.

Nevertheless, the musical fields opened by this -very stimulating- research faced a real problem, when adressing the conduction of sound. Indeed, if space is omnipresent metaphorically in music, musical instruments are not meant to move sounds in space. The question is thus to imagine a system able to simultaneously displace sounds in space while making their spectrum evolve.

The 2nd goal of the Meta-Instrument was to be able to play the « musique concrète » invented by Pierre Schaeffer[1], not for recording's sake, but to play it live in a concert. To link the revolution in electroacoustic techniques, that allow the musician to work with all recordable sounds, not to create musical pieces that remain fixed on sound-tapes, but find the ephemeral dimension of playing live music again.

2. HOW THE META-INSTRUMENT WORKS
These two questions led to the conception of a general system, made of three parts: gesture sensing, gesture transformation, and perception.

The Meta-Instrument main function is to catch the gesture. It is thus a transductor meant to measure precisely and digitalize the musician gesture. A first version was built in 1989 and still works today. A second generation, compatible with the first one, exists since october 1995. A third version, compatible as well with the previous ones, is operational since december 2004.

The Meta-Instrument is connected to an ethernet analogic interface. The 54 sensors data are sampled at 500Hz, with 16bits resolution.

This interface is then linked to a computer, where the gestural data are analysed and processed by programs developed with Max (© Cycling74/IRCAM) [2]. Today, there are more than 150 « software instruments » developed for various compositions. Each « software instruments » runs within a standard architecture called « bank », allowing switching and mixing management between the various « software instruments », driving sound, graphic or lighting systems.

3. THE SURPRISES OF PRACTICE
3.1 The pulp or finger intelligence
Essentially, the Meta-Instrument is a measuring system, with its tolerances and measurement errors. Three generations of them were needed to reach the gestural finesse of the fingers pulp. Indeed, each finger act simultaneously upon 4 keys, stimulated by longitudinal and lateral movements of the finger's pulp.

The minimal pressure measure is of 10 milligrams, and sampled every 2 milliseconds. This precision allows now to compute speed and acceleration accurately enough to link sound energy to gesture energy. This quality of measure gives the sensation of « seeing » the finger movements on the screen. As a comparison, the first Meta-Instrument measured only 10g or so every 20ms, with only one key per finger! Moreover the rigidity of the key used to give tenditinis.

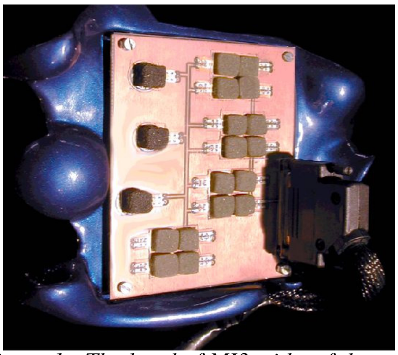


Figure 1: The hand of MI3 with soft keys

3.2 A wide range of possibilities

The number of possibilities given by the 2nd generation of Meta-instrument at every time sample is up to 2^{7x32}, or 2^{224}, which is about a billion times more than the number of atoms in the universe! This vertigo can be sensed when changing « software instrument ».

It happened at the beginning to the meta-instrumentist not to know anymore « where he/she is », and which musical space, which algorithm runs under the tip of the fingers. This kind of « blackout » disappear with the gesture memory linked to musical memory. Moving while hearing inwardly the musical software wakes up the gestural knowledge of the algorithm.

3.3 More conductor than instrumentist

In this temptation to play the orchestra-man, an attractive instrument is one producing broad and generous sound phenomenons, which surpasses power and tessitura's limits of traditional acoustic instrument. An instrument generating much more with less, as does the conductor baton with the orchestra.

Most of the software-instruments are closer to the conductor logic, than the instrumentist's. The relation with sound is often macroscopic, driving fluxes, speeds, orientations, probabilities... It is also in this direction that the possibilities are the newest, allowing interaction of sonic structures through phrasing and precision that have never been heard.

In this case, the gesture is not necessarily linked to the creation of acoustic energy, nor is it to the systematic choice of notes. For each musical idea, we have to decide on the gesture/sound mapping.

3.4 Position and variation gestures

Several software-instruments work with a double pinciple:

- Position gestures, which determines a more or less stable and quick selection in the algorithm.
- Variation gestures which , allowing for variations around the selected stable state, or “grosse note” as called by Pierre Schaeffer. [3]

In this duality between continuous and discrete behaviours, it seems that the Meta-Instrument fits the continous control better, allowing for arabesques, and roundness. The first corresponding more to keyboard instruments, the other to instruments like violin or voice.

3.5 Fast fingers, stable and precise arm

A software instrument can be controlled through many different ways. The same gesture can be mapped to any parameter. Although, in this seemingly endless space of mapping, there are some rules quite commonly followed. Fingers are mostly used for their quickness, they can run through the 64,000 values in about 10ms! The fore arm is much slower, but also more stable and precise.

3.6 The eye can amplify the hearing

Since 1991, the Meta-Instrument plays the lightings and since 1999, it controls computer graphics. The arrival of visual elements, in a musical context, sometimes encountered hostile reactions: “Music is enough itslef, graphics are here just to hide misery”

And yet, playing a musical instrument is also perceived by the eye. For exemple, the movement of a musican juste before starting, silence at the end before applause, are musical moments despite the absence of sound.

These disappear on audio recordings, when the audience cannot see what happens. Similarly, the score is a visual element which, when followed, greatly modifies the hearing. The experience of listening to a piano play, while watching all the mechanics is another exemple.

The Meta-Instrument always uses symbolic representation, in the software instrument, before producing sound.

It is thus possible to use the same parameters to simultaneously run visual algorithms, and to amplify the hearing, either by extending the notion of score, or by a representation of the algorithms.

Of course, interaction between the eye and the ear is a very complex topic, which goes beyond the scope of this paper. Let's just keep this exciting idea in mind: music is enlighting the shadows' light.

3.7 Silent instrument

The development of this work eventually gave birth to purely graphic software instruments. These instruments are interesting, because they show a very musical feature: the art of temporal variation. The musician then becomes a movement manipulator, “movement” meaning here the temporal evolution of an object, which can produce sound or something else.

These silent instruments also emphasize the closeness to dance, as well as the difference with it. In this case, the movement of the body is discreet, movements of the fingers' tips are only a few millimeters long, and these gestures need amplification to be seen. The instrument can thus be seen as an extension of the body.

3.8 Static and dynamic force feedback

The force feedback was particularly taken care of. Keys are soft, and continuously measuring pressure, from 0 to 200g, on a 3mm depth. The pressure roughly match an exponential perception of the touch. In the middle of each key, a small spot allows to feel where the center of the key is.

Also, all keys are covered by a soft material, so that fingers can move with ease. This refinement made us let the dynamic force feeback aside. Though research works such as ACROE's [4] really show hopefull results in this direction, dynamic feedback on the Meta-Instrument is only visual and acoustic, for the present time.

Figure 2: Static force feedback - springs & adjustable friction

3.9 After 15 years with crutches, the Meta-Instrumentist now walks!

Two previous generations of Meta-Instrument made use of stands. The third generation is mobile, with straps, and even wireless thanks to WiFi protocol.

This choice was made after considering the changing role of the Meta-Instrumentiste, from soloist playing in front of the audience, to the “Opera conductor” standing in withdrawal compared to the scene. The mobility offers the possibility to stand at the right place, which can change during the concert.

The goal is also to improve the contact with the audience: a musician will play differently when sitting or standing.

Figure 3: The MI3 is portable and wireless

4. WHY PLAY?

This question may sound weird, as we are used to think of music being played by musician. However, recording techniques deepely modified musical practice and the number of professional musicians plummeted over the last years. Is it necessary to mention the meaning or the “play” key, on tape and CD players?

As far as the Meta-Instrument is concerned, the question is all the more important that, beyond the instrumentist gesture, it is possible to record the whole score, with all the nuances, sounds and images.

4.1 An immense pleasure

Without any doubt, the first answer is the pleasure of playing a sound circulating faster than sound, along hundred meters. It is about manipulating images sized a few hundred square meters. More than a megalomaniac pleasure, it is about animating metaphors of the Creation.

4.2 Acting at the right moment

Here, the work is open, and allows to modify trajectories, and reshape forms at any time. Playing consists in finding this fragile, ephemeral, unique moment, and standing ready for the time passing, for the concert room, and of course, for the audience.

4.3 Putting the notion of Art work in question

Rather than “why play?”, the question could be “what to play?”. What is worth being phrased, and what can be automated? What interactions to play? These questions put the notion of art works in question. There underlie a definition of composition which could be “to create a space where to play”.

4.4 Playing to explore new musical spaces

The territories that have been opened by this practice are vasts. We are only at the beginning of discoveries. Many directions, like the research on the sound of gesture, or the sound of image, are still quite unexplored. The progress of this huge work will only be possible through the increase of the number of composers, musicians, developers, and teachers interested in this field of research.

5. FURTHER WORKS

At this stage in the evolution of the Meta-Instrument, and in order to spread its practice, several ways are being investigated.

5.1 Toward a plug'n play instrument

Due to the lack of computing ressources, the Meta-Instrument used to need a complex and heavy system, with a MIDI interface, a digital mixing console, samplers, lighting systems, and an octophonic sound system, along with one or more computers to analyse the gesture data and send control informations.

The Meta-Instrument 3 is now directly connected to the computer through Ethernet or WiFi protocol. The digital mixing console, samplers, and lighting/visuals have been integrated in the Max/MSP/Jitter layers of the software-instrument, thanks to the incredible evolution of computational speed.

The next step is to redefine and standardize an interface for the Meta-Instrument, so that using the software part will not require the musician to be a Max developer, and also to make available the pieces of code, which are constantly re-used, and emerged from long-time practice. In this goal, mapping tools such as IRCAM's MnM [5], and Physical Modeling for Pure Data (PMPD) [6] libraries are two noticeable effort which will ease the complex task of mapping.

5.2 Meta-Mallette: a collaborative alternative to the orchestral instrument

The heaviness of the system described above tended to discourage a teaching of the Meta-Instrument, that had been proposed in a few willing conservatories. Also, starting from scratch with the complexity of the Meta-Instrument may puzzle somehow the novice student.

In 2003, PuceMuse launched the development of a collaborative music system using algorithms similar (but simpler) to those used for the MI, played with joysticks – a cheap interface well known to children. Most of them just did not know that their virtuosity at playing videogames could be used for something else than driving race-cars at 1000 Mph or killing ennemies...

The Meta-Mallette (which fits in a wallet), introducing historical and new synthesis algorithms to the players in an entertaining musical activity, encountered a real success, giving a accurate response to a real need in music schools.

Several composers, musicians, and multimedia artists developed collective games meant to 8 to 30 players. Regular workshops have started to work with various social centers, music school and conservatories, and which enventually constitute a first step for the musicians to more evolved interfaces.

5.3 Workshops and growing community

Besides that, a one year long workshop has been launched in 2004 gathering 10 musicians to learn, and practice the Meta-Instrument along with collective consideration on the complex relation between sound, music gesture, image being controled all at the same time in real time.

The result was very fruitful, raising different approaches, ranging from perceptually relevant gesture/sound algorithms, to adaptation of electroacoustic pieces for live playing, and interactive cinema.

This experience led PuceMuse to build a few specimens, as research centers such as LaBRI[1] and LAM[2] ordered Meta-Instruments. It is hoped that the scientific collaboration will be improved by this direct use of the Instrument in the laboratories.

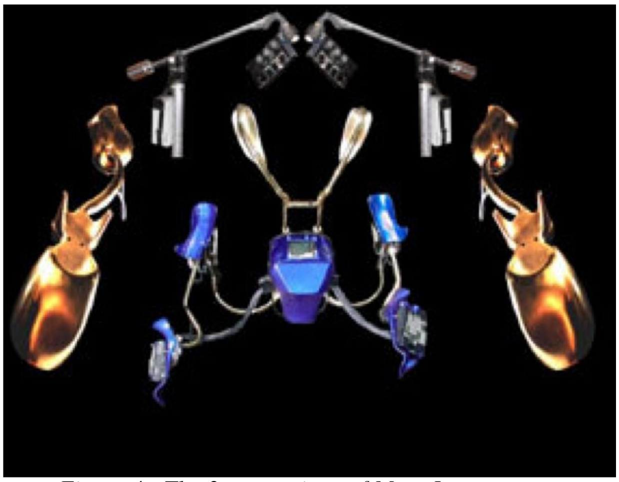


Figure 4: The 3 generations of Meta-Instrument

6. REFERENCES

[1] P. Schaeffer. La musique concrète. Presses Universitaires de Frances. Paris, 1967.

[2] Max/MSP/Jitter 4.5 reference manual. Http://www.cycling74.com

[3] Schaeffer, Pierre, Traité des Objets Musicaux, Ed. Seuil, Paris, 1966.

[4] C. Cadoz, A.Luciani, J.-L. Florens, and N. Castagné. ACROE-ICA: Artistic creation and computer interactive multisensory simulation force feedback gesture transducers. In *Proceedings of New Interfaces for Musical Expression (NIME'03), 2003*.

[5] F. Bevilacqua, R. Müller, and N. Schnell. MnM: a Max/MSP mapping toolbox. In *Proceedings of New Interfaces for Musical Expression (NIME'03), 2003*.

[6] C. Henry. pmpd: Physical Modelling for Pure Data. In *Proceedings of the International Computer Music Conference (ICMC' 04)* , pages 37-41, Miami, Florida, USA, 2004.

[1] Laboratoire Bordelais de Recherche en Informatique. http://www.labri.fr/

[2] Laboratoire d'Acoustique Musicale. http://www.lam.jussieu.fr/

The Case Study of An Application of The System, “BodySuit” and “RoboticMusic” -Its Introduction and Aesthetics

Suguru Goto
Instrumental Acoustics
IRCAM
1, place Igor Stravinsky
75004 Paris, France
+33 1 44 78 48 43
Suguru.Goto@ircam.fr

ABSTRACT
This paper is intended to introduce the system, which combines “BodySuit” and “RoboticMusic,” as well as its possibilities and its uses in an artistic application. “BodySuit” refers to a gesture controller in a Data Suit type. “RoboticMusic” refers to percussion robots, which are applied to a humanoid robot type. In this paper, I will discuss their aesthetics and the concept, as well as the idea of the “Extended Body”.

Keywords
Robot, Gesture Controller, Humanoid Robot, Artificial Intelligence, Interaction

1. INTRODUCTION
The system, which I introduce in this paper contains both a gesture controller and automated mechanical instruments at the same time. In this system, the Data Suit, “BodySuit” controls the Percussion Robots, “RoboticMusic” in real time. “BodySuit” doesn’t contain a hand-held controller. A performer, for example a dancer wears a suit. Gestures are transformed into electronic signals by sensors. “RoboticMusic” contains 5 robots that play different sorts of percussion instruments. The movement of the robots is based upon the gestures of the percussionist.

Working together with “BodySuit” and “RoboticMusic,” the idea behind the system is that a human body is augmented by electronic signals in order to be able to perform musical instruments interactively. This system was originally conceived in an art project to realize a performance / musical theater composition.

This paper is intended to introduce this system as well as the possibilities from my experiences in an artistic application.

2. General Description
This system is intended to be utilized in a project, which is entitled, “Artificial Body and Real Body.” The theme is to explore this dualism and the relationship between artificiality and reality of human body in a context of musical theater. Artificiality and reality sometimes seem to be conflicted with each other, but they can work together, or their meaning can be transformed for an audience depending on the context. The context provokes the audience to play with the ideas of reality and artificiality. A performance involving “RoboticMusic” and “BodySuit” challenges the audience by confusing the line between virtual and reality. I am a composer and intend to create this composition, which emphasizes the importance of performance aspects with this system.

This project originally started in 2002. This system is intensively experimented with and was shown on several occasions during 2005. The last performance was realized in, “Utopiales,” a festival in Nantes, France in November, 2005 (Fig.1).

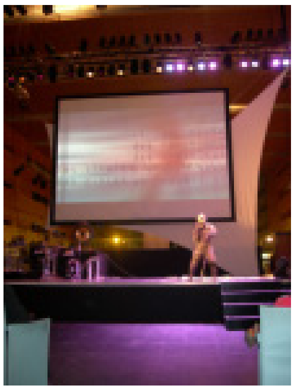


Fig.1: A performance with “BodySuit” and “RoboticMusic”. A photo from its rehearsal.

“BodySuit” was first created by an electronic engineer Patrice Pierrot, in 1997. Although it was originally conceived to work with “RoboticMusic,” it had to wait many years until “RoboticMusic” was ready. Meanwhile, many possibilities of “BodySuit” were explored, for instance, it was experimented with to control computer generated sounds and video images (Fig. 2).

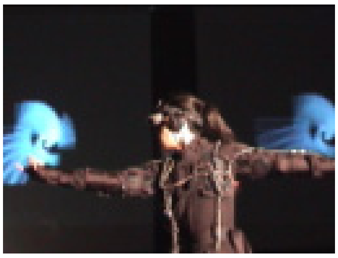


Fig.2: BodySuit can also control sounds and video images in real time.

"RoboticMusic" was created in 2003. The original concept and the design were done by me, and the robots were realized by a humanoid robot specialist, Fuminori Yamazaki, of the iXs Research Corporation in Japan. The project is still a work in progress, the goal is to eventually form a robot orchestra.

A gesture of performer with "BodySuit" is translated to gestures of "RoboticMusic." Instead of having a computer-generated sound, one can interactively have an acoustic percussion sound.

One of the important elements is the relationship and the communication method explored within this system. One may consider "BodySuit" and "RoboticMusic" as a relationship between a conductor and an orchestra, where dance-like gestures merely triggers instruments. In other words, this is an instrument that relies on physical gestures.

Another point is the method of translation used by the computer. For example, signals from "BodySuit" are transformed by Mapping Interface and Algorithm in a computer, and then are sent to "RoboticMusic." One gesture may trigger one attack on one instrument. However, it is also possible to trigger 5 instruments at the same time. Otherwise complex musical data, which is automatically generated by a computer and then reproduced by "RoboticMusic," is altered by gestures with "BodySuit" to modify the parameters of algorithm in real time.

3. Detailed Description of Percussion Robots – "RoboticMusic"

RoboticMusic contains 5 robots, which play percussion instruments, such as a Gong, Bass Drum, Snare Drum, Tom-Tom, or Cymbal. These instruments can be replaced as long as the replacement instruments can be played with Mallets (Fig.3).

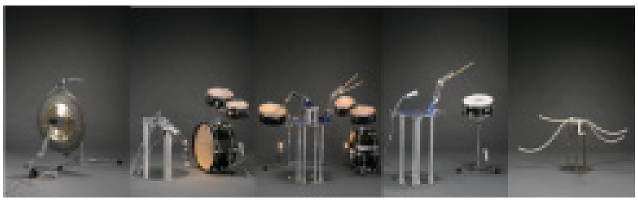


Fig.3: RoboticMusic. From the left to right, Gong, Bass Drum, Tom-toms, Snare, Pipes.

One of robots plays numerous pipes, and rapidly spins to create Flute-like sounds, which are generated while the air goes through them. These pipes are different lengths according to the pitches one desires. As it spins faster, the pitches become higher as following an overtone series (Fig.4).

Fig.4: Pipe Robot appears behind the Cymbal Robot on this photo. Pipe robot changes the pitches according to the speed of spins.

The latest technology of humanoid robots is applied to this, but "RoboticMusic" doesn't walk on two feet, nor does it contain eyes, a mouth, etc. "RoboticMusic" does contain robot's arms. The gesture of a human percussionist is modeled in order to have musical sound and expression. Yet a robot can perform without any rest, more precisely and faster than a human being.

Max, Cycling'74 is utilized as an interface and to generate musical data. With this, one can also send basic parameters to the robots, such as a position of the robot's arm, an offset position, intensity (how hard it hits) and so on. This sends the signals to another computer with Linux via UDP. A software in Linux is developed by iXs Research Corporation. This has an important roll, since it controls the movement of robot. From the computer with Linux to the robots, these are connected via USB. Each robot has its own interface, which is connected with an actuator and a sensor.

The robot has a special sort of springs to imitate a human muscle. Each holds a mallet at the end of his arm (Fig.5).

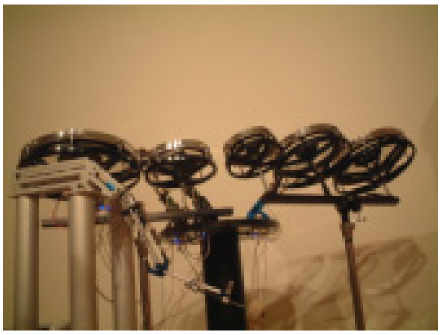


Fig.5: There is a special sort of springs in the arm of the robot. At the end of this, it holds a mallet.

The major advantage to "RoboticMusic" is that it interactively plays an acoustic instrument with the aide of a computer. There is no problem to play complex rhythm and it easily goes beyond the limit of human performance capabilities. Therefore, it gives new potentialities in a composition for acoustic instruments.

Another point is an acoustic sound. While a computer generated sound has many capabilities, an acoustic instrument has rich sonority and enormous possibilities of expression, especially from the point of view of a composer. When it is played on a stage, the vast possibilities of the acoustic aspect are obvious when compared to sound coming from speakers. Another benefit is that the audience may observe both sound and its gesture of performance.

To master one instrument is huge task for a musician, but to play together with others in an ensemble is another difficulty. Having 5 robots, one may extend the new possibilities of ensemble. For example, “RoboticMusic” allows 5 different tempos at the same time, or intricate accelerando and ralentando, but theses are exactly synchronized in music.

There is not only an artistic advantage with “RoboticMusic”, but also a research aspect. As one works more with a robot, which works with the gestures of a musician, one can discover how a human gesture contains complex movement, although it sometimes looks fairly simple, for instance, the gesture to hit a percussion instrument. A musician knows how to play an instrument, but he may find it difficult to explain exactly how he controls each part of his muscles and bones, and how he increases and reduces speed and intensity instinctively within a very short instant.

When one hears the word, “robot”, one is perhaps reminded of an industrial robot, or maybe sometimes a robot in a science fiction movie. However, it is not the case here. This is due to the latest development of artificial intelligence and is the case of application to a hardware. This has a lot to do with the robot, which performs instruments with a human-like gesture. In particular it refers to the humanoid type of robot that contains sensors and advanced programming, which the slave type robot in a factory, and at last we can profit from this in the field music. One may consider these robots as collaborators with humans.

4. Detailed Description of Gesture Controller – “BodySuit”

“BodySuit” has 12 sensors, which are placed on each joint of the body, such as a wrist, an elbow, a shoulder on the left and right arm an ankle, a knee, and the beginning of the left leg and right leg. The bending sensors are placed on the outer sides of the arms and on the front sides of the legs and fixed on a suit. Each sensor is connected with a cable to a box, and then it is connected with A/D interface.

A performer wears this suit, but doesn’t hold a controller or any instruments in his hands (Fig.6).

Fig.6: Upper half of body with “BodySuit.” The 12 bending sensors are placed on each joint of body.

Therefore, his gesture doesn’t have to be based upon playing an instrument, but could be liberated to become a larger gesture, like a mime. This allows for collaboration with a person in a different field, for instance a dancer or an actor.

The audience easily observes this larger movement. That is to say it can be well adapted to a performance and musical theater situation.

One may consider this as a body instrument. This efficiently works in a percussionist-like gesture. This is one of the best controller conjunctions with “RoboticMusic.”

Since this is not like a physical controller or instrument, which is held by hands, it allows to be collaborated with the idea, “ Augmented Body” or “Extended Body” in the work. His body is amplified by electric signals to control something remotely or to be extended from his abstract gesture to a meaningful gesture.

5. The system in “Augmented Body and Virtual Body”

The system of “BodySuit” and “RoboticMusic” is much explored within the last work, “Augmented Body and Virtual Body”(Fig.7).

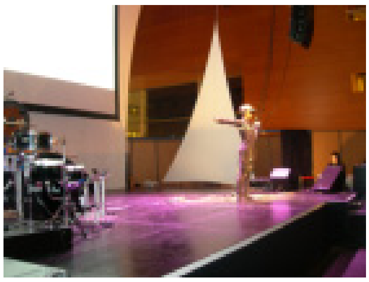


Fig.7: The performance of “Augmented Body and Virtual Body” in Nantes, France, in November, 2005.

In this work, the gestures with “BodySuit” are translated or altered by the algorithm in a computer, and then are sent to “RoboticMusic.” The interesting point is an idea of programming in order to alter one gesture to another state. For example, one single movement with a left elbow appears to be hitting a percussion instrument in the air. This triggers “RoboticMusic,” which plays 5 percussion instruments from left to right in a space with a gradual slight delay on each.

However, with the rising of a right shoulder, it changes the amount of delay to play in the order of left to right alternatively. The other case is one gesture translated into one gesture. A gesture with an arm in "BodySuit" is completely imitated by an arm of a robot, like someone copying another person. The method of communication has a great deal of importance here. In this sense, "BodySuit" and "RoboticMusic," two of them, but with a computer, three of them should be perhaps regarded as one system.

Since this is done in stage work, the fact, "gesture vs. gesture" should be much considered. The gesture with "BodySuit," which doesn't emit any sound by itself, is related with the gesture with "RoboticMusic," which is intended to create sound. In terms of interaction, the visual aspect between one gesture and another gesture provides clearer feedback and brings a different and interesting dimension. Incidentally, this communication does not refer to the one between the first person to another. In this case, the communication means the point of view from a third person to observe this relationship objectively.

6. Application of work

This system was applied to a performance / musical theater work in a composition (Fig.8).

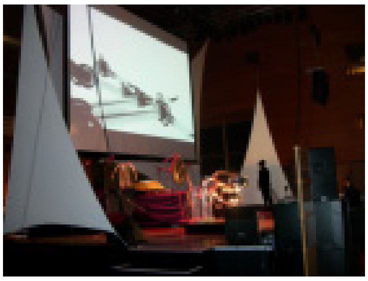


Fig.8: This system is applied to a music theater work. There is a 3D image (by Yann Bertrand) behind RoboticMusic. © Utopiales

At the end, this provides a lot of further and newer possibilities than I originally expected. While the communications of gesture with "BodySuit" and the gesture with "RoboticMusic" are observed, one notices different phases, which are the interaction and its perception, and the interaction with its consciousness. These two poles are the important keys in this field. With this articulated visual and oral experience in this work, one may cognate different experiences that constantly deal with something to expect, to understand, to notice, and to perceive.

Furthermore, the relationship between gesture and sound are also regarded by a very different view with this system. In other words, the idea, "music to see, visibility to hear" brings a different context in a theatrical performance.

While the concept, "Extended Body" was conceived to be realized with this system, the theme "Augmented Body and Virtual Body" is meant to question what a human body is and what his own identity is with this. Man and Machine seems to be dualistic, in which one may think that these are conflicted each other. Here, they coexist within this system, in fact, it is more correct to say that these are regarded as being one, "Extended Body". Therefore, our identity is not merely within our own body, but may communicate with outside and may be extended.

7. Conclusion

Historically, art has been always relating with a society where it exists, and has been always profiting from contemporary culture. As mentioned earlier, the robot here means the application of the latest development of artificial intelligence. Robotized instruments may have a lot of possibilities with this. At this point, this "Extended Body" is a reflection of society, especially where I grew up, in Japan. This is not an abstract image, but is just practically realized technically and aesthetically.

These technical possibilities and the aesthetic points have created further new potentialities by this system, however, this is not my only goal. While interacting with this system and myself, creating a new language and its perception are the most important goal in terms of a composition.

Furthermore, one may consider again that his own identity is not merely limited within his possessed body. While doing this "Extended Body", one may perhaps ask, if such an identity really exists or not, since there is not much limit or border line to extend. Music has to progress anyway, or it goes further by itself, whether we want it to or not. Likewise, a definition of music is limitless. Perhaps co-existence with this "Extended Body" may help to develop new possibilities in the composition of music.

8. ACKNOWLEDGMENTS

I would like to thank Fuminori Yamazaki (iXs Research Corporation), Patrice Pierrot, René Caussé (IRCAM), Alain Terrier (IRCAM), Instrumental Acoustics Department at IRCAM and Patrick Gyger (Utopiales) for their assistance in the realization of this project. I also thank Tom Johnson and Andrew Stewart for their useful comments on the manuscript.

9. REFERENCES

[1] iXs Research Corp. http://www.ixs.co.jp/

[2] Suguru Takashima : T Automated Performance Robots of Saxophone and Trumpet, JRSJ Vol.14 No.2, pp.196-199, 1996, Tokyo, Japan

[3] Suguru Goto: Virtual Musical Instruments: Technological Aspects and Interactive Performance Issues, IRCAM. Trends in Gestural Control of Music, 2000, Paris, France

[4] Suguru Goto: The Aesthetics and Technological Aspects of Virtual Musical Instruments: The Case of the SuperPolm MIDI Violin, Leonardo Music journal, Vol.9, 1999, California, U.S.A

[5] Cadoz, C. 1994. "Le geste, canal de communication homme/machine, la communication instrumentale." Technique et science informatiques 13(1): 31-61.

[6] Schnebel, D. Anschläge - Ausschläge, Text zur Neuen Musik. Carl Hanser Verlag:München, Wien, Edition Akzente Hanzer, pp. 37-49.

[7] Gilles Deleuze, Félix Guattari, Mille Plateaux, version of Japanese translation, Kawade Shobou Shinsha, 1980

Modal Kombat: Competition and Choreography in Synesthetic Musical Performance

[The First Ever Instrument-Controlled Video Game Battle]

David Hindman*
New York University, ITP†
721 Broadway, 4th Floor
New York, New York
david.hindman@aya.yale.edu

ABSTRACT

Public competition has been a part of the human experience for at least all of recorded history. The same is true for music. *Modal Kombat* is a live guitar competition channeled through the video game *Mortal Kombat*. The performance work builds on the idea of *dueling banjos*, one of the few examples of public musical competition. *Modal Kombat* takes this concept one step further by allowing a virtual mythological warrior to embody musical gestures. The result is a musical competition that utilizes advanced technology and popular media to exemplify the human passion for the spectacle sport.

This paper describes the technological and musical challenges that face performers, composers, and programmers when producing performance works that incorporate the control of pre-recorded graphical animations with audio and MIDI. Topics covered will include the musical instrument as an interface, choreography and competition, visual theatrics, performance flow, and compositional structures that allow for asynchronous rhythmic patterns inherent to and resultant from game controller input. In addition to its theoretical inquiry, the paper will discuss the practical artistic and technical needs required to successfully create a public instrument-controlled video game competition/concert, through documentation of *Modal Kombat*, the first ever instrument controlled video game battle.

Keywords

Video Games, Guitar, Interface, Instrument, Performance, Video Game Battle, Music Games, Sonic Games, Controlling Video Games With Sound, Modal Kombat, Mortal Kombat, Competition, Choerography, Human-Computer Interaction.

*Composition and Performance of *Modal Kombat* in collaboration with Evan Drummond of the Eastman School of Music

†Interactive Telecommunications Program

1. INTRODUCTION

Figure 1: *The Sonictroller (www.sonictroller.com)*, by David Hindman and Spencer Kiser, is a system that allows existing video games to be controlled by traditional, full-range musical instruments. *Modal Kombat (www.modalkombat.com)* builds on this system, extending its application in to the live performance setting as a hybrid musical performance and public video game battle.

Musical video game interfaces have existed for some time, but have not reached their full potential because they are game-specific and lack the versatility and tonal range of traditional musical instruments. With few exceptions, existing music gaming interfaces are nothing but toys and could never be used for music production apart from the single video game they were designed to control.[1] *Real* musical instruments have no doubt been overlooked as interfaces in the gaming industry because the market for such interfaces (skilled musicians) is much smaller than that of unskilled or hopeful instrumentalists. Also, interfacing a real musical instrument with a video game is far more complex than simply engineering a simplified instrument-like interface to be distributed with an accompanying game. This is why the most successful musical interfaces so far have relied on simple musical input, like the rhythmic interface for Nintendo's *Donkey Konga*, and single line melodic recognition

[1] *Karaoke Revolution*, by Harmonix, is an exception, it relies on the human voice for input. The interface for Nintendo's *Donkey Konga* also can be used as an acoustic instrument.

Figure 2: *Modal Kombat* in live performance. Two classical MIDI guitars control video game characters in the popular video game Mortal Kombat. The result is a hybrid musical concert and public video game competition.

or trigger recognition in Harnomix' *Karaoke Revolution* or *Guitar Hero*. As a result, practicing musicians have had to take step backward to participate in the musical gaming process. For a solution to this problem, I have created a system called *The Sonictroller*, a device which allows traditional, full-range musical instruments to control sprites in a wide range of console video games. I was motivated by the following questions: Why not have the option of playing a video game with a real electric guitar or bass? Or piano? Or drums? If anything, this option would engage current instrumentalists, and more importantly, it may create new ones. Furthermore, why limit a musician to a particular video game?

Figure 3: Shown above is the congo drum interface for Nintendo's *Donkey Konga.* It is the closest match to an real musical instrument gaming interface, as the device produces acoustic sound in additon triggering electronic events. The musical input is simple momentary controls in the form of beats, the game accepts no harmonic or melodic input.

Now, with the ability of concert-quality instruments to control video game sprites, we are witnessing the convergence of two previously separate forms of entertainment: the musical concert and the public video game competition. This paper focuses on the new performance genre that *The Sonictroller* has allowed to emerge, and describes the musical and technical considerations required to produce a work in this genre, building from the personal experience of creating *Modal Kombat*.

2. BUTTONS AND HANDLES

The content in this section is largely influenced by William Verplank, through a series of lectures at the Physical Interaction Design Workshop for Music at CCRMA 2004.[2]

Verplank describes how human-computer interfaces can be described as either **buttons** or **handles.** Buttons are momentary controls, switches, a brief change from one state to the next, a plucked string, a drum beat. Handles are always updating, continuously sending information, a long vocal melisma, a bowed violin. Musical sound is nothing but information; it can be listened to by the human ear for the purpose of pleasure, or it can be listened to by computers for the purpose of control. For example, a guitar can not only produce beautiful music for the human audience, it can also be a control panel, an array of 114+ discrete switches. A French Horn could be a mouse or joystick, volume on the x-axis and pitch on the y-axis.

Controlling video games sprites with musical gestures relies heavily on the concept of the musical instrument as valid human-computer Interface. When controlling the visual gestures of video game characters, I have found the knowledge of fundamentals of human-computer interaction to be invaluable in understanding how the musical gesture maps to the virtual physical gesture. At the heart of this lies the two main components of human-computer interaction: momentary and continuos controllers.

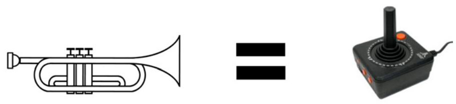


Figure 4: Wind Instruments, like the trumpet, are capable of acting as *handles*, continuous control devices because of their ability to output a constant stream of information.

Figure 5: Plucked or percussive instruments, like the MIDI piano or guitar shown above, are capable of acting as *buttons*, momentary control devices, because of their ability to trigger instantaneous, short-lived data. Note: Many instruments, such as wind or bowed instruments, can act as buttons or handles.

3. SHALL WE DANCE

The original goal behind the creation of the *The Sonictroller* was to create situations in which musical performers could compete through the control of video game characters. The most surprising by-product of this invention was the

[2]CCRMA: Center for Computer Research in Music and Acoustics. Physical Interaction Design workshop for music, July 2004.

discovery that the graphical animations of virtual characters could be re-purposed to make them dance, rather than fight, advance, shoot, conquer, race, etc. While the pallet of moves in this case is limited by the programmers intentions, there is no reason why a game character's moves can not be recycled specifically for the purposes of testing small or large-scale dance choreography. In this specific case, the single notes cause both characters to duck, so along with the audible staggering of pitches we see a visual staggering of characters kneeling and standing. The end result looks very much like the characters are dancing along *with* rather than *because of* the music.

3.1 Virtual Choreography

Movement I of *Modal Kombat*, entitled *Rain Dance*, makes use of sonically-driven virtual choreography through controlling the character named *Rain*. The peice begins with two guitars alternating single, staggered pitches, shown in Figure 6 and Figure 7. Each of these single notes triggers a single mommentary action of a corresponding video game character.

Figure 6: Above, the score for Movement I of *Modal Kombat* is shown. To start, staggered single notes between the two guitars trigger a choreographed sequence. Each character ducks on their respective beat, resulting in a dance-like effect.

4. COMPOSITION

While an unexpected outcome of controlling video games with instruments has been the effectiveness of the choreographed virtual dance, the main goal for this research was motivated by the idea of musical competition reflected in a visual feedback scenario. The challenge of creating a competition-based musical performance is mainly devising a compositional structure that allows for asynchronous rhythmic patterns inherent to and resultant from game controller input.

4.1 Harmonic Structure in the Musically Controlled Video Game Competition

Once the initial mechanism for mapping musical gestures to game characeters is in place, a greater challenge arises. The challenge is creating listenable music out of random streams of input. Since video game competition is by nature unplanned, the music in *Modal Kombat* is largely *harmonic contextual*, a series of tone areas that work well as

Figure 7: Shown above is the visual accompaniment to measures 1-4 of the score shown in Figure 6. As we can see, each character ducks when a single note is struck, resulting in a choreographed section resembling a tribal dance. This can viewed online at http://www.modalkombat.com/trailer.mov

a foundation for improvisation. Once a harmonic ground is established, parts of it can be looped- recorded live, and then improvised over. The improvisation of guitar solos over a harmonic ground drives the competition aspect of the performance. This way, the performance can be unplanned and different each time, two fundamental components of the video game competition. While random cacophony is nice at times, the challenge of this work is to create listenable music while still allowing the inevitable madness native to a public video game battle. The next section describes various mechanisms for establishing harmonic foundations on which to base improvised game play.

4.2 Technical Mechanisms for Maintaining Harmonic Continuity

There exist a number of different mechanisms for looping harmonic passages for the purposes of contextual improvisation. Hardware devices such as the Lexicon Jamman or Gibson Echoplex are popular in the looping community. *Modal Kombat* employs a custom performance patch written in Max/Msp that records live passages and then loops them under improvised solos. The patch, entitled *LiveLooper* is controlled with a MIDI footpedal and allows up to seven live overdubbs. Built-in to the patch include digital delay and separate solo and looping volume controls.

5. VISUAL THEATRICS: USING OPEN GL TO REMIX LIGHT AND ORIENTATION

Once a harmonic context is established for improvisation and a seamless program is implemented, we can add visual theatrics to the sonically-controlled video game battle. Cycling '74's Jitter library for Max/Msp provides an excellent way of adding extra visual dimensions to existing game graphics. Open GL gives us three-dimensional world in which we can render the two-dimensional graphics of older console video games. Using the OpenGL environment in Jitter allows built-in lighting and camera effects at minimal cost to processing speed. The following paragraphs discuss various OpenGL-based theatrical lighting and camera effects that can be controlled with sound, to enhance the visual experience of existing video games.

Figure 8: OpenGL allows 3-dimensional theatrical lighting effects to enhance the 2-dimensional space of the game Mortal Kombat. The lighting features in OpenGL allow for different instruments to generate different sonically mapped color explosions. In this figure, the volume from Guitar 1 generates red explosions of light.

5.1 Lights

Jitter's OpenGL functionality includes built-in lighting parameters for RGB color including light position, diffusion, color, and intensity. Movement II of *Modal Kombat* is based on sonically controlled theatrical lighting and camera effects made possible by Jitter's OpenGL. One obvious but effective mapping of audio in this movement is sonic amplitude to lighting intensity. As Figure 8 illustrates, it is possible to focus light on certain characters to create the illusion of the sprites performing under theatrical spotlights on a stage. Mapping the sonic amplitude to lighting intensity allows for effective explosions of focused light, that follow the many discrete levels of audio volume.

Equally interesting lighting effects include specific color mappings. In *Modal Kombat*, volume of each guitar generates certain colors: Guitar One generates red light while Guitar Two makes blue light. This is a simple but meaningful use of color to distinguish one competitor from the other, in the same way that teams in sporting events must where noticeably different colored uniforms.

5.2 Camera

Lighting is one feature of OpenGL that maps well from sound. Another component that we can use for theatrical effects is camera position. Some effects include setting the camera to zoom in on key harmonic figures or creating automated zooms to emphasize particular musical moments. Figure 9 shows a portion of an automated zoom placed at the end of Movement II of *Modal Kombat.* Automated zooms are useful for creating effects concerning the orientation of the two-dimensional screen. For example, figure 9 shows a the OpenGL window in the max patch for *Modal Kombat*, with the video plane tilted and camera at an altered z- axis, yielding the impression of the screen flying through the air. Note that the game charactrers are still fully functional- we can interface sound to all elements of screen orientation as well as to the actual content dislayed in the original game.

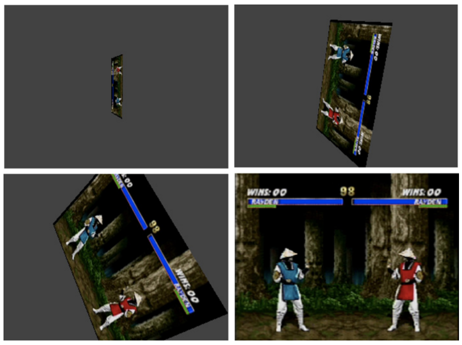


Figure 9: Camera positions can also be automated, to create the effect like the one shown above. The 2-dimensional plane flies through the 3-dimensional space, an effect inspired by the glass prison in the movie Superman 2.

6. CONCLUSIONS

The production of *Modal Kombat* has required an understanding of the musical instrument as a control device, a mechanism for this interface, thought-out compositional structures, well-planned performance transitions and remixed visual presentations. Certainly a video-game based musical production need not demonstrate all of the above qualities, but the success of *Modal Kombat* lies not simply in the ability to control video games with musical instruments but in the ability to use this technology to create large, seamless, well-planned performances. *Modal Kombat* has given careful consideration not only to just the technology that forms the foundation for its existence but to the musical language and presentation of this new genre of multimedia performance. This paper, in addition to serving as written documentation of the existing work *Modal Kombat* should also provide an overall outline of artistic and technical considerations for anyone choosing produce a work in this genre. Game Over.

7. ACKNOWLEDGMENTS

We would like to thank the following: Tom Igoe, Spencer Kiser, Chris Carrington, Jeff Gray, Mary and Jeff Hindman.

8. REFERENCES

[1] *Physical Computing*, by Dan O'Sullivan and Tom Igoe
[2] CCRMA Physical Interaction Design Workshop, Summer 2004, Lectures by William Verplank
[3] Midway Games, Inc
[4] Nintendo Co., Ltd.
[5] Harmonix Games
[6] Cycling '74, http://www.cycling74.com/

A "Ballet mécanique" for the 21st Century: Performing George Antheil's Dadaist Masterpiece with Robots

Paul D. Lehrman
Department of Music, Tufts University
Medford, MA, USA
+1 781 393 4888
paul.lehrman@tufts.edu

Eric Singer
League of Electronic Musical
Urban Robots (LEMUR)
Brooklyn, NY, USA
apps@ericsinger.com

ABSTRACT

An installation to perform *Ballet mécanique*, one of the most notorious works of the early 20th century, using acoustic instruments entirely under computer control, was constructed at the National Gallery of Art in Washington, DC to accompany a major exhibit on Dadaist art.

Keywords

Robotics, computer control, MIDI, player pianos, mechanical music, percussion, sound effects, Dadaism.

1 INTRODUCTION

George Antheil's 1924 *Ballet mécanique* for percussion orchestra, sound effects, and multiple player pianos, a composition which was never heard in its original orchestration until 75 years after its creation, is considered one of the major "lost" works of the early 20th century. The National Gallery of Art presented an opportunity to bring *Ballet mécanique* into the 21st century by inviting the authors to install a completely computer-driven orchestra to perform the piece, as part of a major exhibit on Dadaist art.

2 HISTORY OF *BALLET MÉCANIQUE*

Ballet mécanique was composed in 1924 by George Antheil, a young American composer and pianist living among the literary and artistic elite of Paris. His most outrageous work, *Ballet mécanique,* called for an orchestra of three xylophones, four bass drums, two pianists, a tam-tam, a set of electric bells, a siren, and three airplane propellors, as well as 16 synchronized player pianos.[1]

But the technology to perform the piece, and to link it to the film, actually didn't exist at the time, and so when it was performed, it was in a reduced version.

In the early 1990s, New York music publisher G. Schirmer, enlisted the aid of current author Lehrman to convert the player-piano parts in Antheil's score to a multitrack MIDI file, which could be played from a standard sequencer on MIDI-compatible player pianos.[2] In this form, the piece had its premiere in 1999, and has since been performed over 20 times in North America and Europe. It was published by Schirmer, with the MIDI files on CD-ROM, in 2003.[3]

3 DADA AT THE NATIONAL GALLERY

From 19 February to 14 May 2006 the National Gallery of Art was host to "the most comprehensive museum exhibition of Dada art ever mounted in the United States,."[4] Prior to coming to Washington, the exhibit appeared at the Musée national d'art moderne, Centre Pompidou, Paris, and after Washington it was scheduled to go to the Museum of Modern Art in New York City.

Stephen Ackert, head of the music department at the National Gallery, originated the idea of incorporating *Ballet mécanique* into the exhibit in Washington. His initial concept was for both an automated installation and a performance of the piece with a live orchestra. Unfortunately, it became apparent that the logistics of doing a live performance would be too complex, and so efforts were concentrated on the installation.[5]

4 CREATING THE INSTALLATION

Ackert and his design team scheduled the *Ballet mécanique* installation to be on view March 12-29. The location was the mezzanine lobby of the East Wing building, one flight above street level, directly outside the gallery hall in which the Dada exhibition took place (Figures 1 and 2).

4.1 Pianos

The score for *Ballet mécanique* has four separate player-piano parts, each of which is designated to be played on four instruments, for a total of 16. However, the piece can be (and has been) played on fewer instruments, as long as the number is a multiple of four.

Most previous performances utilized Yamaha *Disklaviers*, but Yamaha Corporation elected not to participate in the installation. Instead, QRS Music Technologies loaned 16 Gulbransen baby-grand pianos equipped with their *Pianomation* MIDI-controllable player system. The company also sent an identical

Fig. 1—End view of the Ballet mécanique installation at the National Gallery of Art.

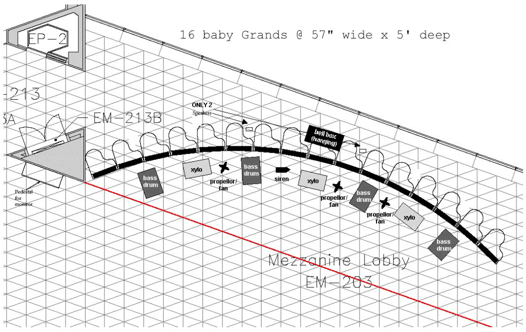


Fig. 2—Schematic diagram of installation. The entryway to the right of the pianos leads into the Dadaist art exhibition hall.

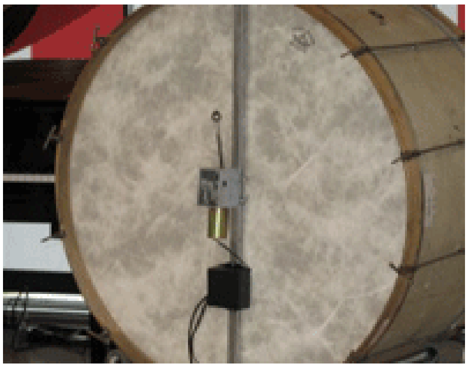


Fig. 3—Bass drum with BeaterBot.

instrument to Lehrman's home for for six weeks prior to the exhibition, to allow him to test and modify the sequence file, since the *Pianomation* system responds to MIDI slightly differently from the *Disklaviers*.

The two human pianists' parts were performed using two Kurzweil *MicroPiano* MIDI-controlled modules, amplified through JBL powered speakers. In order to blend the tone of the electronic and acoustic pianos as well as possible, Lehrman copied some of the human pianists' parts to the player piano tracks, taking care not to allow the parts to overlap (i.e., doubling notes) on a single instrument.

4.2 Other instruments

The Gallery's original concept was for the percussion and sound effects parts in *Ballet mécanique* to be provided by recordings, or by MIDI synthesizers or samplers, synchronized with the player pianos. But when Lehrman met with Ackert and the Gallery staff in Washington for the first time in October, 2005, he recalled seeing the work of current author Singer at NIME05,[6] and proposed that those parts be performed on real percussion instruments and mechanical noisemakers, played by MIDI-controlled robots.

Lehrman contacted Singer, who enthusiastically agreed to enlist his group LEMUR (League of Electronic Musical Urban Robots) to design and build the robotic players, installing them on rented percussion instruments.

4.2.1 Percussion

Most of the instrument mechanisms LEMUR constructed for this installation were based on the "BeaterBot" mechanism developed for LEMUR's ModBots. This is a microprocessor-controlled solenoid and lever mechanism used to move a beater at high velocity to strike a drum surface. The mechanism was used more or less directly for the bass drums and tam-tam, and adapted for use with the xylophones and propellors.

For the bass drums, they devised a cross-bar bracket to span one side of each drum. A BeaterBot mechanism was mounted on the bracket and fitted with a steel ball for a striker; steel was chosen because it produced a better attack transient than other materials tested, improving the audibility of the drums (Figure 3). Similarly for the tam-tam, a bracket arm fitted with a BeaterBot mechanism was mounted to the tam-tam stand, and a steel cylinder wrapped in suede was used as a striker. These materials produced the best combination of transient response and a sustained "blooming" of sound (produced by multiple strikes) that is an important part of a tam-tam's timbre.

LEMUR designed and constructed new robotic mechanisms to play the xylophones. They first considered a design using a small number of beaters which could move around to play different keys on each instrument. However, to achieve the playing speeds required by the score, they decided to use a separate beater for each key. This required 44 beaters per instrument (Figures 4 and 5).

To simplify construction of a large number of beater mechanisms, beaters were divided into sets. Each instrument had four sets of beaters, with two sets of 13 beaters each on the diatonic side and two sets of 9 beaters on the chromatic side.

Each beater within the set had an individual solenoid, pivot mechanism, and beater rod, with the pivot bars in the set mounted on a common shaft. The solenoids pull down on the pivot on the same side as the beater. This is in contrast to the original BeaterBot mechanism, in which the solenoid pulls from the opposite side in a standard lever fashion. This design modification was done for aesthetic reasons, so that the solenoids would hang below the keys, not stick up above.

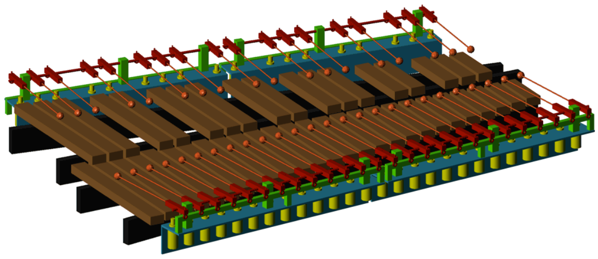


Fig. 4—Schematic drawing of XyloBot mechanisms.

The beaters are fitted with 3/4" Delrin® balls. This was found to produce the best tone out of several materials tested (e.g., other plastics, hard rubber), without damaging the wooden bars.

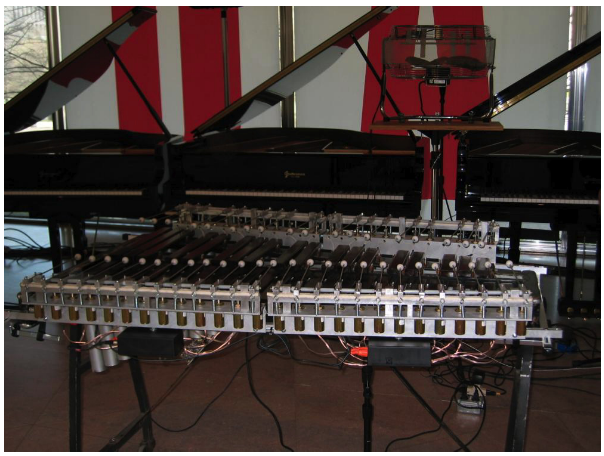


Fig. 5—XyloBot mechanisms on a concert xylophone. Propellor mechanism is in the background.

Each set of beaters is controlled from a custom PIC microcontroller-based circuit which receives MIDI note commands, maps them and converts them to timed signals to fire the solenoids. Velocity control is effected by controlling the gate time of the solenoids: within a certain set range of gate times, the shortest time will yield the minimum achievable strike velocity and the longest will yield the maximum. This range of gate times is determined experimentally. Firmware parameters are then stored in EEPROM which map note velocities 1-127 to this time range.

4.2.2 Propellors

As they were in Antheil's original performances, propellors are simulated using industrial fans. To create an appropriate sound, a piece of flexible material was inserted into the spinning fan blades—the "baseball card in the bicycle wheel" effect—at specific points in the score. Since the propellor sounds need to start and stop quite quickly, this was deemed more practical than using a fixed piece of material and turning the fans on and off.

To accomplish this, the team used a push-type solenoid mechanism fitted with a crossbar holding four .04" thick, 1-inch-wide strips of MDS-filled nylon (chosen for flexibility, durability and sound quality). When energized, the solenoid pushes the bar down, moving the nylon strips into the spinning blades. Again, several materials in various thicknesses were auditioned, with the final choice based on maximum volume of the mechanism and longevity of the strips.

The propellor parts call for notes to be held for long periods. Therefore, the intermittent-duty solenoid normally used in the mechanism was replaced with a continuous-duty model. A continuous-duty solenoid operates with a lower force (which is not an issue in this case) but may be energized for long periods of time without heat build-up and consequential damage to the solenoid.

4.2.3 Siren

The siren is an electric fire-engine-type wailing siren, running on 117 volts AC. It is controlled using a Mediamation LM-4 MIDI-controlled light dimmer. The three remaining outlets on the LM-4 are used to switch the propellor-fan motors, so they run only when called for in the score.

4.2.4 Bells

The score for *Ballet mécanique* is ambiguous about how many electric bells are required, but when Lehrman was preparing the files for Schirmer, they decided to make the number seven. For the Lowell premiere, with the help of engineer Coleman Rogers, Lehrman built a plywood "Bell Box" (Figure 6).

It is equipped with bells from 2" to 10" in diameter, obtained from various sources, with the largest being an old Radio Shack alarm bell, similar in appearance and tone to that found in schools and other public buildings.

Originally Lehrman intended to mount all of the bells on the front surface of the Bell Box, but he found when he did that all of the bells (except the largest, which has its own shock mount) sounded more like buzzers, as the plywood resonated louder than the bell gongs. The solution was to suspend the bells in free air: each bell was attached to a small piece of dense butcher-block wood, which was in turn hung using short chains from a pair of hooks at the bottom of the Bell Box.

Since the Lowell premiere, the Bell Box has been used for several other performances of *Ballet mécanique*, and there was no question it would be used in the Washington installation. Gallery personnel suspended it from the mezzanine ceiling with aircraft cable. It was noted that the ringing bell mechanisms produced very large transient voltages, which were damaging the DC power supplies originally used. Singer therefore rewired the unit, driving the bells with individual AC transfomers to improve reliability.

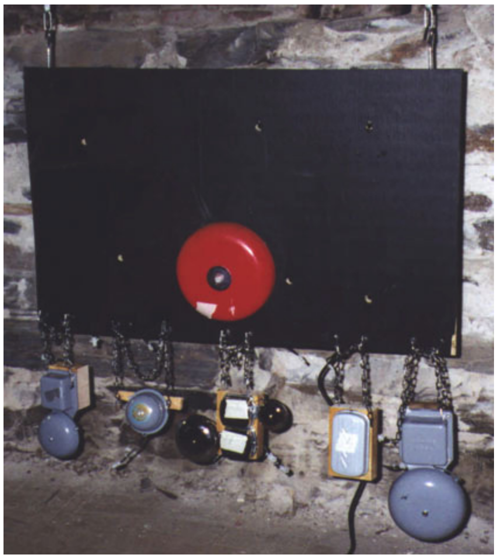


Fig. 6—MIDI-controlled Bell Box

4.3 MIDI Control and Networking

Ballet mécanique was performed by a Macintosh G5 computer running Mark of the Unicorn's *Digital Performer* software. The MIDI streams were generated by a Mark of the Unicorn *MIDI Time Piece USB*. Because of the density of the MIDI data, each group of pianos was assigned a separate MIDI cable, and the signal was daisy-chained within each group. The XyloBots all shared a single MIDI output from the *MIDI Time Piece*, with the MIDI signal being distributed via custom MIDI splitters and MIDI thru chains and each XyloBot responding to a different MIDI channel. Similarly, signals were distributed from other outputs to the other MIDI robotics, with each instrument responding to a specific MIDI note (Figure 7).

A Mark of the Unicorn *828 Mk II* audio interface (see below) supplied one extra MIDI output, which was used to drive the Kurzweil *MicroPiano* modules.

MIDI control of the Bell Box is effected by a MIDI Solutions R8 MIDI-controlled Relay Array. The low-current relays in the R8 are not sturdy enough to withstand the heavy currents drawn by the bells, so a secondary tier of relays was necessary.

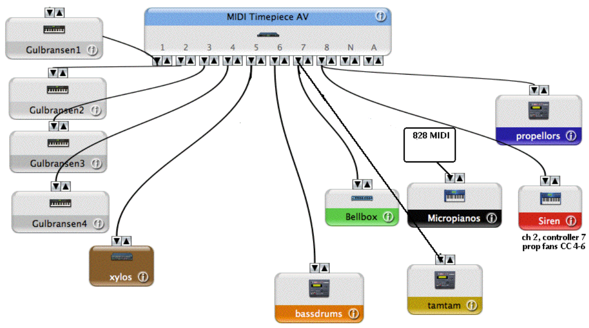


Fig. 7—MIDI network for the Ballet mécanique (Apple AudioMIDI Setup)

4.4 Testing and Modifying the Files

A complete performance of *Ballet mécanique* is 25-30 minutes long (depending on the tempo), but the National Gallery requested that each performance there be no more than ten minutes, so as not to frighten too many patrons. As Lehrman had edited the piece a number of times for several applications with the full cooperation of the publisher and the composer's estate, he was given a free hand to decide where the score would be trimmed.

Before any editing was done, it was necessary to determine how fast the piece was going to be played. Antheil's tempo markings are ambiguous, but the best interpretation says that he intended it to be played at about 150 beats per minute (bpm). However, this tempo is far beyond the capabilities of any live performers, and the fastest performance to date has been at 120 bpm. Absent the human beings, on the other hand, the limiting factor becomes the mechanical instruments, specifically the player pianos, which have the most complex mechanisms.

In empirical testing when he was preparing the original MIDI sequence for the Lowell performance, Lehrman determined that Yamaha *Disklaviers* could play the piece at 133 bpm, but at any faster tempo repeated notes would start to be skipped. Even to get the instruments to play at this tempo, the MIDI data had to be carefully massaged, with certain counterintuitive changes made to velocities and durations in order to get the desired response.

Lehrman's tests on the Gulbransen piano in his home showed that it could play the piece slightly faster—138 bpm—without a great deal of customization of the sequences, and so that was the tempo chosen. Note velocities in the sequence were set to a constant value of 100, and an initial Controller 7 (volume) command with a value of 127 was sent on each track. Lehrman then edited the score to the requested length, while attempting to preserve all of the thematic and orchestral elements that make the piece unique. Similar to the *Disklaviers*, the Gulbransen pianos have a built-in delay of 500 ms after receiving a MIDI command, so those tracks needed to be advanced by 500 ms.

After editing the sequence, Lehrman sent it to Singer, who tested the appropriate tracks on his instruments, He found that the solenoid instruments responded the best within a small range of velocity values, and although each instrument had its own optimum range, the best velocity for fast, repeated strokes was consistently 30% lower than for individual strokes. Lehrman modified the tracks to accomodate this. Singer also determined that the latencies in his instruments were insignificant and so no track offsets were required.

4.5 Amplification

As the installation was being completed on site, it became apparent that the airplane propellors and xylophones could not compete in terms of volume with the 16 grand pianos—especially with the latter's lids fully open, which is how the Gallery preferred to set them up. The Gallery was able to supply amplification, in the form of Shure SM57 microphones and JBL EON powered speakers, for these instruments. We were able to take advantage of the built-in mixing capabilities of the EONs, accomodating all of the necessary inputs while using only five speakers.

One unanticipated problem was that the noise from the fans when they were spinning would now be amplified even when they were not making the propellor sounds. Since the fans required a spin-up time of at least five seconds before each cue, amplifying their sound prematurely would significantly lessen their dramatic effect. This problem was solved using a Mark of the Unicorn *828 Mark II* audio interface and a Shure mic preamp. The signals from those microphones were sent to the 828's inputs, and in turn routed into audio channels within *Digital Performer*. Those channels were record-enabled with "full-time" monitoring, thus allowing the software to control the level of the signals passing through the *828*. Three audio tracks were added to the sequence which contained nothing but fader moves, timed to the start and end of each of the propellor cues. The signals from these tracks were sent to three outputs on the *828*, and from there to the JBL speakers.

6 CONCLUSIONS

The *Ballet mécanique* installation at the National Gallery of Art proved to be one of the most popular exhibits in that institution's history. Hundreds of listeners gathered for the twice-daily performances, and the *Washington Post*, in a highly laudatory review, deemed it "the best ten minutes of free fun in Washington."[7] Although the life of the installation was originally supposed to be 17 days, soon after the opening, the Gallery extended its run an extra six weeks, through May 7th.

We were delighted to have been asked to participate in this monumental undertaking. We believe that the spirit of the installation was faithful to the composer's intentions, extending and modernizing them in a way he would have been most approving of. Charles Amirkhanian, executor of the Antheil estate, flew in from San Francisco just to hear the opening-day performance, and pronounced it "Perfect." In addition, we were pleased to be able to bring this unsung composer's music to thousands of people who would otherwise have never experienced his unique vision.

7 ACKNOWLEDGMENTS

The authors wish to acknowledge the invaluable assistance of Stephen Ackert and the audio-visual, design, and electrical staff of the National Gallery of Art; Tom Dolan and the employees of QRS Music Technologies; William Holab; Charles Amirkhanian; Julian Pellicano; and Ron Frank.

8 REFERENCES

[1] Antheil, G. *Bad Boy of Music*, Garden City, NY, Doubleday, 1945.

[2] Lehrman, P., *The Revival of George Antheil's 1924* Ballet mécanique, unpublished Master's thesis, Cambridge, MA, Lesley University, 2000.

[3] Antheil, G., *Ballet mécanique*, New York, G. Schirmer, 2003.

[4] "National Gallery of Art—Current Exhibitions," http://www.nga.gov/exhibitions/index.shtm (11 April 2006).

[5] "National Gallery of Art to Premiere Unique Version of Film Score," http://www.nga.gov /press/exh/202/filmscore.shtm.

[6] Singer, E., "A Large-Scale Networked Robotic Musical Instrument Installation," Vancouver, BC, Proceedings of the conference on New Interfaces For Musical Expression, 2005.

[7] Kennicott, P., "'Ballet Mecanique': Dada's Siren Song-A Bravura Staging of George Antheil's Classic Brims With Mirth and Meaning," Washington, DC, *Washington Post*, March 20, 2006, Page C1.

Using the augmented trombone in "I will not kiss your f.ing flag"

Serge Lemouton
Ircam-CGP
1, place I.Stravinsky
75004 Paris
serge.lemouton@ircam.fr

Marco Stroppa
Hochschule für Musik und Darstellende Kunst
Urbanstr. 25
70182 Stuttgart
+49 (0)711 2124655
stroppa@marcostroppa.com

Benny Sluchin
Ircam-EIC
1, place I.Stravinsky
75004 Paris
sluchin@ircam.fr

ABSTRACT
This paper deals with the first musical usage of an experimental system dedicated to the optical detection of the position of a trombone's slide.

Keywords
Augmented instrument, trombone, performer–computer interaction, chamber electronics.

1. MUSICAL MOTIVATION
When planning his work "I will not kiss your f.ing flag"[1], Marco Stroppa expressed his interest in realising a gestural interface adapted to the trombone for several reasons :

- to search for a more musical, intimate relationship between an acoustical instrument and a complex electronic environment than offered by conventional interfaces and techniques based on the detection of a signal sent through microphones (a MIDI trombone does not exist!)
- to trigger events and control electronic parameters also when the performer is not playing, only by moving the slide (instead of using a set of pedals)
- to use the position of the slide as a source of information for a score follower, in order to improve its reliability and allow for the detection of silent movements
- to eventually use data extracted from the gesture of the performer to control some global features of the electronics, such as the overall dynamics, or the rise time of transients

In addition, since the musician will stand in different places on the stage during the performance of the work, the system has to function wirelessly.

2. SENSOR FOR THE SLIDE
The current version of the device employed to sense the position of the trombone's slide consists of :

- a source of red laser light fixed on the moving part of the slide (figure 1)

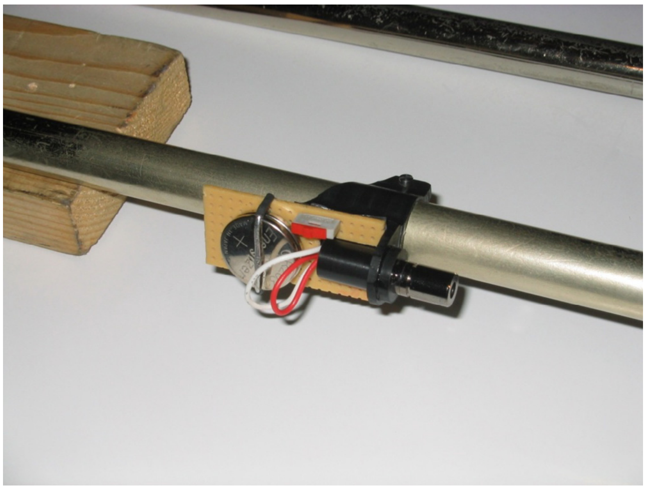


Figure 1. Laser source

- a receiver made up of two photo-electric diodes on the fixed part (figure 2)

The laser is powered by two lithium coin cell batteries and is supported by a snugly fit, hand-designed plastic ring. The diodes are fixed on an aluminium bar fastened to the trombone with a caoutchouc toric joint. They are connected to a Wise Box and send values to the computer every 5 milliseconds. The Wise Box, developed at IRCAM, is a multi-performer wireless sensor interface using WiFi and OSC [1].

The surface and intensity of the light spot received by the diodes varies as a function of the distance of the laser source (figure 3).

An accurate measurement of the position of the slide requires that the angle and focus of the laser be carefully adjusted.

[1] for augmented trombone and chamber electronics, commissioned by the Wittener Tage 2005, and premiered in Witten, Germany, on April 21st, 2005, with Benny Sluchin, trombone, Serge Lemouton, musical assistant and Jérémie Henrot, sound engineer.

Figure 2. Diodes

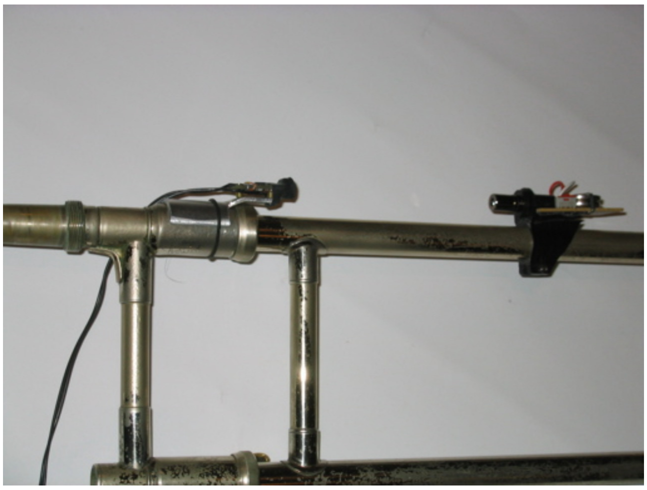


Figure 3. Slide with the position sensors

3. SYSTEM CALIBRATION

In order to map the values sent by the Wise Box to the positions of the slide, the system has to be calibrated[2]. We chose a scale of 7 values corresponding to the 7 positions of the Bb trombone[3] [3].

The calibration is done with a Max patch (figure 4) : while the performer plays a note for each position, the current value of the sensor will be tracked and a breakpoint function will be then generated for the conversion (figure 4 : on the x-axis the sensor's value, on the y-axis the scale of the positions).

This system allows for a precise, continuous detection of the positions of the slide with a reasonable spatial resolution.

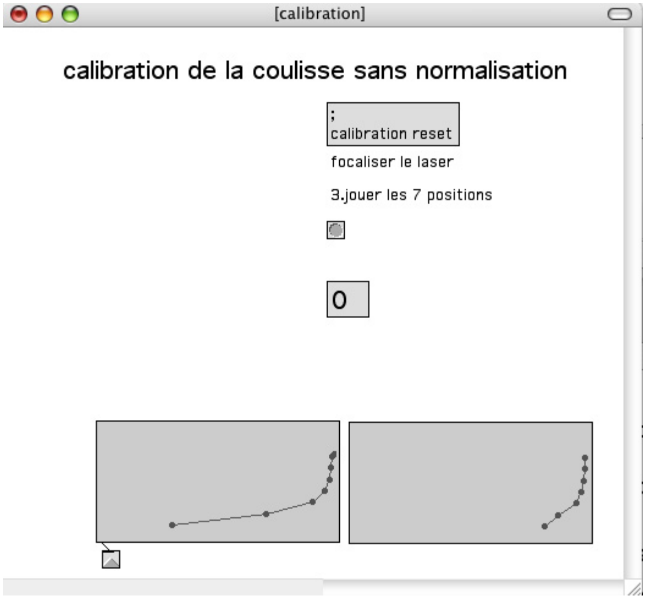


Figure 4. Calibration of the slide in Max

4. THE SLIDE AS A CONTINUOUS CONTROLLER

In several movements of "I will not kiss your f.ing flag", the position of the slide is used to continuously control some electro-acoustical parameters. For instance, in the second movement the length of the slide controls the centre frequency of two formantic filters acting on pre-recorded trombone sounds similar to those played by the performer at the same time. This is a kind of simulation of the wa-wa effect controlled by the slide (figure 5).

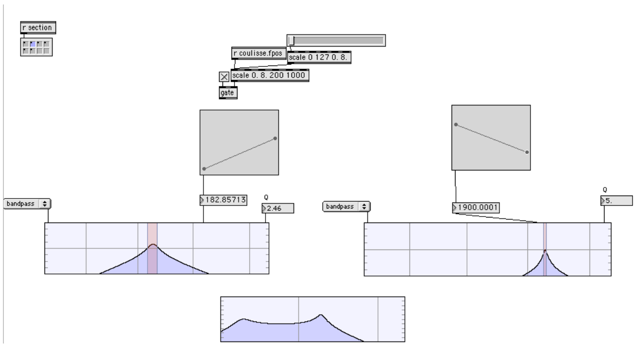


Figure 5. The slide controlling two filters

In the fifth movement, the slide is used to control the frequency of a frequency shifter, resulting into inharmonicity that corresponds to the frequency modulation

---

[2] at least each time it is turned on, often after some time as well, since the sensors can be sent slightly off position by the jerks of the instrument while playing

[3] cf. www.yeodoug.com/resources/faq/faq_text/slidechart.html

naturally produced by the musician singing into the instrument while playing (figure 6).

Figure 6. Excerpt of the trombone part, mov. 5

The fact that in the normal technique of the instrument the length of the slide is not necessarily correlated to pitch (longer slides can generate higher pitches than short slides, depending on which harmonics are selected) adds to the musical interest of this process, by avoiding too a stereotypical relationship between the position of the filters or the amount of modulation and the perceived result. Moreover, in the latter example, the amount of inharmonicity originating from the system depends on the relationship between the position of the slide and the pitch being played. Simple controls thus yield complex results. This is consistent with the nature of the music and the role of the electronics in this movement.

Looking for this kind of interaction between the controls of the electronics and the instrumental material has been one of the main compositional challenges the composer had to face.

5. POSITION FOLLOWING

The seven positions are approximately 10 cm away from each other. As a matter of fact, for the same musical interval, this distance is variable. A semitone in the first position is approx. 8 cm long; the size then augments progressively until the last position, where a semitone corresponds to approx. 11 cm.

We have tried to use this information when designing a score follower for the discrete positions of the slide. At first, Benny Sluchin "fingered" the score of the first movement, a very rhythmical music, with several controls triggered by a silent slide (figure 7).

A discrete position was detected as soon as the slide stopped its movement, using a simple threshold on the first derivative of the position.

Finally, after several tests, the score follower only based on the gesture of the performer proved to be not reliable enough for several reasons :

- The gesture of the trombonist's left hand is not so simple as one might imagine. The hand does not precisely stop on the position [2], but is often adjusted to that position.
- The wrist also plays an important role : supple and flexible, it acts as a spring to absorb the motion of the hand.
- Furthermore the position itself is slightly modified, in order precisely adjust the tuning of some notes of the harmonic series.
- The values given by the sensors, although very precise, are subject to too much fluctuation to generate the absolute values of the position of the slide for every circumstances.
- The high positions are especially difficult to track, since the receiver is farther away from the light source, and is too sensitive to changes of the ambient light and to small lateral displacements of the slide.

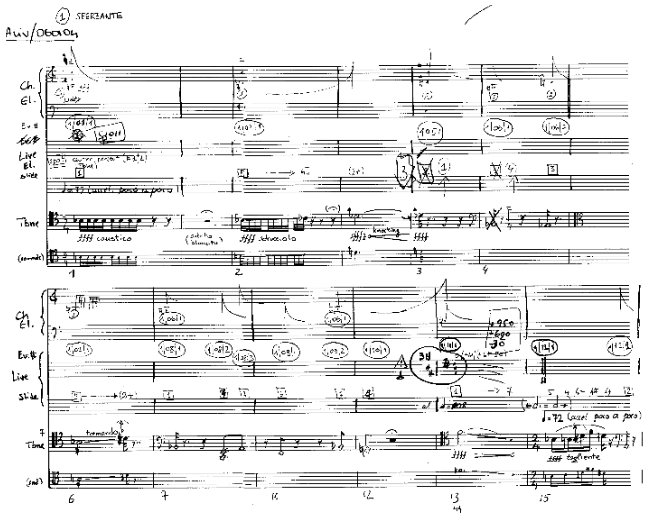


Figure 7. Beginning of the piece

Benny Sluchin suggested then to add a switch close to the mouthpiece, played with the left hand, to indicate the position changes (figure 8).

This switch makes the high rhythmic precision required in the first movement easier to perform.

Figure 8. Complete instrument with the switch

6. CONCLUSION

Given the lightness and simplicity of the prototype, the global results can be esteemed as satisfactory, both on a technical and a musical standpoint.

Several improvements to the current system could however be imagined, among them :

- The reliability of sensor itself might be improved, by either finding a better laser light, or imagining other sources.
- The algorithm to detect the positions ought to be made more precise and to use more data from research into the actual gesture of a performer's hand.
- The score follower should use as much information as possible to achieve a successful detection and not be limited only to the position of the slide.
- The whole system must function for a longer amount of time on a light battery before running out of energy.

7. ACKNOWLEDGMENTS

Our thanks to Emmanuel Flety, Frédéric Bevilacqua, Nicolas Leroy, Alain Terrier and the production team of IRCAM for their contribution to the realisation of this prototype.

8. REFERENCES

[1] The WiSe Box : a Multi-performer Wireless Sensor Interface using WiFi and OSC - Emmanuel Flety - in Proceeding of NIME 05, Vancouver, Canada, 2005

[2] Mark Kruger et al. : "Music Performance Physiology: Development of Expertise in a Natural Task: The Right Arm of Trombone Players" (http://music.utsa.edu/tdml/conf-III/III-Kruger-et.al./III-Kruger-et.al.html)

[3] Benny Sluchin, Contemporary Trombone Excerpts, A Practical Introduction to Contemporary Trombone Techniques, EME, Paris, 1995.

On making and playing an electronically-augmented saxophone

Sébastien Schiesser
Laboratoire d'informatique, acoustique et musique (LIAM)
Faculté de musique, Université de Montréal
200, avenue Vincent d'Indy
Montréal (Québec), H3C 2J7
Canada
sebastien.schiesser@a3.epfl.ch

Caroline Traube
Laboratoire d'informatique, acoustique et musique (LIAM)
Faculté de musique, Université de Montréal
200, avenue Vincent d'Indy
Montréal (Québec), H3C 2J7
Canada
caroline.traube@umontreal.ca

ABSTRACT

A low-tech electronically-augmented saxophone has been developed with a modular palette of sensors and their corresponding control interfaces. This paper describes the modules and proposes mapping strategies derived from a reflection on the various uses of live-electronics and an analysis of the functions of gestures applied to the saxophone. It also discusses the functional mutation of the performer's gesture induced by the electronic augmentation of an acoustic instrument.

Keywords

saxophone, augmented instrument, live electronics, performance, gestural control

1. INTRODUCTION

A plethora of new interfaces and gestural controllers have been presented these last years, as the high quantity of papers and conferences on this subject tends to show. Nevertheless, the amount of musical works written for these new instruments remains dramatically low in comparison. Many of these new instruments will never overcome the prototype stage. In fact, there is generally a wide gap between the invention of an instrument and its acceptance from a broader audience, depending on many technical and socio-economical factors. For example, the saxophone – which first appeared in public in 1842 and was lauded by Berlioz [10] – has become one of the most popular instruments in jazz and entertainment music, although it is still maverick in classical and contemporary music.

The current project was based on the idea to develop a simple (*low-tech* and *low-costs*) electronically-augmented saxophone (as defined in [16]) and to exploit its potential to compose and perform music. Comments and ideas from composers have influenced the development of this project from its very beginning. Similar to J. Impett's Meta-Trumpet [9], M. Burtner's Metasaxophone [4] or C. Palacio-Quintin's Hyper-Flute [12], this saxophone is augmented with sensors, where each device may be used independently from one another. Since this work is still in progress and only few experiences have been collected in concert situations, this paper prepares the ground for new compositions and further performances outcomes.

The long-range goal of this project is to develop a toolbox with many *modules* – *i.e.* sensors and their corresponding control programs – and play pieces from this young and growing repertoire for saxophone and live-electronics. According to the needs of the composition and the performance (gestures, staging, ability of the performer), the best combination of modules should be found for each piece. We also plan to develop a completely portable system (wireless microphone and interface) which will allow us to explore some interesting staging possibilities made possible by the greater freedom of movement that will be gained.

In the next section, as a starting point for the description of our electronically-augmented saxophone, we define the gesture parameters on an acoustic saxophone. Then, in section 3, we describe the configurable set of sensors that can be added to the saxophone and we provide some examples of mapping. Before concluding with an overview of potential and further developments of this instrument, section 4 proposes a general reflection on the various issues and mutations in the performer's control and practice induced by the electronic augmentation of a traditional acoustic instrument.

2. ON PLAYING AN ACOUSTIC SAXOPHONE

2.1 Functional levels of gestures

The performer's gestures can be categorized in three functional levels [5] [7]:

- *Effective* or *instrumental gestures* – necessary to produce the sound, *e.g.* blowing into the mouthpiece, closing and opening keys, etc. The initial conditions of these gestures (*e.g.* pressure exerted on the reed before blowing) may be considered as *biasing gestures* [16].
- *Accompanist* or *ancillary gestures* – body movements associated with effective gestures, e.g. inclining the instrument;
- *Figurative* or *sonic gestures* – perceived by the audience, but without any clear correspondence to a

physical movement.

The instrumental gestures can be sub-categorized in three functions:

- *Excitation gestures* – that provides the energy, *e.g.* plucking a string;
- *Modification gestures* – related to the modification of the instrument's properties, *e.g.* modulating the air flow to produce a vibrato;
- *Selection gestures* – choosing among multiple similar elements in an instrument, *e.g.* choosing a fingering to produce a given pitch.

Performers always aim to refine the gestural control of their instrument. After years of practice, they generally loose awareness of some movements that have become reflexes. Many tactile or kinaesthetic perception – touch sensitivity of the skin, position and orientation of limbs and other parts of the human body, etc. [3] – are so deeply internalized that any change in an instrument has to be domesticated through hours of practice. For example, changing from the alto to the baritone saxophone requires hours of adaptation, although these two instruments have the same mechanism.

As we will see in section 4, the electronic extension of an acoustic instrument such as the saxophone induces a redefinition and a functional mutation of some types of gestures.

2.2 Case study: the saxophone

Analyzing gestures on a saxophone is a difficult task given that gesture parameters within and across functional levels are often strongly interdependent. Here is a non-exhaustive list of gestural parameters involved in saxophone playing :

Embouchure – *excitation, modification*
The embouchure – here defined as the vertical jaw pressure combined with the round-shaped lip pressure on the mouthpiece – is one of the necessary parameters to produce tone[1]. It also plays an important role as a modification parameter in the production of vibrato and timbral nuances, from sub-tones[2] to very bright sounds.

Tongue – *excitation, modification*
The position and movements of the tongue play a role in the excitation and the modification of a saxophone tone. The position of the tongue in the mouth influences timbre and its back-and-forth movements can be used to modulate the air flow in order to produce a vibrato.

Throat – *excitation, modification, selection*
The degree of opening of the throat and larynx directly determine how the air flows to the instrument, enabling sound production, as well as affecting timbre and selecting pitch. The influence on pitch selection is demonstrated by the possibility to play partials of a given note, without changing fingering but rather by modifying the air flow.

[1]If an embouchure is required to produce tone, it does not obligatory play a role to produce sound, since percussion or squealing effects can also be produced without any embouchure.

[2]Sub-tones are defined here as mellow tones containing very few partials above the fundamental frequency.

Breath pressure – *excitation, modification*
Besides its role in the excitation gesture, the breath pressure can be varied to produce vibrato, in the same manner as for the flute. Any lack of stability will also cause intonation problems, particularly in the upper register.

Fingering – *modification, selection, accompanist*
On the saxophone, fingering mainly determines the pitch of the note and is thus considered as a selection gesture. But since each fingering also induces structural and timbral changes in the instrument, it can be seen as a modification gesture as well [8]. The accompanist function is not obvious, but is sometimes specified on the score by composers, who can ask, for example, to use the right hand (instead of the left hand) to play on the upper part of the instrument[3] or making demonstratively large finger movements.

Body movements – *modification, accompanist*
The sensation of the body's center of gravity is directly linked to the stability of breath pressure and therefore influences sound parameters. Accompanist gestures are composed of all the expressive movements made by a performer while playing, as well as some parasitic movements due to tensions in the upper part of the body, such as shoulder or elbow twitches. The performer is not always aware of these movements and does not usually control them precisely. Nevertheless, some specific controlled accompanist gestures are sometimes required by composers to affect the perception level of a performance[4].

3. ON MAKING AN AUGMENTED SAXOPHONE

In this section, we present the toolbox of sensors we have developed for the saxophone. Sensors are classified in terms of output modalities.

3.1 Interface and control patches

We built a simple microcontroller-based interface to read signals from digital and/or analog sensors and send them to a computer[5] (see Figure 1). The current version of this interface reads up to six analog channels on 10 bits (1024 values), communicates to the computer *via* USB and is recognized as a six degrees of freedom (DOF) joystick. It can be used in Max/MSP 4.5 with the *hi* (human interface) object. Further improvements will include the addition of digital ports in order to make the analog ports available for continuous signals. We are also planning to implement wireless communication to the computer.

A collection of Max/MSP patches were programmed to process the data generated by each type of sensor. All sensor/patch units – from now on called *modules* – are autonomous and independent. They can be used or not in

[3]In Luciano Berio's *Sequenza VIIb* for soprano saxophone, some left-hand trills are proposed by Claude Delangle with the right hand, first to play faster, but in a acting way as well.

[4]Karlheinz Stockhausen's *In Freundschaft* defines three melodic levels, emphasized by three body positions of the performer. It also requires back and forth movements while playing.

[5]For more information on the AVR-HID, please see the project web-page at `http://www.music.mcgill.ca/~marshall/projects/avr-hid/`

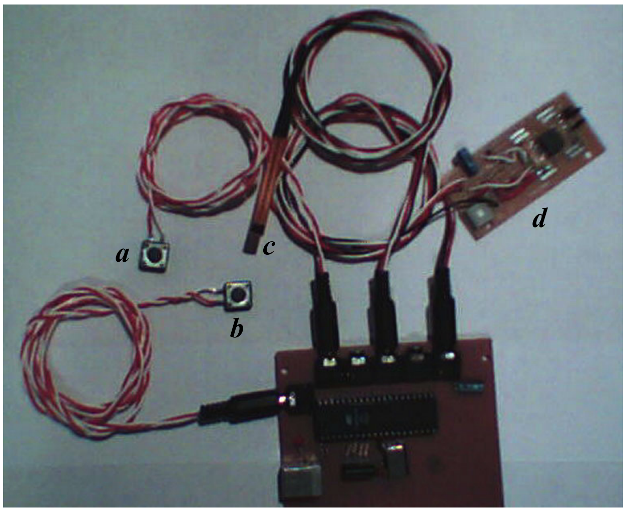


Figure 1: Microcontroller-based interface with different sensors: *a), b)* push-buttons, *c)* FSR, and *d)* inclinometer with scale adjustment electronic and low-pass filter.

a particular set-up, depending on the specific needs of the piece to be played.

Each module comes with a specification chart (see Figure 3) which defines the hardware and software parts, the number and the type of usable parameters, as well as the range, name and type of inputs and outputs for an implementation in a higher-level Max/MSP patch. Figure 4 shows a Max/MSP patch for video-tracking with a webcam as input.

3.2 Sensors and mappings

The sensors used with the interface previously described can be categorized by output modalities – *i.e.* the ways humans physically control things [3] – rather than by the physical energy they measure. In the present case, two modalities are used:

- *Muscle action* – movements or pressures induced by body parts (see Figure 2)
- *Sound production* – acoustic parameters of the emitted sound

Muscle actions can be detected by various types of sensors since they can produce changes in many different physical quantities, such as kinetic energy, light, sound or electricity. The coordinated muscle actions involved in sound production result in pressure changes only, which can be captured by microphones.

3.2.1 Muscle actions

Isometric actions

For muscle actions inducing variable pressure with no large-scale movement, two types of sensors are used :

- *Push-buttons.* Two triggers have been placed under both thumb-rest pads of the saxophone. These allow some event-triggering without movement and constitute a good alternative to foot pedals which are not always easy to use and might disrupt the flow of the performance. Any kind of triggering action can be made with these buttons: launching pre-recorded sound samples, enabling an effect, switching on or off some light or video projection.

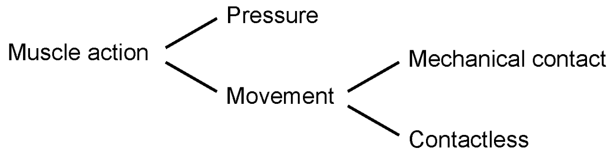


Figure 2: Sub-categorization of muscle action output modalities (after [3]).

- *Force-sensing resistors.* Next to the thumb triggers, two force-sensing resistors (FSRs) can be used to control continuous values such as the amount of harmonic distortion in the signal. A third FSR is mounted under the octave-key, where the cork comes in contact with the body of the saxophone. Since the octave-key is used for a wide pitch range on the saxophone and is not sensitive to small opening changes, continuous values can also be controlled while simultaneously playing and slightly changing the thumb pressure on the octave-key.

  It should be noted that pressing with a finger on an FSR does not generate a very accurate control signal. This should be taken into consideration when determining the mapping.

Movements with mechanical contact

- *Slide potentiometers.* This type of sensor can be placed on the left side of the saxophone bell to control continuous values such as the cut-off frequency of a filter. New instrumental gestures such as "stroking the bell" can be exploited to control effects.
- *Foot pedals.* If needed by the composition or to spare other sensor channels, MIDI foot pedals are added to the set-up (here we use a Behringer FCB1010 with ten triggers and two expression pedals). They can serve to trigger events or to control the main volume of the P.A. system of a concert room.

Contactless movements

Muscle actions inducing contactless movements are captured using three types of sensors which have complementary functions:

- *Inclinometer.* Mounted on the saxophone bell, a one-dimensional accelerometer is used as an inclinometer that generates a continuous value analog to the angle between the instrument axis and the floor. The accompanist gesture associated with the inclinometer output could be mapped to several processing control parameters, such as the amount of pitch-shifting and the reverberation level.
- *Ultrasonic distance sensor.* The distance to a reflecting surface (*e.g.* a metallic instrument) can be derived from the delay between the time a short ultrasonic pulse train is emitted and the time its reflection on the surface is received. From the distance values generated by three synchronized ultrasonic emitter/receiver devices placed at different points on the

Module name:	**VIDEO-TRACKING**
Sensor:	Webcam
Patcher:	jit.videotrack
Communication:	USB
# parameters:	2
Kind of values:	continuous
Resolution:	320 x 240
Accuracy:	medium*
Range:	0.1 – 10 m^2
INs:	- Video ON/OFF (toggle)
	- Open / close video stream (toggle)
	- Get video devices list (bang)
	- Selected device (string)
	- Get input mode list (bang)
	- Selected input (text)
	- Min. RGB value to track (3 x integer)
	- Max RGB value to track (3 x integer)
OUTs:	- Monitor window (jitter matrix)
	- Devices list (string)
	- Input mode list (string)
	- Tracking window (jitter matrix)
	- X coordinate (0 – 1 float)
	- Y coordinate (0 – 1 float)

* Accuracy changes depending on the light conditions and color contrast of the background. Obscurity: very good; daylight: poor

Figure 3: Specification chart for the webcam video-tracking module.

stage, the absolute position of the reflecting surface is calculated by triangulation in a Max/MSP patch. The module would typically deliver the X-Y coordinates of the performer on stage. The most intuitive mapping for this sensor is probably to link the balance between the stereo speakers left and right channels to the X (horizontal left-right) axis and the volume control to the Y (horizontal front-back) axis.

- *Video-tracking.* The color-tracking module is composed of a commercial webcam and a Max-jitter patch as shown on Figure 4. It can follow a LED placed on the performer or a spot light projected on a vertical surface in the X-Z plane (width and height of the stage). We use a one-by-one square meter plate as a projection surface for a laser pointer. The surface is divided in several invisible zones, which can trigger some pre-recorded sound events for example.

3.2.2 *Sound production*

Four acoustic parameters can be extracted from the sounds produced on the saxophone and used as distinct control values.

- *Intensity level.* An envelope follower generates a continuous control signal that can be used to modulate an effect or automatically adapt the volume of any pre-recorded sound event to the instrument's dynamic level. Coupled with some adjustable threshold detection, it also allows conditional triggering of an effect, switching on and off according to the specified thresholds.

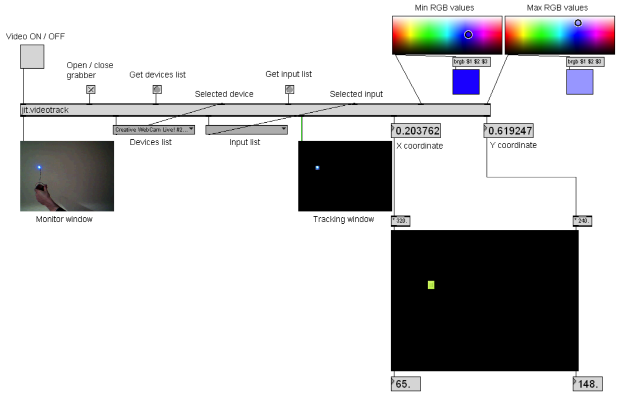


Figure 4: *jit.videotrack* patch with a webcam as input.

- *Attack detection.* Attacks are detected by monitoring sudden level increase in a given time period. The level difference and the time period – which also defines the shortest time between two attacks – are both adjustable parameters. Such a control signal allows quick repetitions and can be used to trigger randomized and short events (*e.g.* grains of a granular synthesis for example or impulses of a strobe light[6]).

- *Pitch estimation.* The pitch is extracted using the *fiddle~* object in Max/MSP. This signal can be coupled with the attack detection module for more reliability in the triggering signal.

- *Zero-crossing.* The number of times a waveform crosses the zero line provides a measure of the sound noisiness (since a noisy waveform crosses the zero line much more frequently compared to a periodic waveform). An adaptive processing of the saxophone tone could be based on this control signal.

4. ON PLAYING AN AUGMENTED SAXOPHONE

4.1 Functional reconfiguration of gestures

The addition of several buttons or movement sensors on an instrument indubitably affects the way the performer interacts with it. The functional levels of gesture (instrumental, accompanist and figurative) can be greatly affected by the electronic extension, by either adding new types of gestures of reconfiguring the functional levels themselves.

At the instrumental level, on an electronically-augmented instrument, excitation gestures not only include blowing into the mouthpiece but also triggering actions, such as pushing a button or moving in front of a video-tracking system. Modification gestures can be pressing on a FSR or moving a slider. The produced sound itself can be used as a control signal. For example, the trigger signal from an attack detection can enable some sound effect.

[6]The input sound is converted into a trigger signal (bang), which is sent through a Virtual COM Port [14] to a USB-to-DMX signal converter [13], then to the strobe as a standard DMX command.

The electronic extension of an acoustic instrument usually causes an important reconfiguration of the functional levels. For example, with an acoustic saxophone, inclination of the instrument as well as circular or eight-shape movements of the bell are accompanist gestures which do not need to be tightly controlled since they have a very subtle influence on the sound [15]. But if the instrument is equipped with a motion tracking device, these gestures become instrumental when they are mapped to directly produce sound events or modify them. As a result, the performer needs to consciously and precisely control this type of gesture, which was mostly uncontrolled when playing the non-extended acoustic instrument. Integrating this functional mutation (from accompanist to instrumental level) is not obvious to the performer and will requires many hours of practice. The performer also has to think of all uncontrolled movements that could interfere with some event-producing gestures. For example, elbow movements could lead to some pressure changes on the thumb, thus interfere with an FSR control.

4.2 Playing one or two instruments

The mapping strategies have an important impact on how the instrument will be played and on how the audience will perceive the performance.

Some carefully designed mappings will allow a good integration of both acoustic and electronic components of the performance, resulting in one single instrument: an electronically-augmented acoustic instrument. In this case, the electronics is used to extend the timbral palette of the acoustic instrument by transcending its physical limitations. The electric guitar is a good illustration of an electric extension of an acoustic instrument. It is in fact a hard task to identify the conditions under which an acoustic instrument can be electronically augmented without loosing coherence with its original acoustic characteristics.

In some other cases, the performer of an electronically-augmented instrument may seem to play two instruments, one acoustic and one electronic [6]. From the performer's standpoint, the instrument is split in two parts being played simultaneously: a "standard" acoustic instrument and a controller driving other events (sound effects, electronic accompaniment of the solo acoustic instrument, video images, ...). It is as if the performer had to play alone in a duet formation, controlling two instruments at the same time. This situation generally leads to an increased cognitive load and requires more practice to achieve accurate control.

4.3 Choosing sensors and mappings

In section 3, we have suggested some mapping possibilities for the various sensors.

The advantage of a fixed mapping is that the performer does not need to relearn each time how to play his instrument. The wah-wah effect on an electric guitar sound is a good example of fixed mapping. It is always controlled by an expression pedal. In the case of this project, the system has been conceived to be flexible and adaptable to the pieces to be performed. Although the system is flexible, it is important to determine mappings which are intuitive to the performer and that take into account electronic, acoustic, ergonomic and cognitive limitations (accuracy, resolution and response time of sensors, added cognitive load corresponding to the type of sensors, player's technical ability, etc.).

In order to decide on a particular setup, many questions need to be answered:

- How many parameters is the performer able to simultaneously control?
- What are the constraints on movements induced by the added sensors?
- How to evaluate the cognitive load of the electronic extension?
- Are they strategies to reduce this cognitive load?
- Which sensors and mappings are more intuitive to the performer?
- How long does the performer need to practice to become comfortable with a particular set-up?

4.4 New ways to practice

Performers should probably invent new ways to practice, learn how to use their feet, body movements or fingers independently. Solutions to this problem could be borrowed from dance or theater domains. Some basic exercises using the various sensors of a set-up – similar to the common scale drills on acoustic instruments – could be developed to get more comfortable with the electronically-augmented instrument. Appreciating this reality and playing with its limits will help to improve the instrument's capabilities, as the extended techniques[7] did some decades ago.

In the design of new instruments, we could question the tendency to reduce learning time and performer's needed skills [1][11]. The instruments that have the highest musical potential are not always the ones which are the easiest to use or to learn. In fact, the investment required in learning to play an instrument and the intimacy derived from that process are worthwhile and essential in the evolution of an instrument. Citing Berio, technical and intellectual virtuosity "may also count as the celebration of a particular understanding between composer and performer, and bear witness to a human situation" [2].

4.5 Interacting with the composer

With a flexible and configurable augmented saxophone, the performer can either play an existing piece for saxophone and electronics (which generally needs some porting and rewriting of code), compose and improvise on the instrument or collaborate with a composer to create new works. In this last situation, the compositional process can adopt a *bottom-up* or a *top-down* approach.

A bottom-up approach refers to the development of compositional ideas according to the possibilities and limitations of a given instrument. The performer explores a particular set-up and feeds the composer with informations on the various capabilities of the instrument and its controls. The composer can also select a set of modules and their gestures and use it as a canvas to work from. With a bottom-up approach, we run a higher risk that technology becomes the main justification of the whole composition.

A top-down approach rather refers to the development of compositional ideas without any regard for instrumentation[8]. The composer first writes the piece without con-

[7]The extended techniques allow to produce unconventional sounds, like flutter tongue, slap tongue or multiphonics. A quite complete list of these techniques for the saxophone can be found at `http://www.jayeaston.com/Composers/sax_techniques.html`.

[8]From Bach's *Kunst der Fuge* to Stockhausen's *Solo*, there are many works written without specific instrument designation.

straints induced by the instrument. Then the best sensors and mappings are chosen to fit the needs of the composition. Ideally, the flow of information between the performer and the composer should run in both directions. This can lead to interesting and new types of work dynamics between composers, performers and engineers.

5. CONCLUSION

In this paper we presented a modular low-tech electronically-augmented saxophone. Modules from the toolbox we developed are selected and added to the saxophone to best fit the needs of a composition. The selection of modules in a particular set-up can also serve as a compositional canvas.

Several on-going musical projects with contemporary composers currently explore the capabilities of this electronically-augmented saxophone. The modular toolbox also makes possible the recreation of older works for saxophone and electronics (for which the technologies does not exist anymore) by choosing the appropriate sensors, mappings and signal processing algorithms.

As the electronic augmentation of an acoustic instrument reorganises the functional level of gestures applied on the instrument, performers have to become more conscious of their movements and need to find new practice methods to become comfortable with the transformed instrument.

6. ACKNOWLEDGMENTS

We are grateful to Prof. Marcelo Wanderley from the Input Devices & Music Interaction Lab (IDMIL – McGill University) for the access to his laboratory, as well as the whole IDMIL crew for sharing their knowledge and infrastructure.

Thanks to Mark Marshall who designed the AVR-HID microcontroller interface and spent a lot of time helping and debugging, as well as to Kris Covlin for corrections, suggestions and exciting exchanges about the saxophone.

Very special thanks to Sonia Paço-Rocchia for fruitful conversations, tips in programming and all the energy you gave during this time.

7. REFERENCES

[1] A. Baumann and R. Sanchez. Interdisciplinary applications of new instruments. In *NIME-02 Proceedings*, 2002.

[2] L. Berio. *Sequenzas*. Deutsche Grammophon, 1998. Booklet of the CD *Sequenzas* by the Ensemble Intercontemporain.

[3] B. Bongers. Physical interfaces in the electronic arts. In *Trends in Gestural Control of Music*. M. M. Wanderley and M. Battiez, 2000.

[4] M. Burtner. The metasaxphone: Concept, implementation, and mapping strategies for a new computer music instrument. *Organized Sound*, 7(2), 2002.

[5] M. Cadoz and M. M. Wanderley. Gesture - music. In *Trends in Gestural Control of Music*. M. M. Wanderley and M. Battiez, 2000.

[6] C. Cox. Interactive technologies in music composition: Towards a theory of interactivity. In R. H. M. Ewans and J. A. Phillips, editors, *Music Research: New Directions for a New Century*. Cambridge Scholars Press Titles, 2004.

[7] F. Delalande. La gestique de gould: éléments pour une sémiologie du geste musical. In G. Guertin, editor, *Glenn Gould, Pluriel*, pages 83 – 111. Louise Courteau Editrice Inc., 1988.

[8] K. Demers. Acoustique de la clarinette et exploration de nouvelles possibilités timbrales au moyen de l' électroacoustique. Rapport. Laboratoire d'informatique, acoustique et musique, Université de Montréal, 2005.

[9] J. Impett. A meta-trumpet(er). In *ICMC Proceedings*, 1994.

[10] T. Liley. Invention and development. In R. Ingham, editor, *The Cambridge Companion to the Saxophone*. Cambridge University Press, 1998.

[11] K. Ng. Interactive gesture music performance interface. In *NIME-02 Proceedings*, 2002.

[12] C. Palacio-Quintin. The hyper-flute. In *NIME-03 Proceedings*, 2003.

[13] ENTTEC. *Open DMX USB*. `http://www.enttec.com/dmxusb.php`.

[14] FDTI Chip. *Virtual COM Port*. `http://www.ftdichip.com/Drivers/VCP.htm`.

[15] M. M. Wanderley. Non-obvious performer gestures in instrumental music. In *Gesture-Based Communication in Human-Computer Interaction: International Gesture Workshop*. Springer Berlin, 1999.

[16] M. M. Wanderley. Gestural control of music. In *Proceedings of the International Workshop on Human Supervision and Control in Engineering and Music*, pages 101–130, Kassel, Germany, 2002.

Handheld Acoustic Filter Bank for Musical Control

Tamara Smyth
School of Computing Science
Simon Fraser University
British Columbia, Canada
tamaras@cs.sfu.ca

ABSTRACT

This paper introduces the design of a handheld musical input device that produces control data by measuring, and analyzing, the resonances of carefully tuned pipes. The device provides input control information to several virtual reed instruments running in parallel, responsible for producing the sound. Inspired by the khaen, a musical instrument from Northeast Thailand and Laos, the controller consists of a row of acoustic tubes, with finger holes that change the tube's resonance when covered. Each tube is equipped with both a microphone recording the change in pressure variations at a set location along the tube. The mic outputs are mixed, and input to the computer via the mic level audio ports, allowing the controller to interface very reliably (and conveniently) to most laptop computers.

Keywords

khaen, sound synthesis control, mapping, musical acoustics

1. INTRODUCTION

Western "breath-driven" musical instruments [1] typically make use of a mouthpiece and a single bore that can take on a variety of shapes from cylindrical tubes to conical or flared horns, or some piecewise combination thereof. If a player is to change the pitch on such an instrument, some mechanism, such as toneholes or nested sliding tubes, must be in place for changing the bore's effective acoustic length[2]. These mechanisms have served musicians well over the course of western music history, so well in fact, that their ubiquitousness continues to pervade our paradigms for new musical input devices. Though these devices have offered substantial possibilities for controlling tone [4, 10, 11, 3, 6], because they are monophonic they are far less suitable as controllers for more general musical needs.

Following research on the generalized virtual reed model [12], there existed a need for an input device that would allow for the control of an arrangement, in series or parallel, of multiple pressure-controlled valves. As is the tradition in the development of new music controllers, the author began by looking at existing tried and true control mechanisms that would meet the requirements. The *khaen*, a breath-driven instrument from Northeast Thailand and Laos, consisting of two rows tuned acoustic pipes which may be played independently or simultaneously, served as the model.

Many new musical controllers are developed independently from the computer sound synthesis algorithms they control. This disconnect has made it increasingly difficult to develop mapping strategies for parameter laden physical modelling synthesis algorithms, for which the input control parameters and the synthesis parameters are very strongly linked. This research therefore also attempts to address the notion of *re-linking* acoustics to controller by measuring and analyzing the resonances of carefully tuned pipes. That is, rather than using sensors to measure the user's input directly, a microphone measures the state of the pipe as it is altered by the user.

The benefits of this approach are twofold: 1) the user isn't forced to interact directly with bulky, often overly wired and flaky electronics that tend to break, and frequently need replacing from over-use and 2) the controller can directly interface with the audio ports on a laptop computer, without requiring a micro-controller and a lot of extra circuitry for data acquisition—returning the concept of portability to musicians who use laptop computers during performance.

The idea for this controller was inspired both by ongoing research on the Khaen, as well as Bernard Chouet's approach of detecting "long period events" for predicting imminent volcano eruptions [8]. Both make use of resonance phenomena.

2. KHAEN ACOUSTIC DESIGN

The *khaen* is a free reed, mouth-organ style instrument found in the Northeast region of Thailand (Issan) and Laos. Very similar to the Japanese *sho*, or the Chinese *sheng*, it consists of two rows of bamboo tubes decreasing in length, each tube having its own reed, and being responsible for a single pitch lying within a diatonic scale [7]. The player supports the instrument upright in front of the mouth, with the

[1]The term "breath-driven" is used to distinguish from *wind* or *wind-driven* instruments, which would include organs and accordions, among others.
[2]Some reed driven instruments, such as the digiridoo, rely exclusively on the mechanical resonance of the reed to alter the sounding pitch, though their pitch range is rather limited as a result.

Figure 1: Mr. Khene, a khaen maker from Roi Et, Northeast Thailand, displays the tuning vents in the pipes.

windchest resting comfortably between the palms of prayer positioned hands, with holes within the fingers reach. No sound is produced from a particular pipe unless its finger hole is covered. Blowing, or inhaling, through a mouth hole provides a pressure or vacuum inside the air chamber surrounding the reeds [2], creating a differential pressure across the reed that forces it into oscillation.

Pipes are inserted into a windchest, called the *tao*, so they form two rows of anywhere from 6-10—but usually 8—pipes (see Figure 1). The perfect skyline arrangement of the pipes is purely aesthetic as the actual pipe lengths are determined using vent holes placed a distance L apart, approximately 3/4 L above and approximately 1/4 L below the position of the reed (see Figure 1). The distance L is therefore, the effective acoustic length of the pipe when the finger hole is covered. An open finger hole, melted into the bamboo using a heated iron awl, drastically reduces the acoustic length of the pipe, altering its resonance so that it no longer couples strongly with the reed.

Finger holes in the khaen do not function as toneholes in single bore instruments. Though the small hole alters the effective acoustic length of the pipe if left uncovered, it also destroys the pipe resonance and prevents the reed from vibrating. Therefore it is possible to blow into the instrument and not produce any sound. The player covers only the holes for the pipe(s) that should sound, and may play several notes simultaneously. The rubbery material, called *kisoot*, that seals the pipes and the *tao*, may also be used to close finger holes, creating a fuller sound with drones or pedal notes.

The arrangement of the pipes and the proximity of their finger holes make it relatively easy to play notes in rapid succession (with a little practice). Since very little finger pressure is needed to play a pipe, it is fairly easy to flutter the fingers over the holes and, again with practice, to play with speed and virtuosity. This is an aspect of the khaen that would be very difficult to capture using force sensing resistors but that has been retained in this design. Khaen players will also say that the texture of the hole outline beneath the finger provides an important cue for locating its position (which, is important given the holes are not visible during performance) [1]. This feature is also retained in this design. An extension of the Khaen within the capability of the controller, is the implementation of partial hole coverage, rather than the binary choice of open or closed. This may be less helpful to traditional players, but certainly offers more flexibility to those who wish to extend the instrument [9].

3. ACOUSTIC FILTER BANK FOR INPUT PARAMETERS

The controller is designed to serve as an input device for playing a parallel arrangement of generalized reed synthesis models [12]. Clearly the arrangement of pipes, one for each sounding pitch, is a particularly useful paradigm for controlling several virtual reeds. The controller however, also incorporates the Khaen's use of resonance as a mechanism for control.

Like the khaen, the controller consists of a bank of carefully tuned pipes with finger holes placed within the reach of the player's hands, and a mouthpiece supplying pressure variations—though this research focuses on capturing the fingering of the player rather than his/her blowing technique (which in reality has very little variation since the player must blow rather hard to produce sound).

Each tube n is cut to a length L_n, and closed at either end to ensure a "perfect" reflection—a speaker seals one end while a piece of acrylic seals the other. A small finger hole is placed in the wall of the tube at the end opposite the speaker, effectively opening that end (see Figure 2). Closing the finger hole changes the configuration of the pipe from one that is closed-open to one that is closed-closed.

Notice from Figure 2 that though displacement of air is greatest at an *open end*, the pressure variation is maximum at a *closed end*. For the first harmonic of a closed-closed tube, there is maximum pressure variation at both ends, with a pressure null at the center. For the second harmonic of the same configuration, adding a node and antinode produces a standing wave pattern with a pressure maximum in the center of the tube. Driving the tube at a frequency equal to this second harmonic, and placing a microphone at the center of the tube, will measure maximum pressure variation when the hole is covered. Uncovering the hole reverts the configuration back to a closed-open tube with an altogether different (lower) resonance, with the second harmonic (and every even harmonic) missing (see Table 1, [5] or any acoustics text for calculating tube resonant frequencies). As a result, there is a considerable increase in the sound pressure level recorded by the microphone if the hole is covered.

A signal is generated from the computer that combines the appropriate driving frequencies (see frequencies from Table 2). This signal is output via the speakers at the end of each tube, and is effectively filtered by a bank of acoustic filters, that is, each pipe acts as a filter, boosting or attenuating only those frequencies corresponding to one of its resonant modes. The output from each of the microphones is added together using a very simple mixing circuit (see Appendix), and the sum is input back into the computer's mic level audio inputs, scanned and analyzed to determine which pipes have been selected for sound production.

The sound produced by the tubes must be within the audio range (not exceeding the Nyquist limit) so that the input and output can be handled using the computer's audio

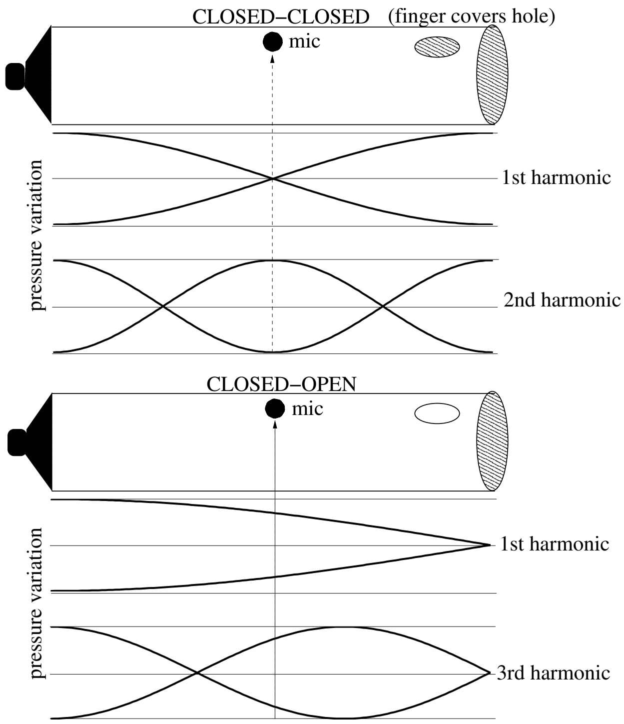


Figure 2: The first two modes of vibration, showing pressure variation maxima and minima, and standing wave patterns, for closed-closed acoustic tubes (resulting from the the finger covering the hole), and closed-open tubes (where the finger hole is open).

Boundary	Harmonic k	Wavelength λ	Frequency $f_k = c/\lambda$
Closed-Closed	1	$2L$	$f_1 = 425$ Hz
	2	L	$f_2 = 850$ Hz $= 2f_1$
Closed-Open	1	$4(L + 0.61a)$	$f_1 \approx 206$ Hz
	3	$\frac{4}{3}(L + 0.61a)$	$f_3 \approx 618$ Hz $= 3f_1$

Table 1: Modes of vibration for a closed-closed and closed-open pipe of length $L = 40$ cm and radius $a = 2$ cm, computed using speed of sound, $c = 340$. In an actual pipe, pressure variations drop to zero slightly beyond an open end, effectively increasing the acoustic length for this condition by approximately $0.61a$, where a is the radius of the pipe.

		Closed-Closed		Closed-Open	
pipe	L (cm)	f_1 (Hz)	f_2 (Hz)	f_1 (Hz)	f_3 (Hz)
1	20	1700.0	**3400.0**	0757.6	2272.7
2	12	1416.7	**2833.3**	0643.0	1928.9
3	14	1214.3	**2428.6**	0558.5	1675.4
4	18	0944.4	**1888.9**	0442.2	1326.7
5	20	0850.0	**1700.0**	0400.6	1201.7
6	22	0772.7	**1545.5**	0366.1	1098.2
7	30	0566.7	**1133.3**	0272.3	0816.8
8	35	0485.7	**0971.4**	0234.7	0704.0

Table 2: Tube lengths L (in centimeters), closed-closed (Cl) or closed-open (Op) and frequencies f_0 and f_1 (in Hz) for 8 pipes.

ports. The closed-closed tube configuration was chosen to limit sound radiation to the player's ear when a tube resonates, as the user holds the instrument (at the mouth) in fairly close proximity to the ears. Since all the tubes receiving the same excitation signal, the tubes had to be given a variety of lengths, shown in Table 2, to ensure that the 2nd harmonic of one closed-closed tube doesn't excite a harmonic of another closed-open tube.

Selecting a tube on the controller activates a synthesis model of a generalized reed, fully detailed in [12], producing the instrument sound. Though the controller currently doesn't allow the user to modify many of the model's continuously variable input parameters, it solves the problem of playing several reeds simultaneously. For some reed types, particularly free reeds, where only a limited number of combinations of input parameters produced satisfying sound quality, the ability to control several models was all that was required.

4. CONCLUSIONS

With the surfacing of new controllers and computer sound synthesis algorithms, there has been a clear disconnect between the device that produces the sound and the device with which the player/musician interacts. Various sensors may be used to capture human input that has very little, if anything, to do with the mechanism or acoustics of the produced sound. Though this would seemingly increase the possibilities for the design and development of new controllers unfettered by the demands of the acoustic systems they control, in actual practice it seems to have been more of a hindrance in the development of quality musical instruments with intuitive mapping.

This work presents the design of an instrument that uses the acoustic information of the controller, and remaps it for parameter control of a computer synthesis model—effectively preserving, but without being limited or constrained by, the acoustics on which it is based.

5. ACKNOWLEDGMENTS

The author is indebted to Dr. Chris Adler from San Diego University for his ongoing help on khaen instrument research. Many sincere thanks are also extended to Agnieszka Roginska, who spent three weeks with the author on a field recording trip in Thailand, to AuSIM Inc., and in particular William L. Chapin, for donating employee time, and to Kritsana Naklang, Jarernchai Chonpairot, and the late

Figure 3: Mr. Samong plays a Khaen using "extended" techniques.

Mr. Utit, who were all indispensable in guiding us through villages in Northern Thailand, and serving as translators.

6. REFERENCES

[1] C. Adler. Private correspondence.

[2] J. P. Cottingham and C. A. Fetzer. The acoustics of the khaen. In *Proceedings of ISMA*, pages 261–266, Leavenworth, Washington USA, 1998.

[3] A. da Silva, M. M. Wanderley, and G. Scavone. On the use of flute air jet as a musical control variable. In *Proceedings of NIME*, pages 105–108, Vancouver, Canada, May 2005.

[4] S. Fels and F. Vogt. Tooka: Explorations of two person instruments. In *Proceedings of NIME 2002*, Dublin, Ireland, May 2002. Conference on New Instruments for Musical Expression.

[5] N. H. Fletcher and T. D. Rossing. *The Physics of Musical Instruments*. Springer-Verlag, 1995.

[6] S. Hughes, C. Cannon, and S. O'Modhrain. Epipe: A novel electronic wind controller. In *Proceedings of NIME 2004*, Hamamatsu, Japan, June 2004. Conference on New Instruments for Musical Expression.

[7] T. E. Miller. *Traditional Music of the Lao: Khaen Playing and Mawlum Singing in Northeast Thailand*. Greenwood Press, Westport, Connecticut USA, 1985.

[8] NOVA and B. Chouet. Volcanoe's deadly warning. http://www.pbs.org/wgbh/nova/volcano/chouet.html.

[9] R. Raine-Reusch. http://www.asza.com/r3hm.shtml.

[10] G. Scavone. The pipe: Explorations with breath control. In *Proceedings of NIME 2003*, Montreal, Canada, May 2003. Conference on New Instruments for Musical Expression.

[11] G. Scavone and A. da Silva. Frequency content of breath pressure and implications for use in control. In *Proceedings of NIME*, pages 93–96, Vanccouver, Canada, May 2005. International Conference on New Interfaces for Musical Expression.

[12] T. Smyth, J. Abel, and J. O. Smith. A generalized parametric reed model for virtual musical instruments. In *Proceedings of ICMC 2005*, Barcelona, Spain, September 2005.

APPENDIX

A. MIXER DESIGN

The microphone signals must be added together before connecting them to the mic level input of the laptop. Simpling placing the speaker outputs in parallel will overload the outputs and likely increase the distortion in the sound. A better approach is to create a very simple mixer circuit, which simply involves inserting the appropriate resistor (see Figure 4) at the output before summing them together.

Since the computer's mic level audio input is stereo, two mixers can be used to sum the microphone signals from each row, and then input into the computer via the left and right channel.

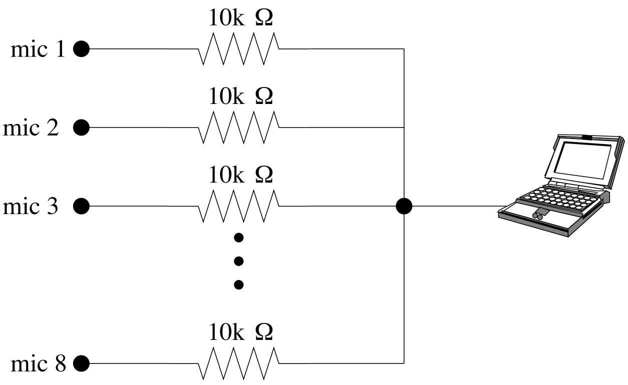


Figure 4: A simple mixing circuit for up to 8 input channels.

Real-Time Sound Source Spatialization as used in *Challenging Bodies*: Implementation and Performance

JJ Nixdorf
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada
nixdjosh@cs.uregina.ca

David Gerhard
Departments of Computer Science and Music
University of Regina
Regina, Saskatchewan, Canada
gerhard@cs.uregina.ca

ABSTRACT

In this paper we will report on the use of real-time sound spatialization in *Challenging Bodies*, a trans-disciplinary performance project at the University of Regina. Using well-understood spatialization techniques mapped to a custom interface, a computer system was built that allowed live spatial control of ten sound signals from on-stage performers. This spatial control added a unique dynamic element to an already ultramodern performance. The system is described in detail, including the main advantages over existing spatialization systems: simplicity, usability, customization and scalability

Keywords

surround sound, sound spatialization, sound localization, sound architecture, real-time systems, performance systems, live systems, pd, GEM

1. INTRODUCTION

In June 2005, researchers from various faculties of the University of Regina, including Music, Theatre, Dance, Kinesiology and Computer Science, came together to collaborate on *Challenging Bodies*[1]. Billed as "A Multi- disciplinary Performance for Variously-Abled Artists", the purpose of the project was to challenge the notion of who is abled and who is disabled. The main performer of this project, Craig Fisher, has cerebral palsy and is confined to a wheel chair. A custom computer system called Cool Moves was developed locally by Music Therapist Doug Ramsay based on David Rokebys Very Nervous System (VNS)[2]. Using this, Craigs limited range of controllable movement was mapped to a virtual instrument. Other facets of the performance included an interpretive dance group featuring a performer confined to a wheel chair (a "sit-down dancer"); a visualization system that projected graphics based on Craigs musical output; A video-mixing artist using previously recorded motion-capture footage of the dancers, and a full-theatre sound spatialization system.

The spatialization system, built in Pure Data (pd) [4], took all of the sound input from each of the ten instruments (played by six performers) and mapped those input channels to a position in a virtual representation of the performance theatre. Using fundamental sound localization techniques, these virtual locations were spatialized to a corresponding position in the actual theatre. Most importantly, this interface operated in real-time, allowing the user to animate a sound throughout the theatre, thus giving the impression of a moving musician who could also respond to the motion of the virtual source through which he was playing. The spatialization system user and the instrumentalists therefore needed to be very attentive to each other, and the spatialization system user became like a member of the ensemble. The interaction brought to the fore some very interesting reactions to live spatialization in performance, which will be described in Section 4 after the system itself is described in the following sections.

[1] http://uregina.ca/~challbod/
[2] http://www.davidrokeby.com/

2. BACKGROUND

Sound spatialization (localization, sound architecture) focuses on the position and motion of sounds through an actual or virtual performance space. Sound localization for musical expression is a familiar concept in current new interface studies. Most presentations of new music and interface technology use some form of spatialization to enhance the performance. Often, this spatialization is recorded beforehand and the movements of the individual sound sources are fixed during performance. *Diffusion* refers to (among other things) the live spatialization of mono or stereo sources by fading or panning them between speakers in a multi-channel environment, causing the sounds to appear to be coming from different angles. The interfaces to diffusion systems are often based on the physical mixers and connections to the speakers themselves, and diffusing more than one source at a time is a difficult task, let alone diffusing ten or more sources independently [7, 3].

The use of spatial music in performance is not new. Classical music examples include spatial antiphony in choir performance (Willaert and Gabrieli); Mozart's pieces for multiple orchestras (K. 239 and K. 286); and Verdi's Requiem (Tuba Mirum), wherein trumpets are positioned offstage. Notwithstanding these, localization is a relatively uncommon technique in popular classical music, and is often considered a novelty rather than an expressive technique. This is not surprising given the cost and effort required to produce any complementary result. Historically, implementing localization on any grand scale would have been a cumbersome endeavor.

As early as 1956 [5], sound installations capable of spatializing sound through a loud speaker setup begin to appear. These installations vary wildly in form and function. Due to the hardware involved, the compositions for these arrangements are usually specific to the installation they were composed for, thus the installation is little more than a glorified instrument. In 1990, a Stanford Researcher, Marina Bosi, produced an entirely digital spatialization system [1], which was an early instance of using multiple speakers in a connected way, as opposed to a forest of independent sound sources. This system accepted MIDI signals as input and spatialized these signals into an adjustable quadraphonic setup. This was a step towards allowing a composer to use spatialization in any performance, however MIDI was still a limiting factor on the performance characteristics of the system. The more recent *GMEM* spatialization system, *Holo-Spat*[3], has improved functionality but will spatialize any sound file, not just MIDI. *Holo-Spat* has an extremely limited interface for spatialization control and requires it's sister program, *Holo-Edit*, in order to perform any complex spatialization. *Holo-Edit* records the desired spatialization to a MIDI control file, which *Holo-Spat* then uses to localize the sound sources. The only downside to this arrangement is that since *Holo-Spat* and *Holo-Edit* are uncoupled, it is impossible for any complex spatialization to be played as it is composed, or to be performed in real-time. Although beneficial for composing and recording purposes, this is certainly not advantageous for live performance. Another current system is the IRCAM[4] *SPAT* , which has many of the same features as *Holo-Spat*. *Holo-Spat* and *SPAT* both require the purchase of Max/MSP[5] in order to customize the system, and Max/MPS licenses are expensive for individual users.

The system discussed in this paper has a unique interface which allows for the real-time manipulation of sound sources in the virtual environment. The spatialization features implemented are based on the requirements for the *Challenging Bodies* application, and do not make use of such straightforward spatialization techniques as distance cues or doppler shift. Earlier versions of the system did have all of the standard spatialization cues, and they are still present within the code of the system.

3. IMPLEMENTATION

The spatialization system was implemented from the beginning based on three primary concepts: scalability, usability, and non-proprietarity (no monetary cost). Each of these concepts represents an advantage over existing systems. When we began this work we could not find a complete, usable, open-source spatialization interface for `pd`. All of the systems described earlier have an associated cost (our system is freely available[6]) and are not as usable as they could be. For instance, the *Holo-Spat* + *Holo-Edit* pair relies heavily on modes, considered to be a disadvantage for usability [6]. Modern usability theory has yet to be applied to the majority of computer-music interface applications.

The scalability of the system can be seen in Figure 1, where each sound source signal is applied to a router, which compares the virtual location of that source with the virtual location of all speakers, and routes the source to each speaker, with the appropriate attenuation. The addition of a new source or a new speaker is theoretically trivial, corresponding of the addition of a new router or mixer respectively. As indicated by the dashed line in the diagram, the GUI currently does not alter the position of the speaker, but there is no theoretical reason why the speakers cannot be manipulated in the GUI.

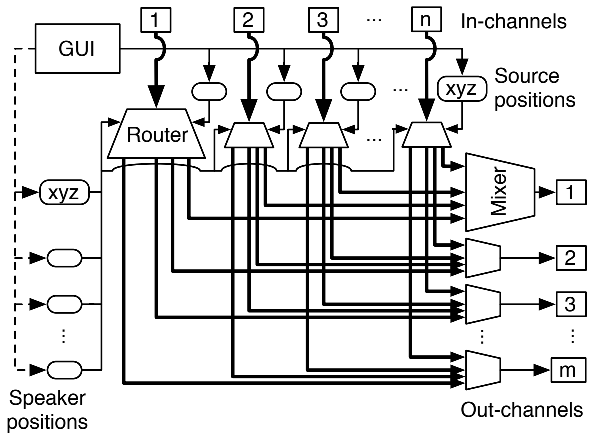


Figure 1: The flow layout of the system. Each router takes the relative position of its source and all speakers and calculates the relative amplitude of that source for each speaker.

The main interface of the spatializer consists of two primary windows, both of which exist on screen simultaneously. The first is the primary graphical interface of the spatializer, the GEM[7] [2] window. It paints a simulated speaker environment to the screen and allows for mouse manipulation of the spatial location of the sound sources. The GEM window, shown in Figure 2, is intended to be a representation of the physical speaker arrangement of the performance hall as illustrated in Figure 3. The speakers are arranged in an array behind the performers, with five speakers at audience level from stage right to stage left, and two speakers in the catwalk for elevation simulation. The decision was made not to have speakers behind the audience in an effort to increase the size of the "sweet spot," the audience location with the ideal listening conditions. The speaker layout in the GEM window is therefore different from the standard speaker-ring configurations of current systems. Instead, the system is presented as if from the audience's point of view. This shows that spatial interfaces can be designed from a multitude of perspectives, and that this system is capable of localizing sound sources in any general speaker arrangement. Indeed, whereas the presented layout is limited to 2-D by the single orthogonal view on the screen, multiple views or a single oblique view would provide opportunities for manipulating sources in 3-d in real-time. The current implementation encodes the location of the speakers and the sources in 3-d, but because a single layer of speakers is used, it can be projected onto a 2-d interface.

The second window is a `pd` patch, `[rtsss.pd]`, which has the master output volume control as well as individual sliders controlling the input level of each sound source. It should be noted here that while the speaker arrangement

[3]`http://www.gmem.org/`
[4]`http://www.ircam.fr/`
[5]`http://http://www.cycling74.com/`
[6]`http://armadilo.cs.uregina.ca/rtsss`
[7]`http://gem.iem.at`

was fixed for this performance, in general the speakers can be moved and positioned anywhere in 3-space. This is a significant advantage of this system over current implementations which normally assume a ring of 5 or 8 speakers in a standard configuration. Standard configurations are ideal for recorded spatialization playback since composers can rely on a known arrangement of speakers, however, this requirement is less important than the actual speaker arrangement for live performance of spatial source positioning, and variable speaker placement allows the accurate representation of the playback space, since speaker arrangements in real performance venues are rarely perfectly circular.

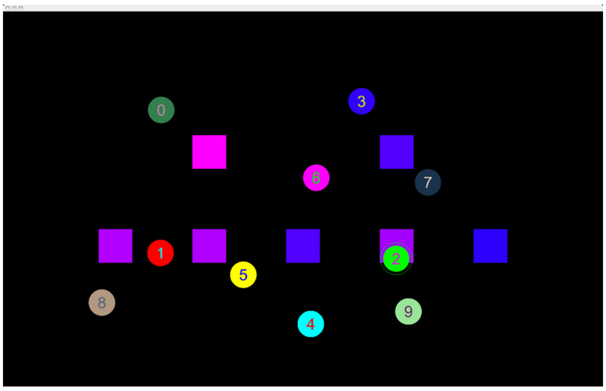


Figure 2: The GEM window. Squares represent speakers. Circles represent sound input signals.

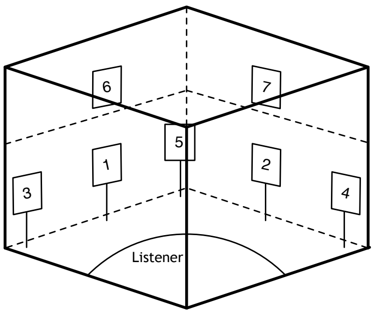


Figure 3: The physical speaker arrangement in the concert hall.

4. PERFORMANCE TESTING

Although the spatializer was completed only a short time before the performance date, it had undergone extensive testing and two complete versions during development. This ensured that the more fundamental issues had been discovered and corrected, and that the system was functionally correct, but this did not ensure performance success. Testing during development typically consisted of spatializing two or three simple sound sources (`[osc~]`) in a lab environment. This ensured that the system performed the spatialization tasks, but gave no guarantees of successful performance use. Final testing came during the dress rehearsal, when ten instruments where used as simultaneous input and spatialized around the performance hall in real-time. Because the *Challenging Bodies* project incorporated performance features from many disparate disciplines, the dress rehearsal was the first time all pieces were assembled into a cohesive whole. It speaks to the professionalism and preparation of all parties that the show came together as expected, but there were a few kinks that were worked out during or immediately after the dress rehearsal.

One technical issue that was resolved during the dress rehearsal was the time delay on the system. The default time delay for `pd` is 50 milliseconds, and the longer the time delay, the larger the audio buffers and the less just-in-time processing is required by the computer system. 50 milliseconds, however, is on the cusp of perceptibility, especially when professional musicians are playing through the system in real time. All of the musicians mentioned that there was a small but perceptible delay in the playback of the sound through the spatializer. The time delay was reduced but at 10 milliseconds there was audible clicking and a reduction in performance. It was discovered that a delay of 25 milliseconds was a reasonable compromise that allowed the performers to interact with the system and still allow the system to perform well.

The performance was a success: the spatializer performed flawlessly from a technical standpoint The show itself, however, shed light on two performance issues that had been overlooked, and could not be fixed between the dress rehearsal and the show proper. These issues were the effectiveness of spatialization of multiple sources and performer interaction.

4.1 Spatialization effectiveness with multiple sources

During the testing phase, only a small number of sounds were spatialized and they proved to be easily discernible by the listeners. In the *Challenging bodies* show, however, there were ten independent sources which were often spatialized to ten different locations. Listeners were able to discern the localization of a single source and follow it around the room in sparse pieces where only a few sources were present, but in pieces where many instruments were localized at the same time, it was more difficult to discern the location of one source in the midst of the soundscape. Spatialization was most effective when only a few musicians were playing, and was especially effective during solos. At the very beginning of the show, during the initial blackout, Craig introduced the show (using his text-to-speech system) and this introduction was spatialized around the room in a very effective use of the technology.

4.2 Monitoring real time live spatialization

A key feature of performance-based sound reproduction systems is monitoring: Performers must be able to hear themselves in the context of the performance in order to be able to interact with the other performers and to fine-tune their own sound. In the *Challenging Bodies* show, the speakers were positioned behind the artists, and it was expected that this would provide sufficient monitoring, given the size of the performance space and the nature of the instruments being played. In a static system (with no spatialization) this would be the case although independent monitor mixing would be impossible. The spatialization of the source sounds added a dynamic element to the output produced by each speaker. The artists, being used to static systems, expected that the speakers would produce a consistent reproduction of their music, but because the sounds were moving around the performance space, the sounds coming from the individual speakers faded in and out. This reduced the monitoring effect of the speakers and reduced the interaction of the musicians with each other and the musical whole, although again because of the smaller performance space, the musicians were also able to interact directly with each other. The overall time delay between the performers actions and the sound from the speakers to their ears varied depending on the virtual location of the sound, but it was the change in amplitude that was most noticeable and most difficult to deal with. Delay between action and sound has an interfering effect when the delay is not correlated with the perceived distance from the sound to the ear. Since the perceived distance was also changing, there was no overt delay effect in the monitoring.

The issue of the monitoring of real-time sound source spatialization is important and difficult, since musicians are more able to interact with the soundscape if they can hear where their source has been moved to, but as evidenced in the experiences of *Challenging Bodies*, direct individual monitoring is also important. More study needs to be done on the balance between these two reinforcement techniques.

5. FUTURE WORK

The system presented herein was developed for a particular show, and as such it had some limitations that will be easily removed in future versions. The main development next in line is to implement the proposed 3-d interface, incorporating multiple views of the speaker environment, and an oblique view. The development of this 3-d interface means exploring means of manipulating sources in 3-d with a standard 2-d screen window or windows. Several implementations of 3-d object manipulation exist for other applications, and these will be studied in order to inform this system.

The second improvement that will be added to the current system is the ability to manipulate the location of the speakers. The problem with allowing the speakers to be manipulated is that they may be inadvertently moved during performance. One possibility to alleviate this problem is to use a speaker manipulation mode and a normal operation mode, with obvious visual feedback when in speaker manipulation mode (e.g. a change in background colour). Since usability theory suggests that modal interaction should be avoided unless necessary and obvious, a second option is to have a lock on each speaker what would reduce the likelihood of inadvertent manipulation.

A third improvement would be adding the ability to record source motion paths and re-play them, similar to existing spatialization systems. A related improvement would be the ability to manipulate groups of sounds, for example, by moving stereo pairs or clusters of sources.

The system was developed for musical performances, and it was shown to be a useful and powerful tool for live performances. Another area of future work will be to apply this system to other application domains such as sound effects for movies and video games, and for spatialization of sound effects for live plays or musicals. Having the ability, in a usable interface, to manipulate sound effects or foley in real-time for these applications has considerable potential. For instance, it would be quite useful to be able to produce off-stage voices in a live play. We will investigate opportunities to introduce the system to other live venues and applicaitons.

6. CONCLUSIONS

A real-time sound-source spatialization system was developed with three main aims: scalability, usability and open-source. It was implemented with a specific application target, that of the *Challenging Bodies* transdisciplinary project. The advantages that it has over existing spatialization systems are that it allows real-time spatialization of live sound sources, that it is open-source, is intentionally simple in order to enhance usability, and that it allows general speaker arrangements in vertical as well as horizontal layouts. The use of spatialization in the context of live performance was studied, and a number of issues were identified and flagged for future research. The system as developed will be enhanced to provide 3-D interaction, speaker location modification, cluster movement (movement of multiple sources in relation to each other), and implementation in other performance contexts (e.g. theater, games, virtual reality).

Although spatialization has been used in musical performance in the past, it is still just beginning to catch on as a creative aspect of musicality. By employing spatialization technologies such as SPAT, Holo-Spat, or the system described in this paper, spatialization in music can be further explored. Non-real time systems allow for an individual acting as a 'spatial composer' to later add a fixed sequence of effects to a performance, but real-time sound source spatialization as described in this paper turns this composer into a performer.

7. ACKNOWLEDGMENTS

The project was funded by the University of Regina Transdisciplinary project fund, The Natural Sciences and Engineering Research Council of Canada, the Social Sciences and Humanities Research Council of Canada, and the Canadian Foundation for Innovation. The authors would like to thank all who participated in Challenging Bodies, and in particular Doug Ramsay, Daryl Hepting, Ann Kipling-Brown, Kathleen Irwin, and John Barden.

8. REFERENCES

[1] M. Bosi. An interactive real-time system for the control of sound localization. In S. Arnold and G. Hair, editors, *Proceedings of the ICMC*, Glasgow, September 1990.

[2] M. Danks. Real-time image and video processing in gem. In *Proceedings of the ICMC*, pages 220–223, Thessaloniki, 1997.

[3] J. Harrison. Sound, space, sculpture: Some thoughts on the 'what', 'how', 'why' of sound diffusion. *Organised Sound*, 3:117–127, 1998.

[4] M. Puckette. Pure data: another integrated computer music environment. In *Internation Computer Music Conference*, 1996.

[5] C. Roads. *Computer Music Tutorial.* MIT Press, Cambridge, MA, 1996.

[6] M. B. Rosson and J. M. Carroll. *Usability Engineering: Scenario-based Development of Human-Computer Interaction.* Morgan Kaufmann, 2001.

[7] B. Truax. Composition and diffusion: space in sound in space. *Organised Sound*, 3:141–146, 1998.

Mapping with planning agents in the *Max/MSP* environment: the *GO/Max* language

Paolo Bottoni, Stefano Faralli, Anna Labella, Mario Pierro
Università di Roma La Sapienza Dipartimento di Informatica
Via Salaria, 113, 00198 – Rome, Italy
(bottoni,faralli,labella, pierro)@di.uniroma1.it

ABSTRACT

GO/Max is an agent programming language that facilitates the design of algorithms for real-time control of sound/music generation programs crafted in the Max/MSP environment. We show how software planning agents programmed in GO/Max can be used to transform abstract goal states specified by the performer in potentially complex sequences of Max/MSP control messages.

Keywords

mapping, planning, agent, Max/MSP

1. INTRODUCTION

In the context of real-time interactive musical systems, *mapping* represents the correspondence between the control space in which the performer acts and the parameter space of the system that is being controlled. The central role of mapping and the importance of having efficient methods to design mappings have been exposed by several works [1].

Several approaches for translating from the control space into the parameter space have been proposed (see section 1.1). In this paper we present our experiments on using *state models* and *planning agents* to perform such translation. The basic idea behind this approach is to represent the controlled system using a *state model*. If a state model for the system is finite, it is possible to implement a software agent that will be able to foresee the evolution of such model using a *planning* algorithm [2]: the agent will be able to produce a command sequence that leads from one model state to another goal state, provided that such transition is possible, or state its non-existence. If the model also specifies a correspondence between state transitions in the model and commands for the system, a sequence of states in the model can be used to drive the actual system.

The combination of the system model and the planning agent will thus form an *abstraction* layer between the performer and the controlled system. The performer will drive the agent by sending it requests for *goal* states; the agent will devise sequences of operators to reach those goal states; operator sequences correspond to message sequences used to steer the abstracted system (see Figure 1).

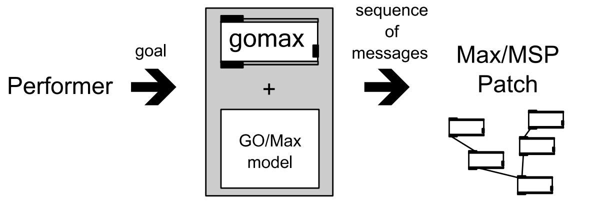


Figure 1: Performer requests a goal to the gomax agent, agent's plans to reach the goal are translated in control sequences for the Max/MSP patch.

The abstraction is obtained because of the independence between goal/operator definitions and messages which are sent to the patch. This can be particularly effective when used in conjunction with other mapping techniques which can be used to form goal state requests depending on the performers input. One example could be the mapping of a discrete set of hand gestures to a set of agent goal states.

We have experimented abstracting the control of Max/ MSP patches. To do so we developed a language (*GO/Max*, standing for Goal-Oriented Max) to describe deterministic state models of Max/MSP patches, and a Max/MSP external (`gomax`) to implement a software agent which performs planning and translates state model transitions into actual patch commands [3].

1.1 Related Work

The mapping problem has been approached in numerous ways: [4] is a comprehensive review of mapping techniques. Generally mapping is a *one-to-one*, *one-to-many* or *many-to-one* correspondence between the control space and the parameter space, where their dimension numbers generally differ. Van Nort et al. [5] have formalized mapping as a continuous function from the control to the parameter space and, representing those as subsets of a highly-dimensional Euclidean space, have analyzed the geometrical and analytical properties of that function.

Some continuous mapping algorithms such as Escher [6] use an intermediate layer of abstract parameters to translate between the control space and parameter space. The recent MnM toolbox [7] uses matrices to represent mappings and provides methods to build them iteratively. Many-to-one mappings have also been approached as a *pattern recognition* problem to which neural networks have been applied (most recently in [8]). The recent proposal of `ChucK` ([9], [10]) has exposed the central role played by the programming paradigm in the context of high-level computer music languages, and specifically in the definition of complex mappings.

2. PLANNING AND MAPPING

The task of *planning* is generally summarized as "finding a sequence of actions that will achieve a goal"[11].

A *deterministic state model* is defined as a set of states S, a set of actions A, and a transition function $f : S \times A \rightarrow S$ which describes how actions map one state into another. If the system to which actions are applied is represented by a deterministic state model, the planning problem is a *deterministic control problem* [2].

An instance of a planning problem can be described using a formal language: *STRIPS* [12] is the classic example on which also GO/Max is based. A STRIPS description of a planning problem describes a deterministic state model in which the action costs are all equal [2]. If the agent environment is "fully observable, deterministic, finite, static (change happens only when the agent acts), and discrete (in time, action, objects, and effects)" then the planning task is called *classical planning* [11].

A planning problem instance described in GO/Max is a tuple $P = \langle L, A, I, G \rangle$ where:

L is a set of literals (*atoms*), so that every state s of the state model is a subset of L: $\forall s \in S, s \subseteq L$. In a GO/Max program, literals can be defined explicitly

```
literal someBooleanCondition;
```

or they can be implicitly specified using variables on finite domains. Domains can be either integer intervals or explicit lists of strings:

```
domain intDomain: 0..8;
domain listDomain: {a, b, c, d};
var someVariable: intDomain;
```

A is a set of *actions*. For every $a \in A$ we define the *preconditions* set $pre(a) \subseteq L$. An action is applicable in a state s if $pre(a) \subseteq s$; the *postconditions* describe the effects of the action and are represented by a set pair $post(a) = (add(a), del(a))$ where $add(a) \subseteq L$ and $del(a) \subseteq L$. The state s' resulting from the execution of the action a in a state s is defined as $s' = (s \cup del(a)) \backslash add(a)$. Extending the syntax of the traditional STRIPS language, every operator is described in terms of:

- *parameters*, as local GO/Max variables,
- *preconditions*, either expressed as literals or as boolean conditions on the variables,
- *postconditions*, either expressed as literals, negated literals (using the `!` operator) or as GO/Max variable assignments (using the `:=` operator)
- a Max *message*, which will be sent during the operators execution. Message syntax is a subset of the one used in the standard Max *message box* object, so a message can be composed of integers, floats, bang and symbols. Several messages can be sent sequentially by separating them with a comma. It is not possible to use the semicolon to send the message to a specified receiver object. An expression evaluation operator `&( )` has been added to create integer and symbol values from GO/Max variables with integer and string-list domains respectively, as well as expressions containing integer variables.

Here is an example of an operator declaration:

```
operator someOperator( someParameter: intDomain )
pre( someBooleanCondition, someVariable < 3 )
out( set &(someVariable) )
post( someVariable := someParameter );
```

This operator will allow the agent to change the value of `someVariable` to any of the values in `intDomain`, provided that the `someBooleanCondition` literal is present in the current state, *and* that the current value of `someVariable` is less than 3. Every time an agent will use this operator, it will choose a target value in `intDomain`, output the Max message `set` followed by the current value of the variable `someVariable`, and finally update the variable's value to the chosen target value.

$I \subseteq L$ is the initial state, declared explicitly using the `state` construct:

```
state(someBooleanCondition, someVariable=0)
```

In this case, the literal `someBooleanCondition` is included in the start state, and the initial value of `someVariable` is zero.

G is the goal description, in terms of a set pair (G_{pos}, G_{neg}) where $G_{pos} \subseteq L$ and $G_{neg} \subseteq L$: a state s is a goal state if $G_{pos} \subseteq s$ and $G_{neg} \cap s = \emptyset$. Goal states are specified at run-time, sending the `goal` message to an agent (see 2.1). For example:

```
goal someBooleanCondition someVariable=100
```

Every literal in a GO/Max model corresponds to a boolean condition in the agent environment. A *closed world assumption* is adopted, so that if a state does *not* contain some literal l, then the boolean condition associated to l is *assumed* to be false. Every state is then a complete description of the agent environment.

Given a STRIPS-like problem description, a *planning agent* autonomously finds an action sequence leading from the initial state to the goal state, if such sequence exists. A problem description in a suitable language, such as GO/Max , can therefore be used as a declarative-paradigm agent programming language. This allows the programmer to specify just *what* should be done instead of *how* it should be done, and assures that all states reached during program execution respect the model declaration. By declaring the operations and the initial state, a GO/Max program potentially describes different execution sequences for them, thus being equivalent to several iterative programs. Compared to existing procedural-code features of Max (Javascript, Java, RTCmix), this adds compactness and flexibility to programs, especially in mapping applications where the agents can automatically devise a new action sequence if a new goal request occurs.

States in the model can be mapped to configurations of parameters in continuous data spaces, so that other interpolation techniques can be used to translate from one configuration to another.

Note that since the actions are *assumed* to have deterministic effects, the agent can operate without reading any feedback from the actual state of the system (Fig. 1). Also, classical planning relies on *atomic* plan generation and execution: the environment does not change while the agent is calculating the plan (see 2.2).

The classical planning problem is a well-known AI topic and has been approached with many different algorithms ([11], [13], [14], [15]). STRIPS planning is a highly complex computational task [16] . Nevertheless, planning is a very powerful tool even for models with small state spaces, and is used in many applications which include robotics and computer games.

2.1 The *gomax* software agent

The GO/Max patch model can be compiled by a `gomax` software agent which operates in the patch itself as an external object. A model is compiled with the message `parse (filename)`.

A performer interacting with the patch will be able to issue requests for goal states to the agent using the `goal` message. A goal state will be described in terms of literals in the currently compiled model. When a request for a goal state is made, since every operator is associated with a Max message, the computed sequence of operators is associated with a sequence of Max messages that are sent out via the agent's outlet. While the goal state is specified by the performer, the way in which it is reached will be determined by the agent based on the GO/Max model that has been compiled and on the search algorithm used.

2.2 Patches and state models

As we can see from the scheme in Figure 1, `gomax` agents currently do not receive any feedback from the patch, so it is *assumed* that the actual state of the patch reflects the state of the GO/Max model which is used for planning. This obviously limits the structure of patches which can be represented by a GO/Max model, and it is now needed because the algorithms used to implement planning in the agent do not allow for a non-deterministic effect of actions. Building a model whose changes reflect the patch changes is the model designer's responsibility. Also, since classical planning is used, the modeled patch is expected to change only when the agent acts.

Max/MSP has a well-defined semantics for its patches, where the order of activation of the connections between objects is based on the objects positions and right-to-left inlet ordering [17].

If a patch contains a purposely non-deterministic section, such as a random number generator, that section can be either ignored in the GO/Max model or used to generate goal requests for the agent so that the internal state of the model can be re-aligned with the actual state of the patch.

2.3 Planning algorithms and real-time performance

Since there is no optimal algorithm for planning [11], it is possible to choose which algorithm the `gomax` agent should use by means of the `search` message. This will also determine the behavior of the agent when multiple equivalent plans are found: while normal algorithms will account for operator declaration order to choose which plan will be used, the provided "non-deterministic" variants will choose a plan at random.

In the present version the available algorithms are a simple breadth-first search and a heuristically-guided A-star search. Both operate in the state space and are provided in deterministic and non-deterministic variants. The A-star search is a modified version of the algorithm described in [13], the details of the algorithm are described in [3]. More search algorithms will be added as new versions of the agent will be released.

The current algorithms allow real-time use of the agent with models containing up to about 200 states, even if this greatly depends on the structure of the model. With

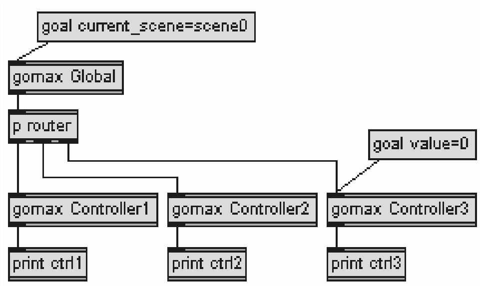


Figure 2: Using multiple agents

a model of this size, plan generation time is under 0.1 seconds using an Intel Pentium 4 1.8Ghz processor. It is expected to optimize plan generation time further in the next agent versions.

However, even with very simple models, using an abstraction layer is a fast way to implement mappings which would be difficult to realize with other techniques. Subdividing a large model into independent sub-models can reduce its computational cost (see 2.5).

2.4 Models and time representation

Once a plan is determined, a gomax agent fully outputs the corresponding sequence of messages *atomically*. The resulting messages can then be, for example, stored in a buffer and read every n milliseconds, so that no explicit notion of time is needed in the model. A different model could output messages with a specific timestamp, thus making the buffer a sequencer. The GO/Max language does not offer any *predefined* construct to specify an operator execution order. Time can be represented in the model used, so that, for instance, certain precedences between operators are established ("always choose a note length before playing a note") or that messages output by the agents carry a timestamp, which is the case when the "abstracted" patch has a sequencer-like behavior.

2.5 Multiple agents

Max messages associated with GO/Max operators are arbitrary and can be used to issue `goal` messages to other `gomax` agents, thus decomposing a large model in smaller independent models, each handled by a separate agent.

The different agents coordinate their operations using the Max/MSP depth-first message handling scheme [17]. A plan generation started from an agent will not return control to the patch until all sub-agents have either generated a plan or returned an error. Also, the plan-starting agent will not update its internal state nor continue the execution of its plan if, at some point during plan execution, one of the sub-agents has reported an error.

In the example of Figure 2, a system containing three independent parameters has been modeled using three separate GO/Max models/agents, and a "master" agent which sends individual goal requests to each. If the parameters are independent, this distributed model offers the same functionality as a single model containing all three parameters, but uses a smaller state space. For example, if each of the three parameters has a 0-127 range, a single model will contain 127^3 states while a distributed model will have only $127*3$. This kind of model decomposition is often a feasible solution to allow real-time control of large patches. Using different agents, every agent offers a different abstraction layer to the performer. Since agent op-

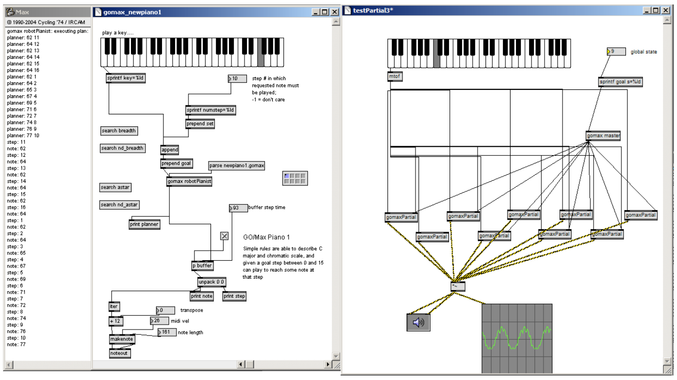


Figure 3: Meta-Piano and Additive patches

erations are always coordinated, the performer can freely switch between the different abstraction layers to control the system in different ways.

3. EXAMPLES

Since GO/Max models use standard Max messages, they have a broad range of applications (music, graphics, etc.).

3.1 Meta-Piano

The model used in this patch represents a 16-step buffer and a writing position inside it. Operators output a buffer step value (`key` variable) and its position (`numstep`), rolling back to zero when the last step is reached. The `key` variable value is a MIDI note pitch, and the operators change its value following rules such as chromatic/major/minor scales, predefined note successions, etc.. When the performer chooses a target note pitch and a target buffer position (Fig. 3) the agent will find a succession of notes that follows the rules specified by the operators and ends exactly on the required step. An example operator is:

```
var key: 36..83; var numstep: 0..15;
operator SemitoneUp
    pre ( )
    out ( &(key + 1) &(numstep + 1) )
    post( key := key + 1,
          numstep := (numstep + 1) % 16 );
```

3.2 Additive Synthesis

Here different agents control partials independently, each using a different model, constructed so that the operators change the agent's state and output fixed values for the partial's frequency, phase and amplitude (Fig. 3). A buffer is used to output agent messages at a fixed rate, and *line˜* objects smooth transitions between values. A further "master" agent can send goal requests to all subagents as shown before (see 2.5). When the performer asks one partial or master-agent to reach a goal state, the activated agent will determine a sequence of its model operators which will respectively correspond to a sequence of partial parameters, or to a sequence of goal requests for all the partials simultaneously. Control is therefore abstracted by generating goal requests from an input parameter, indicating the target sound state. Here we show a short model for one of the partial-agents, which outputs constant phase and volume values and links the partial's frequency to the model's state number, establishing a simple state transition scheme:

```
var pState: 0..10;
operator SwitchToNextState
pre ()
out ( &(pState*50) 0.0 1.0 )
post( pState := (pState + 3)%10 );
state ( pState=1 )
```

When a target value for `pState` is specified, the partial's frequency will thus change in a model-dependent way.

4. FUTURE WORK AND CONCLUSIONS

GO/Max brings the benefits of a high-level, declarative language to Max/MSP, and is currently in beta stage for its 1.0 release. On the language side, an interesting development would be a visual editor to construct a specific subset of GO/Max models. This could be an evolution of the traditional one-to-one *MIDI Learn* facility. On the agent side, we plan to work on an explicit feedback mechanism and on asynchronous (non-blocking) plan generation. We also plan to port GO/Max to other computer music languages such as ChucK or OSC. To achieve this, a suitable state model must be determined as well as efficient planning algorithms for that model.

5. REFERENCES

[1] A. Hunt M. Wanderley and M. Paradis. The importance of parameter mapping in electronic instrument design. In *Proc. of NIME-02*, pages 149–154, 2002.

[2] B. Bonet and H. Geffner. Planning and control in artificial intelligence: A unifying perspective. *Applied Intelligence*, 14(3):237–252, 2001.

[3] M. Pierro. Sistemi di controllo per la generazione musicale, Tesi di Laurea, Corso di Laurea in Informatica, University of Rome "La Sapienza", 2005. *http://hci.uniroma1.it/multimedialab/*

[4] M. Wanderley and M. Battier, editors. *Trends in Gestural Control of Music.* IRCAM, 2000.

[5] D. Van Nort, M. Wanderley and P. Depalle. On the choice of mappings based on geometric properties. In *Proc. of NIME-04*, pages 87–91, 2004.

[6] M. M. Wanderley, N. Schnell, and J. Rovan. ESCHER—Modeling and Performing composed Instruments in real-time. In *IEEE Systems, Man, and Cybernetics Conference*, October 1998.

[7] F. Bevilacqua, R. Müller and N. Schnell. MnM: a Max/MSP mapping toolbox. In *Proc. of NIME-05*, pages 85–89, 2005.

[8] A. Cont, T. Coduys and C. Henry. Real-time gesture mapping in pd environment using neural networks. In *Proc. of NIME-04*, pages 39–42, 2004.

[9] G. Wang and P. R. Cook. On-the-fly programming: Using code as an expressive musical instrument. In *Proc. of NIME-04*, pages 138–143, 2004.

[10] G. Wang, A. Misra, A. Kapur, P. R. Cook. Yeah, chuck it! => dynamic, controllable interface mapping. In *Proc. of NIME-05*, pages 196–199, 2005.

[11] S. Russel and P. Norvig. *Artificial Intelligence.* Prentice Hall, 1995.

[12] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In *Proc. of the 2nd IJCAI*, pages 608–620, 1971.

[13] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *J. Artif. Intell. Res. (JAIR)*, 14:253–302, 2001.

[14] H. A. Kautz and B. Selman. Planning as satisfiability. In *Proc. of ECAI'92*, pages 359–363, 1992.

[15] S. M. LaValle. *Planning Algorithms.* Cambridge University Press, 2006.

[16] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1-2):165–204, 1994.

[17] Cycling '74. *Max Reference manual*, 2004.

Towards a Virtual Assistant for Performers and Stage Directors

Alain Bonardi
MSH-Paris Nord
UMS 2553
4, rue de la Croix-Faron
93210 La Plaine Saint-Denis
alain.bonardi@wanadoo.fr

Isis Truck
MSH-Paris Nord
UMS 2553
4, rue de la Croix-Faron
93210 La Plaine Saint-Denis
truck@ai.univ-paris8.fr

Herman Akdag
LIP6 - UPMC
UMR 7606
8, rue du Capitaine Scott
75015 Paris
Herman.akdag@lip6.fr

ABSTRACT
In this article, we present the first step of our research work to design a Virtual Assistant for Performers and Stage Directors, able to give a feedback from performances. We use a methodology to automatically construct fuzzy rules in a Fuzzy Rule-Based System that detects contextual emotions from an actor's performance during a show.

We collect video data from a lot of performances of the same show from which it should be possible to visualize all the emotions and intents or more precisely “intent graphs”. To perform this, the collected data defining low-level descriptors are aggregated and converted into high-level characterizations. Then, depending on the retrieved data and on their distribution on the axis, we partition the universes into classes. The last step is the building of the fuzzy rules that are obtained from the classes and that permit to give conclusions to label the detected emotions.

Keywords
Virtual Assistant, Intents, Emotion detector, Fuzzy Classes, Stage Director, Performance.

1.INTRODUCTION
Directing is a complex task notably when productions contain a part of improvisation. However the performers of these productions need to be directed since the improvisation always follows some rules.

Thus, computers may probably be of great help in assisting the stage director and/or the performer. Indeed, we think it is important to conceive several tools to help the stage director in his task of actors’ performance supervision. The tool we propose here is a kind of assistant that gives a visual representation – through a graph – of a set of complex data for the exploration and observation of a show. It also permits to understand better the creation and execution of the show. One important point is that, as every computer program, the assistant must be deterministic and systematic, i.e. it should always give the same results for a given entry and it must look over the whole data. Like high-level sportsmen that have tools to analyze, correct and improve their gesture, we want to propose a tool to assist the creative artists, to let them better understand the phases of a performance and to help them in their creation process.

For our practical experiments we have chosen to work on a digital opera called *Alma Sola* written by Bonardi and Zeppenfeld where a performer plays (sings and moves) different blocks from different *universes* (such as Prologue, Love, Pleasure, etc.). *Alma Sola* is an open form of opera [1]. The performer embodies a feminine Faust andwanders through the various *universes* splitinto blocks. She therefore interprets an opera playlist that she selects during the show itself. For instance, a performance can be : Love-3, then Wealth-5, then Pleasure-3, etc. Thanks to Hidden Markov Models, the computer offers continuations to the performer and suggests the next block to be performed (cf. Figure 1).

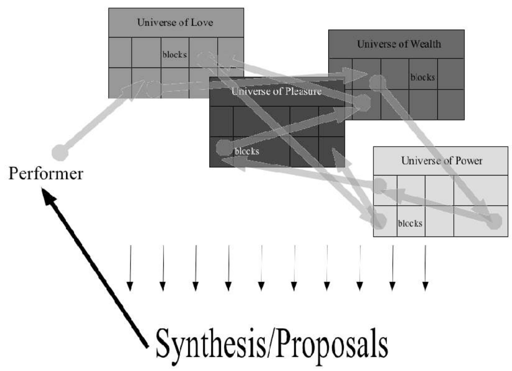


Figure 1. Navigation in Alma Sola digital opera.

We retrieve the actor’s performance on video files and look for emotions in the data. A distinction must be made between intent and emotion: indeed intent corresponds to the conscious part of the emotion. Thus the assistant enables the comparison between the performer’s intent and the rendered emotions in the context of *Alma Sola*. It is widely known that fuzzy logic offers good tools to deal with such subjective concepts [10], emotions here, this is why we shall use a fuzzy rule-based system (FRBS) to detect the performer’s emotions.

The article is organized as follows: first we give an overview of the existing research about assistants in art, then we describe our assistant and notably the way we retrieve emotion descriptors. In the third section we show how the descriptors are partitioned in classes. This step is necessary to build correctly the fuzzy rules that form the inference system. Finally, section 4 concludes this study.

2.Research about performer's assistants

In a way, research about performer’s assistants has existed for centuries. One immediately thinks of the use of mirrors in dance. At the time when mirrors become common place in Europe (Renaissance), the first treatise about dance is released by Thoinot Arbeau in 1589.

The issues raised are still the same :

- first point, to be able to state an ideal prescription of the performance. This generally starts from scores and notations, which have been developped for centuries in music and dance. They include both implicit (you have to play an F, but the score does not tell how to do it) and explicit (cross hands when playing the piano, for instance) gestures to be achieved. From these structured indications, a dancer or a musician tries to infer some of the author’s intentions[5]. His/her representation of the author's intentions become the ideal prescription of the performance.
- second point, to be able to measure a kind of difference between the realized performance and its ideal prescription. For instance, in his approach of “Virtual scores”, Manoury has considered [7] in his pieces for solo instrument and live electronics the computation of this difference (*Jupiter* for solo flute and live electronics, 1987, *En Echo* for voice and live electronics, 1991) as a basis to generate electronic sounds.
- third point, which is correlated to the second one, is the approach to measure this difference. The first approach consists in directly measuring various aspects of the performance, using captors in a broad meaning [11]. Various captors are nowadays available: video camera, wireless microphone, ultrasound device, carpet detectors, digital compass, etc. The second approach consists in collecting human appreciations of this difference between the prescription and the realization. Composer Roger Reynolds has for instance imagined psychological testing [9] with listeners for this piece *The Angel of Death*.

From a technical point of view, dedicated software platforms have been developped. Recently, Camurri and his team have developed the first robust platform for the analysis of gestures and consequently of performer’s emotions. It is named EyesWeb [3], [6]. It is not based on captors implemented on the performer’s body (with heavy batteries and radio transmission), but on video capture with a static shot. It is based on a graphical language that implements many descriptors of gestures: quantity of motion, stability, etc. EyesWeb has become a worldwide standard for performance analysis.

Roughly ten years before, the Ircam institute (and Cycling 74 company) had developed the real-time digital sound analysis and synthesis platform to complete Max software. It is named MSP (Methods for Sound Processing) and is now included in Max software (Max/MSP). This software is a worldwide standard in real-time sound analysis. At the present time the state-of-the-art consists mainly in inferring a few emotional states from the raw data delivered by EyesWeb and/or Max/MSP.

For instance, a project relatively similar to ours is developed by Friberg and his team. They have conceived a real-time algorithm to analyze emotional expression in musical performance and body movement [4]. In the framework of a game named “Ghost in the cave”, the player has to express di•erent emotions using his/her body or his/her voice, and these emotions are the input values of the software. They use EyesWeb to recover body movements and sound descriptors (sound level, instant tempo, articulation, attack rate, high-frequency content). But in this game, there is an immediate feedback, and the player has to move constantly and talk until the software reacts according to his/her wishes. Our aim is not the same, i.e. the assistant adapts to the performer and not the contrary.

3.Project Description

As explained above, we have chosen to work on the interactive opera *Alma Sola* written by A. Bonardi and C. Zeppenfeld. Two dissimilar scenes have been extracted to be used as “sample scenes”: they are the chanted Prologue (which is improvised, unwritten) and the Love Universe (which is “strictly written”). The performer (a singer/dancer, here) is filmed by a camera in wide and static shot (cf. figure 2, left) and his/her voice is recorded in a separate file, in order to handle both sources of data separately. The project is centered on two main phases: the acquired scene processing (with the performer) and the sound capture processing. In this article we focus on the first phase, where EyesWeb has been used (through a dedicated patch we have written) to analyze accurately, understand and exploit non-verbal expressive gestures. Several parameters can be extracted from the video file: quantity of motion, stability, motion duration, pause duration in a scene, contraction index and surface of the performer in the image, convex hull of the body silhouette, velocity, acceleration, etc. This patch construction step is not trivial and represents hours of tests with sample videos and it implies a concertation with the stage directors.

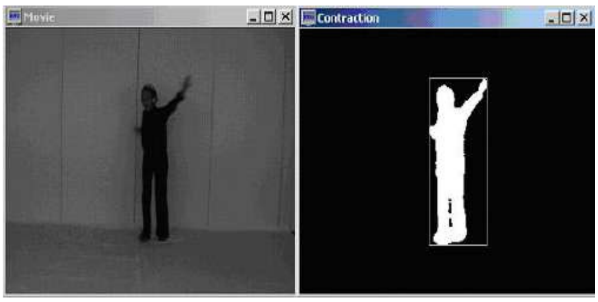


Figure 2. Left, an image taken form the video file; right, the corresponding convex of the body silhouette (performer : Claire Maupetit).

First of all, the background of the image must be removed — using the difference between two frames — in order to keep only the performer’s movement, since the camera is static. Next, the parameters can be easily extracted because we are sure they only concern the performer, and not the background. Here is the description of some of the most interesting parameters for our problem (Table 1 shows some of them). The quantity of motion is computed by the number of pixels changing position between two instants (the *white pixels* in figure 2); the convex hull of the body silhouette is the bounding rectangle of the *white pixels* (cf. figure 2); the stability is the ratio of the height of the silhouette’s center of gravity on the length of the segment connecting the lower points of the silhouette; the contraction index is the ratio of the silhouette's surface over the surface of the convex hull. The stability is an important descriptor since it gives good insight on whether the performer is near the ground or not, i.e. whether the performer puts himself at risk or not. The contraction index is also very useful and reflects whether the performer is effusive or not.

Table 1. Descriptor definitions.

Descriptor	Interval	Intuitive description
Movement duration	$[0, +\infty[$ in ms	
Pause duration	$[0, +\infty[$ in ms	
Contraction surface	[0,nb of pixels in the video[	Quantity of pixels representing the performer in the image
Contraction index	$[0, 1]$	Large: performer's stance is open Medium: performer's stance is normal Small: performer's stance is closed
Contraction matrix coord.	$x, y \in$ [0,nb of pixels in the video[	Performer location in the image
Stability	$[0, 2.34]$	Low: performer is close to the ground, legs spread High: performer is standing, legs tight
Quantity of movement	$[0, 1]$ 0: no movement; 1: all parts of the silhouette have moved	Speed of movement and displaced mass
Center of gravity	$x, y \in$ [0,nb of pixels in the video[	

Choosing these parameters judiciously (called video descriptors) allows us to compute various aggregations of each set of values for each descriptor. The chosen aggregators are: partial and general means, standard deviation, covariance, etc. Then, to a meta-level, we characterize and categorize the sequences of each scene thanks to an FRBS. The number of categories used in this opera is five (Sleepy, Angry, Happy, LoveBeliever (could be also called Effusive), LittleEffusive). Figure 3 sums the whole process up. (We capture 25 frames per second, i.e., 25 values per second for each gesture descriptor.)

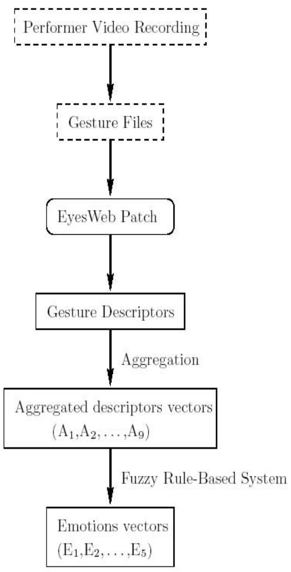


Figure 3. Our system components.

As can be seen in Figure 3, once the descriptors are extracted from the video files, they are aggregated in order to give a description of each emotion we want to recognize. Figure 4 shows graphically the results for the Universe of Love.

The next step is to classify in partitions the aggregation results. For example, when the performer expresses an emotion such as happiness, he/she moves a lot. But this is not always easy to guess even if the emotions share some general patterns, event in the restricted context of *Alma Sola*.

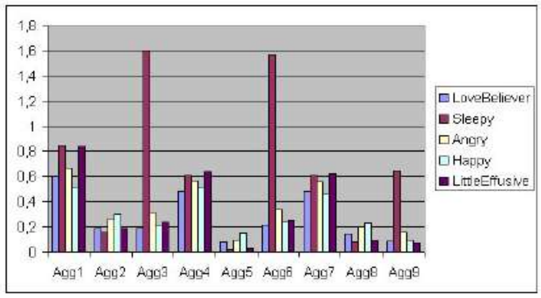


Figure 4. Aggregated descriptors for each emotion in the Universe of Love.

The fuzzy partitioning allowing the construction of the rules is now described.

4.Building the Fuzzy Rules

Fuzzy partitioning needs to spread the various values taken by the aggregation results suitably. However a simple uniform distribution of the classes on the axis is not appropriate since sometimes small, other times average or in other cases big variations lead to an emotion change, depending on the aggregated vectors. Like Martinez & al. in [8], we propose a categorization depending on the data distribution. Five classes are considered: Very Low, Low, Average, High and Very High values (denoted *VL*, *L*, *A*, *H*, *VH*).

Table 2. Classification for the Universe of Love.

Emotions	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
LoveBel.	*L*	*A*	*L*	*L*	*A*	*L*	*L*	*A*	*L*
Sleepy	*H*	*L*	*H*	*H*	*L*	*H*	*H*	*L*	*H*
Angry	*A*	*H*	*L*	*A*	*A*	*L*	*A*	*H*	*L*
Happy	*L*	*H*	*L*	*L*	*H*	*L*	*L*	*H*	*L*
LittleEff.	*H*	*L*	*L*	*H*	*L*	*L*	*H*	*L*	*L*

In the software we propose (see section 5), we also offer the possibility to use L-R fuzzy numbers exclusively (cf. figure 5) or L-R fuzzy intervals exclusively.

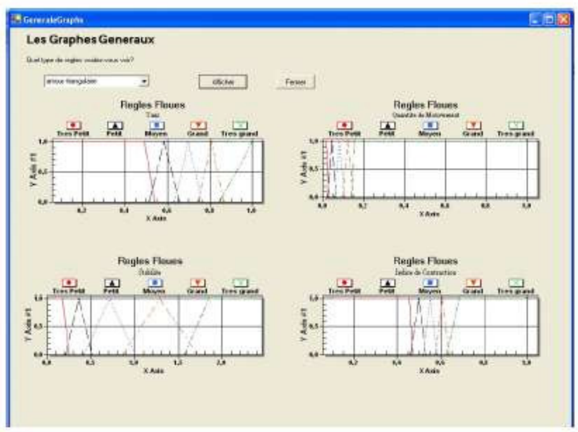


Figure 5. The L-R fuzzy numbers used in the software.

Finally, it is easy to establish the fuzzy rules according to Table 2, one rule per line. Here is an example for the Love Universe:

Rule 1:

If A_1 is L & A_2 is A & A_3 is L &
A_4 is L & A_5 is A & A_6 is L &
A_7 is A & A_8 is L & A_9 is L
Then the emotion is LoveBeliever

5.Application

The application we have developed implements the concepts explained above. We have worked on two universes: Love and Prologue. The performer has played several times each universe and the software has tried each time to detect the emotions perceived. The video files obtained for the performances are split into smaller files that are given to the software. After having chosen the fuzzy subsets that will be used for the partitioning (cf. Section 4), the values of the aggregated descriptors are displayed and a graphical result is also proposed. Figure 6 shows that both Love Universe and Prologue have rather been performed with Sleepy emotion.

The results are better (i.e., the results for the emotions are more clearcut) using partitioning with both L-R fuzzy numbers and L-R fuzzy intervals because they adapt to the level of precision, depending on the values retrieved from the video file.

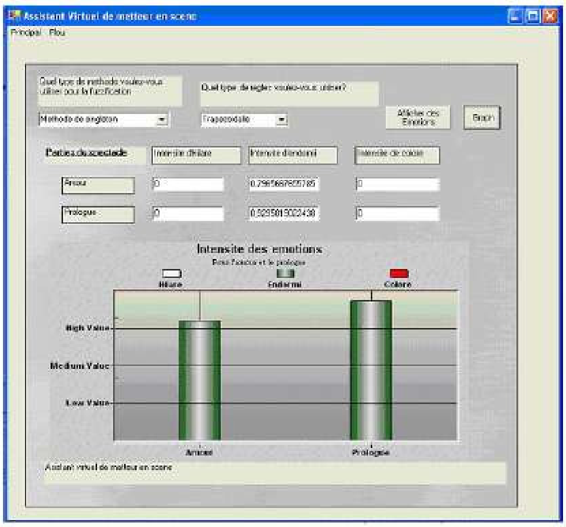


Figure 6. A screenshot of the software.

6.Conclusion

In this paper we have presented a system that is able to give clues to the stage director in order to evaluate a performer's rendition. This is done thanks to a fuzzy rule-based system that detects the actor's emotions during a performance show. One originality is the way we construct the fuzzy classes when partitioning the universes before the rule construction. They are dynamically built according to the values characterizing the performance.

As a future work, we will try our assistant on other test sets, i.e. with more records from *Alma Sola* (with the same performer or not) but also with records from other shows. Moreover, it would be very interesting to include a back propagation in the software: when an unexpected emotion is detected, the assistant should suggest modifications of his behaviour to the performer in order to obtain best results during the next detection.

Concerning *Alma Sola* opera itself, we can also imagine in the future that the assistant could be used to classify the blocks performed according to the detected emotions and then contribute to the design of the open form.

7.ACKNOWLEDGMENTS

We thank Adrien Revault d'Allonnes and Murat Goksedef for their collaboration and help on this work. This project was funded by the Maison des Sciences de l'Homme Paris Nord.

8.REFERENCES

[1] A. Bonardi & F. Rousseaux. New Approaches of Theatre and Opera Directly Inspired by Interactive Data-Mining. In *Proceedings of the Int. Conf. Sound & Music Computing (SMC'04)*, pages 1-4, Paris, France, 2004.

[2] B. Bouchon–Meunier. La logique floue et ses applications. Addison–Wesley, 1995.

[3] A. Camurri, M. Ricchetti & R. Trocca. EyesWeb - toward gesture and affect recognition in dance/music interactive systems. In *Proceedings of the Int. Conf. IEEE Multimedia Systems*, Firenze, Italy, 1999.

[4] A. Friberg. A fuzzy analyzer of emotional expression in music performance and body motion. In *Proceedings of Music and Music Science*, Stockholm, 2004.

[5] K. Kahol, P. Tripathi & S. Panchanathan. Automated Gesture Segmentation From Dance Sequences. In *Proceedingsof the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, pages 883- 888, Korea, 2004.

[6] E. Lindström, A. Camurri, A. Friberg, G. Volpe & M.-L. Rinman. Affect, attitude and evaluation of multi-sensory performances. In *Journal of New Music Research*, 34(1): 69-86, Taylor & Francis, 2005.

[7] P. Manoury & M. Battier. Les partitions virtuelles. In *Ircam documentation*, Paris, 1987.

[8] L. Martinez, J. Liu, Da Ruan & J.B. Yang. Fuzzy Tools to Deal with Heterogeneous Information in Engineering Evaluation Processes. In *Information Sciences*, In Press, 2006.

[9] R. Reynolds. Epilog:Reflections on psychological testing with The Angel of Death. In *Music Perception* n ° 22, pages 351-355, 2004.

[10] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) — an outline. In *Information Sciences*, Elsevier, 172 (1-2), pages 1-40, 2005.

[11] C. Zeppenfeld. L'acteur face aux technologies, Master dissertation, University Paris 3, 2004.

Students' projects of interactive media-installations in SUAC

Yoichi Nagashima

Shizuoka University of Art and Culture
1794-1 Noguchi-cho, Hamamatsu,
Shizuoka 430-8533, Japan
+81-53-457-6215

nagasm@computer.org

ABSTRACT

This is a studio report of researches and projects in SUAC (Shizuoka University of Art and Culture). SUAC was founded in April 2000, and organized NIME04 as you know. SUAC has "Faculty of Design" and "Department of Art and Science" and all students study interactive systems and media arts. SUAC has organized Media Art Festival (MAF) from 2001 to 2005. Domestic/overseas artists participated in SUAC MAF, and SUAC students' projects also joined and exhibited their works in MAF. I will introduce the production cases with interactive media-installations by SUAC students' projects from the aspect "experiences with novel interfaces in education and entertainment" and "reports on students projects in the framework of NIME related courses".

Keywords

Interactive Installation, Sensors, Media Arts, Studio Reports

1. INTRODUCTION

SUAC (Shizuoka University of Art and Culture) was founded in April 2000 [1]. Hamamatsu City, Shizuoka Prefecture, where SUAC is located, is conveniently situated almost midway between Tokyo and Osaka. Hamamatsu has numerous companies globally famous for their advanced technologies [Yamaha, Roland, Kawai, Suzuki, Honda, etc.], and many cultural facilities in harmony with nature. SUAC is located in the center of Hamamatsu City. Museums and other cultural facilities are already clustered in this district. SUAC has two faculties and six departments - the Faculty of Cultural Policy and Management (Department of International Culture, Department of Regional Cultural Policy and Management and Department of Art Management) and the Faculty of Design (Department of Industrial Design, Department of Art and Science and Department of Space and Architecture).

The 2004 International Conference on New Interfaces for Musical Expression (NIME04) was hosted by SUAC. The conference consisted of 3 full-day event where research papers, demos and performances were presented that correspond to the state-of-the-art concerning new interfaces for musical expression (Figure 1) [2].

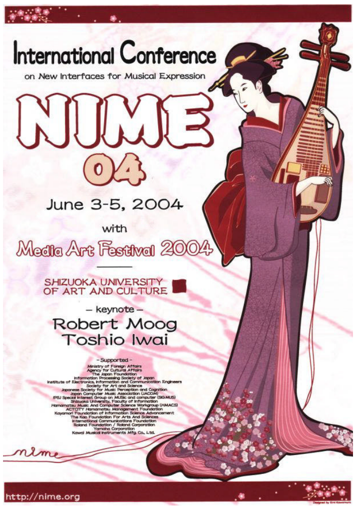


Figure 1: NIME04

2. Shin-Kai (May 2000)

The project "Shin-Kai" [forest and sea] (Fig.2) was an installation exhibited as the first collaboration of students and teachers. This work was opened to the public in the event after two months of the SUAC establishment, so all students were the first graders. The infrared sensors were built in the wood poles, so sounds and graphics were changed in real time with people walking around.

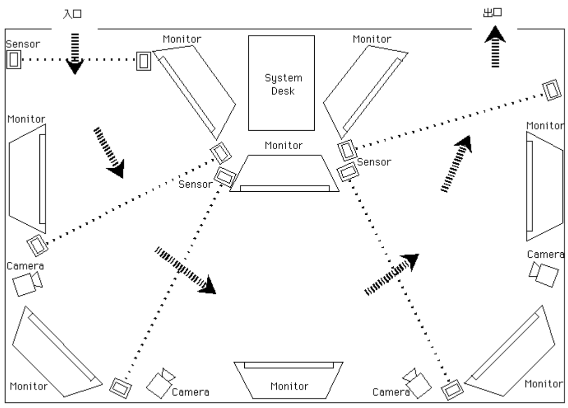


Figure 2: Floor plan of "Shin-Kai"

3. Kazatora (December 2000)

This project is the first case opened to the public as students only. One big pinwheel (Fig.3) and three small pinwheels reacted to the amount of light with the solar battery sensor and rotated according to MIDI information. This work was exhibited not as an installation work exhibited in the gallery but as a performance with seven student performers on stage. All of the production of the pinwheels, the Max/MSP patch development and the composition was produced only by students of the first grader.

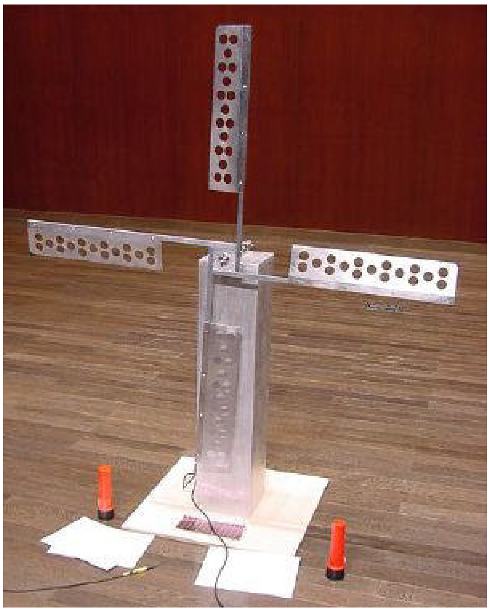


Figure 3: Installation of "Kazatora"

4. Ki-Gen (May 2001)

The project "Ki-Gen" [origin] (Fig.4) was an installation exhibited in the event of 1st birthday of SUAC, and all students were new first graders. The sound and CG were changed in real time with people walking between four plasma displays and three projected screens with the infrared sensors. People could touch objects as touch-sensors or affect CCD camera as visual-sensors.

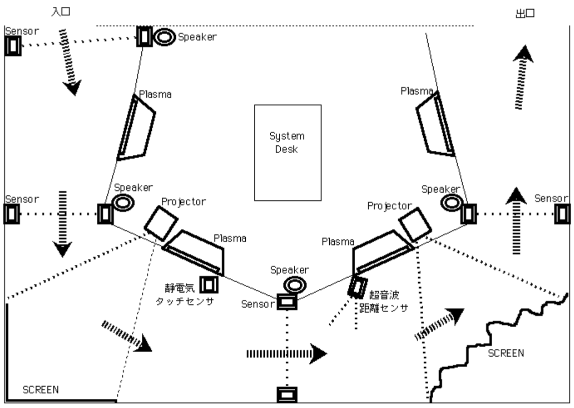


Figure 4: Floor plan of "Ki-Gen"

5. MAF2001

In August 2001, we organized first SUAC MAF (Media Art Festival) 2001[3]. MAF2001 had the following contents : (1) two Live Concerts with 12 composers, (2) Symposium of IPSJ (Information Processing Society of Japan), (3) Installation Gallery exhibiting 15 works, (4) Movie theater with 14 works, (5) CG Gallery exhibiting 12 works. Here, I introduce SUAC students' four projects as follows.

5.1 Tetora (August 2001)

This work was created by four students of the second grader. If people shake hands with the "wrist" object made of silicon rubber, the expression of the face in the display screen changed with strange sounds (Fig.5).

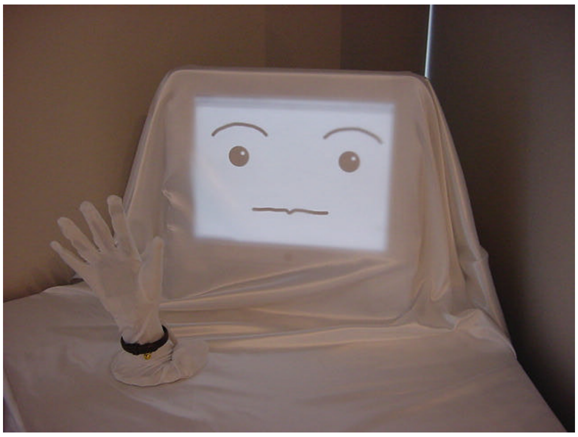


Figure 5: "Tetora"

5.2 Hachi (August 2001)

This work was also created by four students of the second grader. When people were detected by ultrasonic sensors with eight different directions, the table turned and a lot of bees stared him/her in the front (Fig.6).

Figure 6: "Hachi"

5.3 Shocking (August 2001)

This work was also created by four students of the second grader. The title "Shock" means "Eating" in Japanese. When people pulled any codes from the ceiling object, "eating/cooking" images appeared on the screen with generating "eating/cooking" sounds. (Fig.7).

Figure 7: "Shocking"

5.4 Happy Maru (August 2001)

This work was also created by five students of the second grader. The title "maru" means "round" in Japanese. They produced some boxes with a round hole (Fig.8). People could watch "happiness" by peeping at the hole of a certain box. People could hear "happiness" from another box's hole. People could see a beautiful scenery through a kaleidoscope by peeping at the hole of another box.

Figure 8: "Happy Maru"

6. INTERCOLLEGE (December 2001)

In the computer music community in Japan, there is "Intercollege Working Group" that consists of the relating universities. SUAC has participated in "Intercollege Computer Music Concert" sponsored by the "Intercollege Working Group" every year since 2000. Motozono Chisako of the second grader exhibited her installation work in 2001 (Fig.9). This work's shape was as if a big egg made of Glass Fiber, and many small stones were put on surroundings. If people throw a stone into the hole at the center, beautiful light leaked from the inside and the guitar sounds came.

Figure 9: Motozono's Installation

7. MAF2002

In August 2002, we organized the second SUAC MAF 2002[4]. MAF2002 had the following contents : (1) 2 Live Concerts with 11 composers, (2) DSP Summer School with IAMAS (inviting Kit Clayton : programmer of "jitter" in Cycling'74), (3) Installation Gallery exhibiting 12 works, (4) Movie theater that with 15 works, (5) CG Gallery exhibiting 10 works, (6) Europe movie theater. Here, I introduce SUAC students' four projects as follows.

7.1 Ketsu-Puttin (August 2002)

This work was created by 2 students. This work was like two big eggs (Fig.10). People lifted or knocked down this egg, sensors detected the action and the egg vibrated tremblingly by internal motor.

Figure 10: "Ketsu-Puttin"

7.2 Sand Clock (August 2002)

This work was created by 4 students. This work was a huge hourglass (Fig.11). There were 4000 styrene foam balls, it fell by 1 piece per 15 seconds, and the image was projected to this. Another image was projected also to the styrene foam balls that collected on the floor and were piled up. When people upset a small hourglass on the table, the scene was changed and displayed.

Figure 11: "Sand Clock"

7.3 Kana Koubou (August 2002)

This work was created by 4 students. This work was a kind of game. People was displayed on the screen just in front of him/her with CCD camera (Fig.12). Many musical notes fell at random in the screen, too. People moved the frying-pan right and left not to drop the falling note but to catch it. When the frying-pan was upset, caught and collected notes generated the sounds like playback-music.

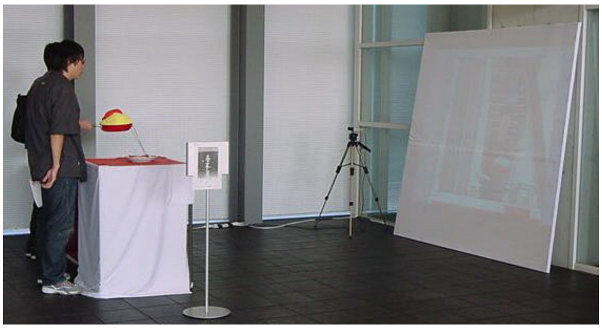


Figure 12: "Kana Koubou"

7.4 Kirameki (August 2002)

This work was created by 2 students. People peeped into the hole of this big box (Fig.13). The ultrasonic sensor detected the peeping action and the system gave the flash one after another by 16 flash-tubes. The shadowgraph was projected by the light of the flash, and it became animation by moving 16 sources of light. There were three kinds of sheets of the shadowgraph. It rotated by 120 degrees, and it changed into new animation by the motor after emitting light.

Figure 13: "Kirameki"

7.5 Chess de Pon (August 2002)

This work was created by 2 students. This work was a kind of game (Fig.14). On the table, there were 8*8=64 points. People could put the objects made of the glass on wherever he/she like. The sounds were generated with each place.

Figure 14: "Chess de Pon"

8. Hakoro (November 2003)

This work was created by 3 students. This work was exhibited in the Intercollege hosted by SUAC in 2003 (Fig.15). This work was a room like the cube of 3m * 3m * 3m, and surroundings were screens of the cloth. There were hundreds of small boxes in the room, and a Japanese character was drawn respectively. If people lifted one small box, then new sounds were generated and four projected images changed on the screen surrounding the room.

Figure 14: "Hakoro"

9. CONCLUSIONS

I have introduced some of the production cases with interactive media-installations by SUAC students' projects from the aspect "experiences with novel interface in education and entertainment" and "reports on students' projects in the framework of NIME related courses". There are still a lot of cases that were not able to be introduced in the limit of space. I plan to introduce it with abundant videos in presentation.

10. REFERENCES

[1] SUAC http://www.suac.ac.jp/index-e.html
[2] NIME04 http://suac.net/NIME/
[3] MAF2001 http://1106.suac.net/SS2001/MAF2001.html
[4] MAF2002 http://1106.suac.net/MAF2002/MAF2002.html

An Acousmatic Composition Environment

Morten Breinbjerg
Multimedia studies,
University of Aarhus,
Denmark.
mbrein@daimi.au.dk

Ole Caprani
Computer Science,
University of Aarhus,
Denmark.
ocaprani@daimi.au.dk

Rasmus Lunding
Information studies,
University of Aarhus,
Denmark.
rasl@daimi.au.dk

Line Kramhøft
Designer, Aarhus,
Denmark
linekramhoeft@hotmail.com

ABSTRACT

In this paper we describe the intentions, the design and functionality of an Acousmatic Composition Environment that allows children or musical novices to educate their auditory curiosity by recording, manipulating and mixing sounds of everyday life. The environment consists of three stands: A stand for sound recording with a soundproof box that ensure good recording facilities in a noisy environment; a stand for sound manipulation with five simple, tangible interfaces; a stand for sound mixing with a graphical computer interface presented on two touch screens.

Keywords

Acousmatic listening, aesthetics, tangible interfaces.

1. INTRODUCTION

Since February 2005, an Acousmatic Composition Environment has been part of the permanent exhibition of the Danish science center "Experimentarium" located nearby Copenhagen, [1]. The Acousmatic Composition Environment allows for the exploration of everyday sounds and for the creation of musical objects and structures by the use of recorded and manipulated sounds. In the design of the environment, we have been greatly inspired by the French composer and theoretician Pierre Schaeffer. Especially his concept of "Acousmatique", his accentuation of audibility, and his compositional approach of sound recording and mixing have been influential to us.

In everyday life sound is indexical to the source and the environment of which it belongs, [2]. Listening to the sound of everyday life as a musical object one need to disregard the source of the sound and focus on its inner qualities. Pierre Schaeffer used the concept of "Acousmatique" to designate this last way of listening and called for a reduced listening process (l'ecoute reduite) in which the sound object was isolated, studied and experienced out of context for the sake of its own timbre and shape, [3]. He achieved this by means of sound recording and techniques of looping and tape speed variation among others. Thus the loop became not only a technical term for the repeating of sound, but his fundament for a new way of listening.

In our project we have created an environment in which a reduced listening process can be performed and in which children/novices are able to manipulate and compose with sounds of everyday life. That is why we have chosen the name: "Acousmatic Composition Environment".

2. THE COMPOSITION ENVIRONMENT

The Acousmatic Composition Environment consists of three stands: a stand for sound recording; a stand for sound manipulation and a stand for sound mixing, Figure 1. Each stand enables sound experiments and exploration and in every stand the sound is played back on stereo loudspeakers integrated into the physical design of the stands.

Figure 1: An Acousmatic Composition Environment at the Danish science center "Experimentarium".

The sounds available in each stand are either fixed, pre-recorded sounds or sounds created and recorded by a visitor. The pre-recorded sounds at each stand secure that visitors can start manipulating and/or mixing sounds without first having to record sounds of their own. This is important in the context of the science center where visitors can freely enter the stands in their own order. The visitor's sounds are saved to and retrieved from a sound database. From all of the three stands the visitor can access his/her sounds in the database by means of a unique barcode for each visitor. This enables sounds recorded in the recording stand to be manipulated in the manipulation stand and later mixed into a visitor created sound composition in the mixing stand. Access to the sounds and sound compositions of each visitor is also available as a net service. A visitor can download his/her sound composition as an mp3 sound and e.g. use it as a ring tone in a mobile phone.

2.1 The Sound Recording Stand

The stand for sound recording includes a Sound Box, Figure 2. The Sound Box is made of Plexiglas and designed as a standard laboratory box with a microphone inside. The purpose of the box is to isolate the sound as it is produced by the visitor. Placed inside the box is a variety of small everyday objects made of solid materials like metal, wood, plastic, glass, etc. The sounds that can be produced by the use of these sound objects covers a wide range of different sound types in both time and frequency domain e.g. short and long attacks, harmonic and inharmonic spectra.

The signal from the microphone inside the box is connected to a computer so that the sounds produced inside the box can be listened to immediately through the speakers inside the stand. A simple graphical interface on a touch screen placed within the box specifies fundamental recording functionalities like record, stop, play, store and save. When played, a recorded sound is automatically looped as a first step towards a reduced listening process, e.g. rhythmic patterns emerge, especially when short sounds are looped.

Figure 2: The Sound Box

2.2 The Sound Manipulation Stand

The stand for sound manipulation holds five *Sound Manipulators* that interface to sound transformation algorithms and control transformations of a recorded digital sound in real time, Figure 3. Sound Manipulators are physical palpable objects with different kinds of handles. Each manipulator has only one handle that can be controlled by using one or two hands. Every handle include sensors connected to a Teleo module, [4]. The sensor data that captures the gesture from one handle is send to a Max/MSP patch [5], as input to one sound transformation algorithm where it controls one or more sound parameters.

Figure 3: The five Manipulators. From top left to right bottom they are: a wheel, a ball, a steer, a gear lever and a roll.

The five manipulators can be operated at the same time and the sound transformation algorithms work on the sound source in parallel. In this way the transformation algorithms work independently on the recorded sound. Consequently you can produce one or more sounds that are perceived e.g. as low frequency background sounds together with high frequency melodic structures when a single sound source is manipulated.

2.2.1 The Sound Manipulators

The Sound Manipulators are small physical interfaces that vary in shape, material, and functionality and each allows a specific sound manipulation. Each manipulator can be used on already recorded sounds that are looped while the manipulation takes place.

In accordance with the context (the science center) each manipulator is designed to be robust and simple to use. Furthermore, the output of the sound transformation is one dimensional in the sense that the visitors will experience continuous sound transformations in both the time and frequency domain i.e. from dark to bright, noisy to harmonic and fast to slowly evolving sounds. Every manipulator has a neutral position in which no transformation takes place; hence the original, recorded everyday sound is heard. Only by continuously pushing, pulling, squeezing or turning the handles, the sound transformation will be applied. If the visitor loosens his/her grip on the handle it moves back to its neutral position like a pitch bender on a keyboard.

Five different manipulators have been developed for the Acousmatic Composition Environment: The wheel, the ball, the steer, the gear lever, and the roll. Below we will describe three of them in greater detail.

2.2.2 The Wheel

The wheel is like a turntable. It is made of steel with small sticks sticking out from the side at regular intervals, Figure 3. This makes it easier to get a grip on the wheel. As the name indicates the wheel can be turned one way or the other and can be brought to spin. The speed of the wheel is measured by the use of a potentiometer and the speed is used as input for a Max/MSP patch. When the wheel is not turned it slowly decelerates. When it stops the sound is not manipulated. The sound transformation uses granular synthesis techniques to cut up the sound in grains. The speed of the wheel controls the grain lengths, how often the grains are activated and the pitch of the grains. The more speed the smaller the grains, the more often they are played, the more variation in pitch.

2.2.3 The Ball

The Ball is a hemisphere made of soft rubber and filled with air, Figure 3. The ball can be squeezed and the more pressure the greater impact it has on the sound being manipulated. The amount of air squeezed out of the ball is measured in a pressure gauge and the amount is sent to a Max/MSP patch. When the visitor lets go of the ball, the ball fills with air and returns to its initial state. The sound transformation uses a three band low pass filter that cuts off the high frequencies. The cut off frequencies move from 18 kHz to 75 Hz in accordance with the amount of pressure that is put on the ball when squeezed. Also an FFT is applied to the sound to approximate the fundamental frequency. This is used to modulate the filter cut-off points.

2.2.4 The Gear Lever

The gear lever can be pulled from the top end towards the visitor, Figure 3. When placed in the top end no sound transformation takes place. The more the gear lever is pulled the greater impact it will have on the sound. The handle of the gear lever connects to a mechanical pump. The resistance in the pump builds up the more the handle is pulled. As a consequence the visitor has to use more power the more the gear lever is pulled downward. The position of the handle is measured by a potentiometer and the position data is send through a Teleo module to a patch in Max/MSP that performs the sound transformation. The sound transformation uses the “bong~” and “paf~” objects developed by Miller Puckette, [6], [7]. It keeps the rhythmic and dynamic attributes of the original sound, but gives the timbre a more and more synthetic like character as the handle is pulled downward.

2.3 The Sound Mixing Stand

The stand for sound mixing consists of two touch screens with a graphical interface designed as a ten track sequencer. Each track loop’s a manipulated sound. The sound can be muted, pitched in a pentatonic scale and its volume controlled by a staircase envelope with a time resolution of 250 msec. A number of preset buttons on the screen can be used to select different staircase envelopes for all the tracks. This gives immediately access to different ways of mixing the sounds.

3. DISCUSSION

We believe the most important aspects of our work are: 1) The construction of an environment that allows children/novices to work with everyday sounds as musical material, 2) An environment that allows for recording, manipulation and mixing of sound objects. As a result a child/novice can perform the basic steps in the creation of a sound composition and e.g. use it as a unique ring tone.

The “Acousmatic Composition Environment” deliberately and continually pursues an aesthetic of music that connects to the acousmatic music of Pierre Schaeffer and to sound art experiments, [8]. By this we wish to underline the importance of listening and to strengthen the ability of children/novices to listen in a qualified manner. The use of everyday sounds as opposed to e.g. instrumental sounds, confronts children/novices with a musical material that they are normally not ask to listen to. As such we emphasize computer technology as an extension of audibility and regard the Sound Manipulators and the Sound Box as media technologies that allow children/novices to hear the world differently and to immerse into a world of sounds that we expect is not normally listened to.

3.1 The Sound Box

The Sound Box is a fairly simple, but yet very efficient way to record sound in public space. Normally sound recording in public space suffers form the noise of the surroundings, but in the Sound Box, which is coated on the inside, the sound producing event is isolated. When the prototype of the Sound Box was tested,[8], it was obvious that children/novices engaged in listening to the sounds that everyday objects produced.

3.2 The Sound Manipulators

As David Wessel and Matthew Wright suggest, [9], the major advantage of computer based instruments is the possibility of “immense timbral freedom”. They believe it should be relative easy to start playing a computer based instrument but points to the fact that making an instrument easy to play often is contradictory to the sounding complexity of that same instrument. As a consequence a “simple-to-use” computer based instrument quickly gets a “toy-like” character. The instrument – the audio output - is not complex enough to encourage a continuing exploration. We find this argument to be a strong one and most relevant in our context.

Our response to this argument is twofold. First we try to secure the ease-of-use by making the physical interfaces quite simple and well-known although not necessarily in a musical context. In the design of the five manipulators we have (as so many before us) been greatly inspired by Gibson’s term affordance, [10], which has been introduced into the interaction design community by Donald Norman, [11]. It basically states that form giving should invite effective action since affordance concerns the relation between appearance and action [12]. We believe the shape of each manipulator expresses the kind of action necessary to operate it. We find this to be crucial in the present context and therefore we have chosen to use well known interfaces like a wheel, a steer, a gear lever, a ball, and a roll. As such the familiarity of the interfaces chosen is meant to ease the understanding of performance: A wheel is for spinning, a gear lever for pulling and a soft ball for squeezing. Hence a visitor can easily figure out what to do with each of the five manipulators. In other words, the way to operate the Sound Manipulators is not very advanced or exotic, but it opens a vast space of aesthetic experience with the sound objects produced and recorded in the Sound Box. Furthermore, the sound transformations react immediately to even the smallest change of the interface. Not in the sense that the sound changes suddenly in all its parameters by the least touch of the manipulator, but in the sense that even a small gesture is audible, letting you feel immediate control over the sound transformation. To strengthen the feel of control we have designed the interfaces to demand continues physical input. Otherwise the manipulator will settle at its initial position where no transformation is taking place. By continuously having to use the power of his/her muscles to control the interface and thereby the sound transformation, the visitor constantly receives audible as well as physical feedback and most important of all a feedback of the correspondence between the sound and gesture performed. As such our interfaces match the interfaces developed by Dominic Robson in the sense that they ”all incorporate a physical change in the sound along the physical continuum”, [13]. This we believe is very important; mapping the gesture to the audible result lets child/novice feel that he/she is in charge of the audible sound.

Second, we have provided algorithms that can produce rich timbral variations. Despite the simple one handle interfaces the complexity is secured by letting the input control several parameters in the sound transformation. Also, the variability of the sounds produced is not only dependent on the complexity of the sound transformations, but also on the recorded sound which varies according to the way the visitor play with the small everyday objects within the Sound Box.

3.3 The Sound Mixer

The Sound Mixer does not reflect our initial proposal. We proposed to make the sequencer as a table with holes, into which small containers holding the manipulated sounds (by the use of RFID tags) could be plugged. Because of the context of the Science Center the idea were refused, since the containers can be moved away or even stolen. As the Sound Mixer Stand in the current setup does not reflect our wishes, we choose not to comment on the design further in this paper.

3.4 Related work

Our use and design of tangible interfaces clearly points in the direction of previous research carried out by Gil Weinberg and

Seum-Lim Gan, [14]. Like their project on”The Squeezables” our project stresses the physical design of the interface as an important way to encourage new ways of interaction. Other projects like Blok Jam, [15], or Agroove ,[16], have inspired us in some aspects of our work, e.g. in using tangible interfaces to interact with prerecorded sound segments. Like Blok Jam our original idea of the mixer board uses physical objects as “sound containers” and like Agroove our system allows for live manipulation and mixing of musical structures.

However, the aesthetic approach of the “Acousmatic Composition Environment” is fundamentally different. Pursuing the aesthetics of making music by recording, manipulating and mixing sounds of everyday life, we encourage children/novices to reveal the musical quality of the sound object it self. As such our goal is to cross the boundary between everyday listening and music listening. Instead of making music with MIDI notes mapping the control information of the Sound Manipulators to the MIDI standard as in Squeezables and thereby remaining within the pitch paradigm of music, we offer children/novices the opportunity to seek the music in everyday sounds.

4. FUTURE WORK

As mentioned in the beginning of this paper The Acousmatic Listening Environment is part of the permanent exhibition of the Danish Science Center “Experimentarium”. In the future we will perform a thorough investigation on how the children/novices use the environment. We had hoped to be able to monitor the installation by video in order to gather the information necessary to validate the human-computer interaction, but we have not yet received the permission to do so, since the science center is a public space. One further development that we would like to develop is a physical mixing table. As mentioned earlier we are not entirely satisfied with the touch screen solution. We believe a physical sequencer and physical objects as containers for recorded and manipulated sounds will be in the line of the overall concept in a much more satisfying manner.

5. CONCLUSION

The goal of our project is to design and develop an Acousmatic Composition Environment in which children/novices can perform a reduced listening process, record, manipulate and mix sounds of everyday life. We are happy to see the environment in a context like a science center and hope in the long run to get valuable experience of the actually strength of the idea and the robustness of the system. Hopefully, the children/novices will get a playful exploration of digital sound and an emotional experience of fun, involvement and beauty. We encourage them to listen in a qualified manner.

6. ACKNOWLEDGMENT

Thanks to Jesper Rasted, Christina Fromberg and Niels Hornstrup from the Danish science center “Experimentarium” and ISIS Katrinebjerg for financial support.

7. REFERENCES

[1] The website of the Experimentarium can be reached at http://www.experimentarium.dk/

[2] Truax, Barry Acoustic Communication 2. Edition, ABLEX Publishing, Westport, Connecticut, 2001.

[3] Schaeffer, Pierre Traité des objets Musicaux. Paris, Edition du Seuil 1966.

[4] Making things, http://www.makingthings.com

[5] Cycling74, http://www.cycling74.com

[6] Puckette, M.,Apel, T., Zicarelli, D., Real-time audio analysis tools for Pd and MSP, Proceedings, ICMC 98

[7] Puckette, M., Formant-based audio synthesis using nonlinear distortion, Audio Engineering Society. JAES 43/1, pp. 40-47,1996

[8] Breinbjerg, M., Caprani, O., Lunding, R., Touch the Sound - the concept of the "object Sonore" as a design metaphor. Proceedings of the International Workshop of "Interaction Design and Children", Eindhoven, 2002.

[9] Wessel, D., Wright, M., Problems and Prospects for Intimate Musical Control of Computers, Computer Music Journal, 26:3, 2002, pp. 11-22.

[10] Gibson, James J. The Ecological Approach to Visual Perception, Lawrence Erlbaum Associates, Publishers, London, 1986

[11] Norman, Donald A. The design of every day things, MIT press, London, 1998

[12] Djajadiningrat T.,Wensveen S., Kees Overbeeke J. F., Tangible products: redressing the balance between appearance and action. Springer-verlag London Limited 2004.

[13] Robson, Dominic PLAY!: Sound Toys for Non-Musicians, Computer Music Journal, 26:3, 2002, pp. 50-61.

[14] Weinberg, Gil and Gan, Seum-Lim. “The Squeezables: Toward an Expressive and Interdependent Multi-player Musical Instrument”. Computer Music Journal, MIT Press 2001

[15] Newton-Dunn, H., Nakano, H., Gibson, J., Block Jam: A Tangible Interface for Interactive Music, Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada

[16] Poupyrev, I. “Augmented Groove: Collaborative Jamming in Augmented Reality” in SIGGRAPH 2000 Conference Abstracts and Applications, ACM Press, NY, p.77

Bioinformatic Feedback: performer bio-data as a driver for real-time composition

Robert Hamilton

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
rob@ccrma.stanford.edu

ABSTRACT
This paper describes a software system using bioinformatic data recorded from a performer in real-time as a probabilistic driver for the composition and subsequent real-time generation of traditionally notated musical scores. To facilitate the generation and presentation of musical scores to a performer, the system makes use of a custom LilyPond output parser, a set of Java classes running within Cycling '74's MAX environment for data analysis and score generation, and an Atmel AT-Mega16 micro-processor capable of converting analog bioinformatic sensor data into Open Sound Control (OSC) messages.

Keywords
Bioinformatics, composition, real-time score generation.

1. INTRODUCTION
The mapping of fluctuations of a performer's physiological state during the performance of a piece of music to specific compositional parameters can be used to form an intimate relationship between a performer and the structure of a piece of music. Human physiological responses, voluntarily or involuntarily generated and measured with physiological sensors, can be mapped in software to compositional parameters. With the use of a real-time score generating and display system, physiological response re-interpreted as notated musical gesture can be displayed to the performer for interpretation.

Continuing performance of this newly generated musical notation data can result in a bioinformatic feedback loop, within which the performer's physiological state reacts to a musical abstraction of the most recent physiological state. The mapping of predominately involuntary performer excitation levels to pre-composed musical events creates a hybrid improvisational and compositional form allowing both the composer and performer to have input into the final compositional structure.

Rather than use voluntarily generated physiological signals as active controls on the musical output, this system seeks instead to modify compositional content to react to involuntary physiological reaction. In this model, autonomic physiological data acts as a control signal while a performer's physical gesture retains its traditional role as an expressive component of performance. In essence, the compositional decisions made by the composer act as a deterministic filter for the autonomic control signals generated by the performer.

By varying the relationship between physiological reaction and resultant compositional output, different compositional forms can be created. When an inverse response mapping is applied, where strong sensor readings generate weak or relatively simple compositional structures, a performer's physiological state can be coerced into a less excited state. Similarly, a performer in a stable or less excited state will be presented with more active musical cells, aiming to excite the performer into a more active state. Conversely, when a direct mapping between physiological state and compositional form is applied, musical output mirrors physiological state, out-putting musical cells that reinforce the current state.

2. SYSTEM DESIGN
To provide for the collection and processing of incoming data streams as well as for the output of notated musical data, the integration of a number of existing software platforms with was necessary. By standardizing data formats and making use of OSC [11] for data transmission, existing open-source software such as LilyPond [7], Pure Data (PD) [9], and GhostView [10] could be utilized alongside commercial software such as Max/MSP for data processing, analysis and display (see Figure 1).

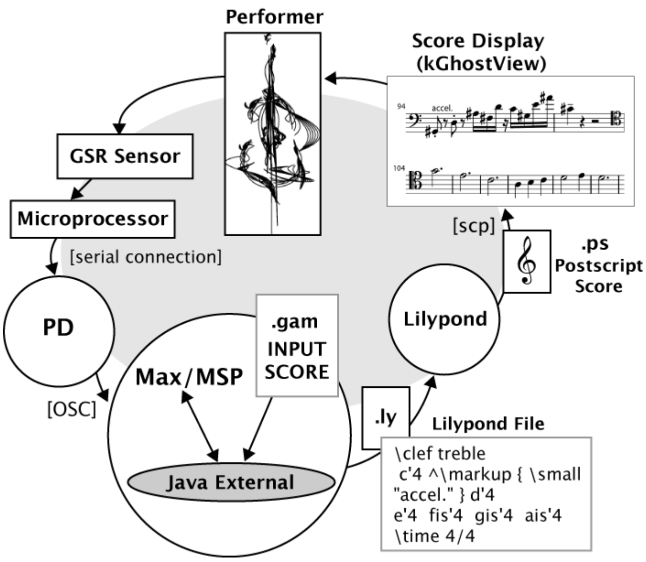


Figure 1. System Data-Flow

2.1 Galvanic skin response (GSR)

For the purpose of system proof-of-concept and initial testing, a simple Galvanic skin response (GSR) sensor was used to measure variances in skin conductance during a musical performance. Galvanic Skin Response can be described as a measured fluctuation in the electrical resistance of the skin. By measuring changes in skin conductivity relative to applied stimuli, it has been proposed that not only can a subject's emotional or attentional reaction be measured but that the GSR can be considered relatively autonomic and not easily controlled by the subject [2]. While a number of pre-recorded data streams of varying physiological data sources have been tested with the system (EKG, EEG), the relative simplicity of implementation of the GSR sensors in a real-time environment led to their use in initial testing and performance situations.

2.2 Hardware

GSR data is extracted from a performer with the use of a custom analog GSR circuit connected to an ATMEL AT-MEGA16 microprocessor over a serial connection. The microprocessor is running custom C-code capable of outputting voltage readings in the OSC data protocol as a scaled stream of values appropriate for analysis. The GSR circuit used in the initial testing and development of the system was designed and built by Jay Kadis of Stanford University's CCRMA (see Figure 2).

While a standard methodology for the measurement of GSR data calls for the attaching of conductive sensors to the fingers of a subject (to take advantage of the greater amount of resistance fluctuation in finger tissue), as musical instruments tend to be performed using the fingers and hands, to reduce data artifacts due to physical displacement of finger mounted GSR sensors during performance, a pair of sensors were instead attached to the performer's toes. Non-performance tests of GSR fluctuations comparing toe and finger placements showed similar results for either location.

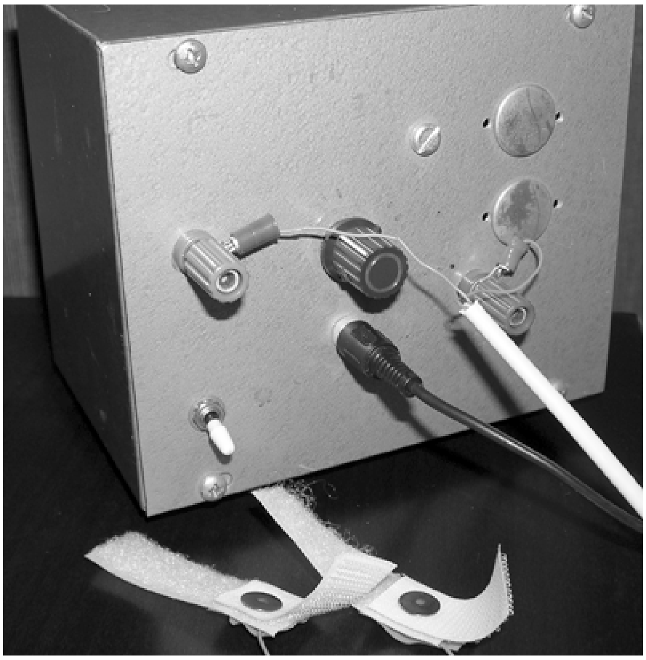


Figure 2. GSR circuit box and finger/toe sensors

Figure 3. Max/MSP GUI

2.3 Software workflow

OSC formatting and routing objects[1] running in a PD patch on a computer with a serial connection provide a steady data stream of resistance values converted from the analog realm to the digital realm by the microprocessor's ADC. In this case, the PD patch simply makes use of objects capable of reading serial data to act as a signal router, sending raw digital resistance values over a local OSC connection to a Max/MSP patch where data processing and analysis take place.

The core of the system lies in a set of Java classes designed to take pre-composed musical cells as input, define probabilistic relationships between each cell's pitch "activity" (defined as an aggregate of pitch-steps through a cell) and to output musical cells in the Lilypond (.ly). These Java classes are instantiated within the Max/MSP environment, allowing for real-time interaction between the data streams and the classes, as well as a fully-featured system GUI for real-time control and data representation (see Figure 3), without a prohibitive development timeframe.

By leveraging Max/MSP's ability to interact with the BSD Unix shell of an Apple computer running OS X (using the "shell" object), programmatic shell calls are made to both LilyPond and to SCP for data-transport from the processing machine to a locally-networked display terminal running a postscript viewing application such as GhostView. This modular approach for processing and data display creates an extremely flexible workflow which can be adapted to run on a number of system platforms and software applications.

2.4 Data processing

Signal levels taken from two GSR sensors attached to a performer's fingers or toes are recorded into sample buffers, creating a windowed data set representing a fluctuation of input signal over a given time frame. Both the frequency of sampling and the number of samples comprising a window are configurable using the Max/MSP GUI. Each windowed data set is first compared against a baseline data set – taken before the start of the performance with the performer in a relatively stable physiological state – and subsequently used to generate a single scaled "activity" value, representing the variation in amplitude of each input sample in relation to the amplitude of its previously recorded sample.

---

[1] *OSC, OSCroute,* and *dumpOSCSerial* objects by Matt Wright et al.

Pre-composed musical data cells comprised of musical note and articulation data called "Gamut Squares" are loaded into memory and used to create a detailed hierarchical musical data structure in Java. In a manner similar to the aforementioned signal "activity" metric, the fluctuation of pitch in a given Gamut Square is used to calculate a musical activity cell. A more detailed description of cell-based compositional techniques and the particular data formatting for cell input is available in [3].

After establishing baseline values for input data, it is useful to establish relative maximum and minimum data values for computation. The subsequent range of possible or probable data values can be subsected into any number of "activity zones" by setting a "zone" value in the GUI. This effectively creates n-number of equally sized ranges of activity for both the musical Gamut Squares as well as for the input GSR data sets. In this manner, increases or decreases in precision can be set to account for more or less active data streams.

2.5 Pitch/rhythmic cell selection

Selection of musical data cells for output occurs by correlating GSR activity readings with musical activity values from respective zones. Cells from the desired activity zone are given a GUI-defined high probability of selection from the overall set of musical cells. Cells from other activity zones are given a correspondingly low probability of selection. Cells are then selected from this macro set of probability-scaled data cells and set into a structure for subsequent output. By selecting musical cells using probabilities rather than by directly selecting cells based on their activity levels the level to which a performer's bioinformatic data can shape the composition is left inexact. In this manner, the composition can always embark on unexpected directions irregardless of its relationship to the performer.

2.6 Dynamics selection

While the current implementation of the system uses physiological data primarily to drive the selection of note cells, other compositional aspects such as dynamic and articulation can be mapped to data sources and selected probabilistically. In tests using pre-recorded or modeled data-streams (EEG, EKG), as well as tests using the live GSR stream, the windowed activity reading was mapped to dynamic selection on a note-by-note basis. Mappings are currently applied directly, where a greater activity reading leads to an increase in probabilistic weighting for dynamic values in corresponding activity zones. In this model, a louder dynamic, such as ***ff*** is regarded as having a greater activity than a softer dynamic, ***pp***.

2.7 Data formatting and output

When a user-defined threshold of beats of music has been generated, the selected musical cells are converted into the LilyPond musical score data format using a custom-written LilyPond parser and output to a text-file. Using Unix shell calls invoked from Max/MSP, this .ly file is then processed by LilyPond into a standard .ps postscript file and moved to a directory being "watched" by a GhostView postscript viewer application such as kGhostView. Any change to the file's modification date results in a refresh of the kGhostView application, refreshing the viewer. The viewer is being presented on a computer monitor to the performer who is then able to perform the recently generated musical phrase. In recent performances with the system, it has been useful to generate two postscript output files for alternate sets of output data, and to use a vertically-aligned pair of kGhostView display windows to alternately update sections of the composition. In this manner, one window of display information can be updated while the performer is still playing the previously rendered and displayed window.

3. PERFORMANCE PRACTICE

As an initial test of the system, a series of performances of probabilistically generated cell-based musical compositions driven by fluctuations in a performer's GSR were given in the Fall of 2005 at Stanford University's Center for Computer Research in Music and Acoustics (CCRMA). Cellist Colin Oldham, a visiting scholar at CCRMA, performed a suite of compositions where short pre-composed phrases of music of varying complexity and pitch variance were dynamically selected and presented for performance based on the real-time windowed analysis of his fluctuating GSR levels. During these initial performances it became clear that while the basic nature of real-time composition necessitated a performer with excellent sight-reading abilities, the pre-composed nature of this cell-based compositional approach allowed the performer to study and practice the source material before performance, greatly reducing performance error due to surprise. Additionally, by viewing score data in two separately-refreshed windows, the performer was able to read ahead while performing less-challenging materials, again reducing possible performance error.

4. RELATED WORK

While there exist numerous projects designed to generate musical construct from bioinformatic response, the majority seem to focus on not only conscious or active control by performers/subjects but also on the application of relatively direct mappings of bio-data to musical form. In this approach, control systems allow performers to use voluntary physiological gesture as a direct controller for musical gesture, turning the body into a sophisticated musical control interface [4][5]. Even when systems incorporate physiological biofeedback signals, many do so to create direct and controllable mappings between performer and performance.

Work by Dr. Geoffrey Wright and NeuroSonics on Brain Generated Music (BGM) addresses the use of EEG data as a musical driver to create more abstract representations of physiological data [6]. Indeed, such an approach makes use of a confluence of voluntary and involuntary bioinformatic data, as well as the generation of bioinformatic feedback during a "performance", as subjects listen to music generated by their brain waves in real-time.

In the paradigm of real-time score generation and presentation systems, Kevin Baird's *No Clergy* project [1] addresses many of the same generation and display issues faced here using a network server and web-browser for score display and a Ruby/Python backend.

Additionally, initial development of the Java probabilistic composition classes used in this project began as the core to the *jChing* compositional system [3], itself designed as a software model of John Cage's influential *I-Ching* compositional techniques [8].

5. CONCLUSIONS

From early testing and performances it is clear that while the concept of physiological data as a compositional driver seems viable, great care must be given in choosing bioinformatic sensors so that fluctuations in body state are consistent and to an extent predictable within a given range. While the data generated by the GSR sensor shows evolution and gradual change over longer time periods (in the n-seconds range), GSR tracking failed to show adequate fluctuation following short-term musical events (in the n-milliseconds range) without extreme stimulation. State changes as measured by GSR seem to develop over longer periods of time rather than discretely measurable periods and might be a better match with other compositional parameters, such as part density in a multi-voiced work.

Future directions for the project include development and integration of additional data sensors, such as EKG, EEG or body-temperature sensors, which should provide a more consistently active state across shorter time frames. By combining a variety of sensors, a more accurate measurement of physiological state and its reaction to musical events should be possible. Additional development of small wired or wireless biosensors capable of transmitting data over standard protocols (i.e. USB, Bluetooth, wireless LAN) is currently under investigation.

To better determine the extent to which compositional cells can affect performers of various instruments, additional testing covering a range of instrumental performers and compositional excerpts is being planned. In doing so, more appropriately reactive mappings between various compositional constructs and the Bioinformatic Feedback system should become clear.

6. ACKNOWLEDGMENTS

Many thanks to Jay Kadis for the generous donation of his GSR signal processing circuit. Thanks also to Jonathan Berger, Bill Verplank, Max Mathews, Colin Oldham and Carr Wilkerson for their suggestions and support.

7. REFERENCES

[1] Baird, K. No Clergy:Real-Time Generation and Modification of Music Notation. In *Proceedings of the 2005 SPARK Festival*, (Minneapolis, USA, 2005).

[2] Greenfield, N.S. and Steinback, R.A. *Handbook of psychophysiology*. New York: Holt, Rinehart & Winston, 1972.

[3] Hamilton, R. The jChing: an Algorithmic Java-Based Compositional System. In *Proceedings of the 2005 International Computer Music Conference (ICMC 2005)* (Barcelona, Spain, September 5-9, 2005).

[4] Knapp, R.B. and Cook, P.R. The Integral Music Controller: Introducing A Direct Emotional Interface To Gestural Control of Sound Synthesis. *Proceedings of the 2005 International Computer Music Conference (ICMC 2005)* (Barcelona, Spain, September 5-9, 2005).

[5] Knapp, R.B. and Lusted, H.S. A Bioelectric Controller for Computer Music Application*s*, *Computer Music Journal*, 14 (1), 1990, pp. 42-47.

[6] Kurzweil, R. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*, Viking Penguin Press, New York, 1999.

[7] Nienhuys, H. and Nieuwenhuizen, J. LilyPond, a system for automated music engraving. *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, Firenze, Italy, May 2003.

[8] Pritchett, J. *The Music of John Cage* (New York: Cambridge University Press, 1996)

[9] Puckette, M. Pure Data. *Proceedings of the International Computer Music Conference, (ICMC 1996)* (San Francisco, California, 1996) pages 269-272.

[10] Thiesen, T., GhostView, http://www.cs.wisc.edu/~ghost/

[11] Wright M. and Freed, A., Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the International Computer Music Conference (ICMC 1997)*, (Thessaloniki, Greece, 1997).

The Light Matrix: An Interface for musical expression and performance

Jonathan Pak
Electronic musician/Engineer
jon@pak.id.au

ABSTRACT
A prototype of the Light Matrix interface is presented here. At the core of this device is a rectangular array of Light Emitting Diodes (LED). Although designed to emit light, it is also possible for LEDs to function as crude photosensors. The Light Matrix exploits the bidirectional properties of the LED to create a light reflection based proximity sensor. The performer interacts with the device through hand movements in front of the LED sensor matrix. The potential of this device as an interface for real-time musical expression for computer based audio systems will be discussed along with its suitability as an instrument for live performance. Here the aspects of performativity covered include performer-instrument interaction and audience perception of the relationship between physical interaction and resulting sound.

Keywords
Music controller, LED photosensors, performativity, real-time expression, mapping visualisation, programmable response.

1. INTRODUCTION
The prototype of the Light Matrix is a computer interface intended for electronic music performance. The face of the device consists of a grid of LEDs functioning as both a proximity sensing array and monochrome display. Figure 1 illustrates the process involved in translating hand movements into musical expression. The intermediate forms of data are shown at each stage of the process.

To begin with, each LED sensor measures the intensity of reflected light, light that is emitted by other LEDs in the matrix. The reflected light intensity is proportional to the proximity of the performer's hand to the plane of the matrix. The raw data from this device may be represented as a moving height field. This data is relayed to a software application running on a computer which translates features of the height field into control messages. In turn these messages may be used to control audio applications, virtual instruments and effects.

The intensities of each LED pixel are independently adjustable which gives rise to some unique features of this device. The Light Matrix responds to the performer not through tactile feedback seen in traditional instruments [10], but through modulating the light intensity pattern of the LED matrix. This optical feedback is derived from three different sources (see Figure 1). These sources are: the height field data, mappings defined by the software application and the resultant audio signal. This optic response modifies the characteristics of physical interaction and provides visual cues to both performer and audience as to how the instrument is operating.

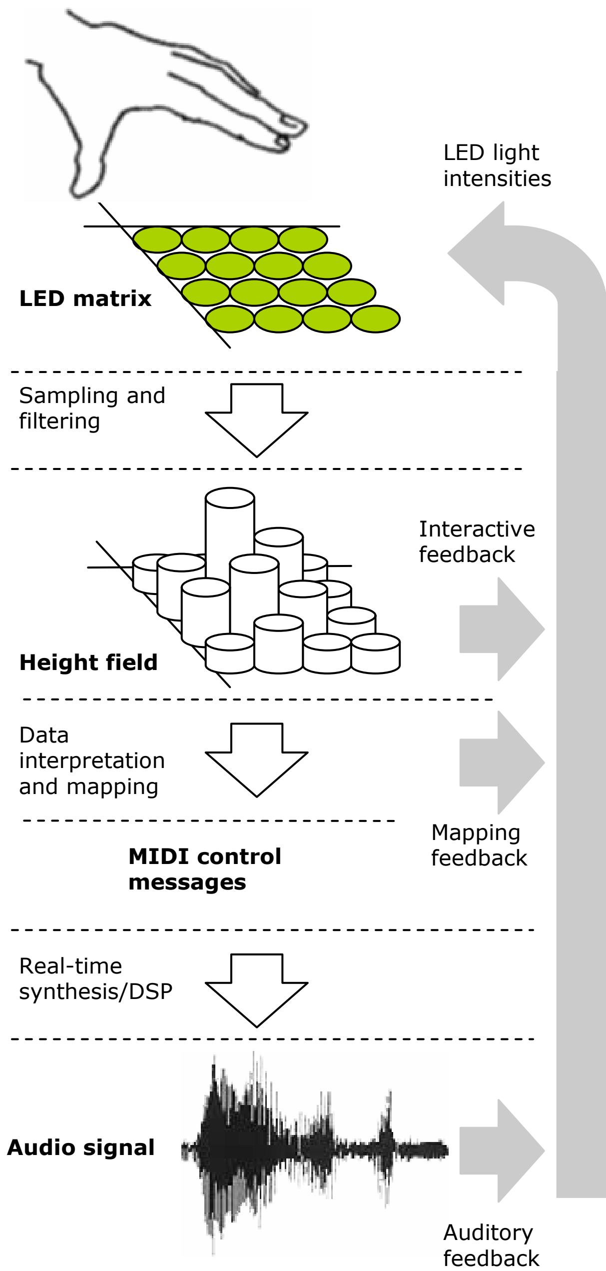


Figure 1. Device concept illustrating multiple stage feedback

2. RELATED WORK

The work of J. Han and associates has also utilised the bidirectional properties of LEDs in their project: Multi-Touch Sensing through LED Matrix Displays [3]. The title of the project alludes to a slightly different focus on multiple contact point detection rather than a quantitative proximity measurement. The display provides no visual feedback and is not specific to musical applications.

The Light Matrix also has parallels with the Tenori-On [4] musical instrument, currently under development by Toshio Iwai. The elements common to both devices are the similar form factor and interactive LED display. One interacts with the Tenori-On by pushing on individual LEDs in the display.

Another comparable device is JazzMutant's musical controller: the Lemur [5]. This features a high resolution pressure sensitive multi-touch display though it lacks the depth sensing capability of the Light Matrix.

The Light Matrix has a tactile counterpart in D. Overholt's MATRIX interface [1]. The limitations of the MATRIX's spring loaded rod assembly are: the substantial force required to push down the rods and the fact that adjacent rods prevent lateral movement of the performer's hand.

3. MUSICAL PERFORMANCE ASPECTS

The advent of the DAW (Digital Audio Workstation) has afforded musicians great scope to explore new sound textures and improvisatory techniques never before possible. Virtual instruments can be much more complicated than their physical counterparts with no upper limit on the number of parameters controlling expression.

Using standard interfaces such as faders, switches, knobs and keyboards, it is difficult to control even a few parameters simultaneously. The consequences of this limited form of control in a live performance are a lack of spontaneity and a possibly unengaging performance.

The Light Matrix is designed to overcome these limitations by providing a greater level of control. The prototype has an 8 × 8 grid of LED sensors each capable of measuring depth with 8 bits of precision. As a controller, the size of the input space makes it possible to realise many complex mapping schemes to translate performative hand gestures into sound.

The low latency of the device coupled with simultaneous control over many parameters makes that the Light Matrix ideal for real-time musical applications in synthesis and DSP (Digital Signal Processing).

The constant low latency of the device means that the interface is both predictable and responsive to the performer's gestures. The degree of simultaneous control gives the artist a broader palette for self expression through music. Both of these factors underscore the immediacy of performance for the audience.

3.1 Physical interaction

The absence of tactile feedback arguably allows a more direct form of expression. Freedom from the physicality of playing a traditional instrument eliminates physical constraints and reduces the effort required to play the instrument. Any effort required to operate the interface translates directly into expressive movement.

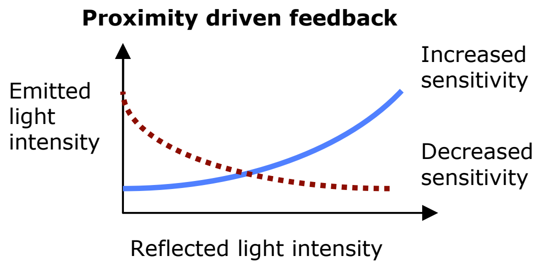
The device is capable of detecting the presence of any object that is reflective but in this case the use of hands as the means to convey expression is only logical. The capacity of the human hand to communicate semantic and emotional content is second only to facial expressions. The inherent expressiveness of human hand gestures presents enormous potential for conveying musical expression.

3.2 Optical feedback

The Light Matrix uses a unique form of feedback through light intensity. A more intense light source will enable the LED sensors to detect hand positions further away by causing a stronger reflection. Conversely a weaker light source will limit the usable range of control.

3.2.1 Performer interaction

By continuously updating the intensities of each LED in response to the motions of the performer's hand, the interactive characteristics of the device can be changed. Figure 2 illustrates some of the basic responses used to alter the parameters of physical interaction.



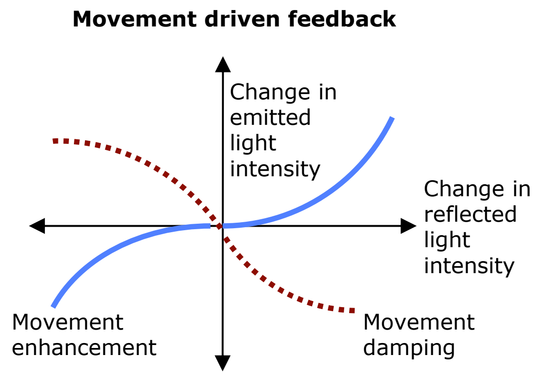


Figure 2. Optical feedback schemes driven by physical interaction.

3.2.2 Functional visualisation of mapping

This technique is used to adjust the interface's responsiveness which is best illustrated with an example:

A practical application of the interface could be to manipulate the timbre of a synthesiser. An illustration of the LED matrix for this example is shown in Figure 3. The volume of low frequency components of the synthesiser's sound is increased by moving closer to the surface on the left side of the matrix. Progressively higher frequencies are controlled by moving towards the right side of the matrix.

Often the volume of high frequencies needs to be tempered in relation to the low tones so as not to sound shrill. The realisation of this mapping should be immediately apparent from the gradient in light intensity. The reduced intensity on the right corresponding to a reduced emphasis on the higher frequencies, making the synthesiser sound less bright. The visual representation of this mapping is not only functional but also informative for the performer and the audience.

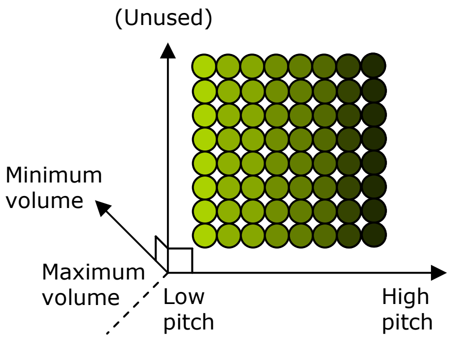


Figure 3. Example of weighted frequency response through visible mapping

3.2.3 Auditory feedback

Auditory feedback is also possible with this system by using sound to control the intensities of the LEDs in the matrix. This can be illustrated by extending the example in section 3.2.2. Figure 4 shows how the sound spectrum of the hypothetical synthesiser may be divided into frequency bands and mapped onto the matrix as variations in light intensity. The result is that the interface reacts more strongly to the most dominant frequencies in the sound spectrum. This type of feedback is a very direct and intuitive form of control as the performer is interacting with some aspect of the sound they are producing.

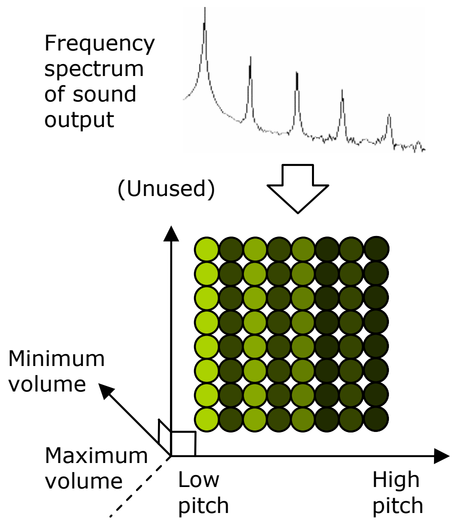


Figure 4. Example of audio-optic feedback

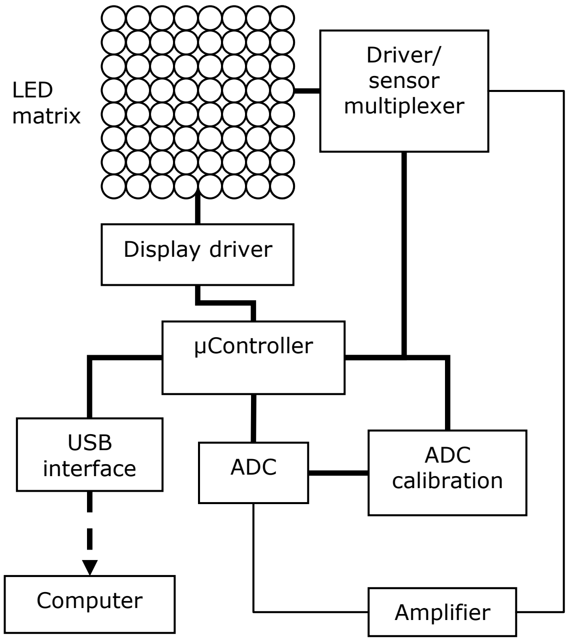


Figure 5. Functional block diagram of hardware

3.3 Audience interaction

The musical purist will always put the sonic outcome foremost in any performance. However a performance not only entails listening but also observing. A recital will always be more engaging if it is interesting to watch. This interest is partly maintained through the audience's understanding of the instrumentalist's craft: how physical interaction with an instrument relates to the sound it produces. It is this cross perceptual reinforcement that enhances the experience of a live musical performance.

When it comes to electronic music performances one of the primary tools is the laptop computer. As a musical instrument the operation of a computer is generally unengaging [11] because there are no motions discernable by the listeners that translate to sound output. The Light Matrix remedies this problem by providing an interface to the computer that removes the physical barrier of the computer screen. Much like a traditional instrument, the gestures and motions of a musician operating the Light Matrix are clearly visible. In addition the interface has a certain aesthetic appeal because the glowing surface of the device highlights the performer's hands.

4. HARDWARE IMPLEMENTATION

The major components of the device prototype are shown in Figure 5. The device communicates with the computer via USB (Universal Serial Bus) protocol. The microcontroller contains inbuilt USB interface circuitry and is also serially programmable over USB.

The height field data is constructed by scanning each LED sequentially in a raster pattern. There are two phases to each LED pixel read operation:

1. A reference measurement: This is performed under "dark" conditions where all the LEDs in the matrix are switched off during the sampling period.
2. The actual measurement: While sampling, all other LEDs in the matrix remain in a light emitting state.

The reference measurement forms the baseline for comparison with the actual measurement. Using the difference between the reference measurement and actual measurement reduces the interference caused by ambient lighting.

The scanning operation is performed at the maximum rate allowable by the ADC (Analogue Digital Converter) chip. Consequently the strobing caused by alternating between these two measurement phases is not perceptible due to the response time of the human visual system. The net effect being that the device is on continuously even though it is actually switched off for some of the time.

4.1 LED matrix

LEDs as sensors in this application offer several advantages. Firstly they are cheaper than photodiodes or phototransistors and the bidirectional property of each element simplifies wiring, mechanical construction and circuit complexity. This in turn reduces cost and allows for increased matrix density and size.

When the LED is forward biased, the current flow causes the device to emit light as per normal. Under certain conditions LEDs can be operated as photosensors [6][7]. As photosensors, LEDs act as a narrow band light sensors, generally picking up wavelengths shorter than those emitted [2]. The prototype utilises high intensity red LEDs which exhibit relatively good sensitivity because the emission spectrum is close to the wavelengths in the absorption spectrum.

4.2 Photosensor amplifier/ADC

The role of the amplifier is to convert the photocurrent from each LED into a voltage, the current being proportional to the light intensity impinging on the diode junction.

The output voltage from the amplifier is then digitised by the ADC which is a 13 bit switched capacitor variety. Calibration of the ADC is done electronically by a pair of DACs linked to the microcontroller. These provide the upper and lower reference voltages needed by the ADC.

4.3 Display driver/multiplexer

The display driver consists of a bank of LED driver chips interfaced with the microcontroller that provide constant current PWM dimming. The multiplexer circuitry is responsible for interleaving the display and sensing phases.

5. SOFTWARE/FIRMWARE

The software is an application written in C# that performs the following tasks:

- Communicating with the Light Matrix device over USB. This includes reading raw height field data from the microcontroller and setting the LED intensity values.
- Mapping height field data to MIDI control messages and sending these messages to a specified MIDI output port.
- Mapping height field data to LED intensity values.
- Frequency and loudness analysis of an audio signal.
- Mapping frequency and loudness data to LED intensity values.

An audio signal is fed into the Light Matrix control software through a purpose built VST [12] effect plug-in. This plug-in acts as a bridge between the host audio application and the Light Matrix control application.

6. FUTURE ENHANCEMENTS

Exploration of the optical feedback and mapping possibilities of the Light Matrix has only just begun. More flexible software is needed to define these mappings.

Use of the antiquated MIDI protocol limits the amount of control available. Open Sound Control (OSC) [8] or a more direct form of communication should be employed.

More investigation is needed to determine an arrangement of LEDs that is more conducive to performance gestures made with hands. The arrangement of the LEDs in a square matrix is probably not the most suitable.

7. REFERENCES

[1] Baker, Bonnie C., *Keeping the Signal Clean in Photosensing Instrumentation*, http://www.sensorsmag.com/articles/0697/photo_s

[2] Dietz, P., & Yerazunis, W., & Leigh, D., *Very Low-Cost Sensing and Communication Using Bidirectional LEDs*

[3] Han, J., *Multi-Touch Sensing through LED Matrix Displays*, http://mrl.nyu.edu/~jhan/ledtouch/

[4] Iwai, T. *Yamaha Tenori-On*, http://www.global.yamaha.com/design/tenori-on/

[5] JazzMutant Lemur, http://www.jazzmutant.com/lemur_overview.php

[6] Mims, Forrest M., III, *Siliconnections: Coming of Age in the Electronic Era*, McGraw-Hill, New York, NY, 1986.

[7] Mims, Forrest M., III, *LED Circuits and Projects*, Howard W. Sams and Co., Inc., New York, NY, pp. 60-61, 76-77, 122-123.

[8] Open Sound Control, http://www.opensoundcontrol.org

[9] Overholt, D. *The MATRIX: A Novel Controller for Musical Expression*

[10] Roads, C. *The Computer Music Tutorial,* The MIT Press, 1996, pp. 656-657.

[11] Stuart, C. *The Object of Performance: Aural Performativity in Contemporary Laptop Music*

[12] VST, Steinberg's Virtual Studio Technology, http://www.steinbergcanada.com/technology/vst.htm

GAINER

A reconfigurable I/O module and software libraries for education

Shigeru Kobayashi, Takanori Endo, Katsuhiko Harada, Shosei Oishi
IAMAS
3-95, Ryoke-cho, Ogaki City
Gifu, Japan
{mayfair, endo, harada03, shosei05}@iamas.ac.jp

ABSTRACT

How to teach creating musical interface or installations to students who don't have backgrounds in electronic engineering is a longtime issue. This paper describes how it is taught at IAMAS (an institute dedicated to media arts) with the use of the newly developed environment, 'GAINER.' The GAINER environment consists of a reconfigurable I/O module and software libraries for common programming environments.

Keywords

learning, reconfigurable, rapid prototyping, sensor interface

1. INTRODUCTION

In 2004, we held a one-month workshop about physical computing[1] (mainly focused on creating musical interfaces and installations) using commercially available I/O boards, a few readymade sensors and actuators (one set lent to each student). The participants were six students who wanted to acquire skills to create interfaces or installations. Most of them had had some soldering experience (in junior high) and some programming experience (Max/MSP and Java), but little knowledge of electronics. During the workshop, we found the following difficulties:

- The participants feel a big gap between using readymade sensors or actuators and using bare sensors or actuators.
- The participants tend to break I/O ports by accident (i.e. electric overload). For most commercial products, repair parts are provided. But, it's difficult to purchase repair parts from foreign countries, so I/O modules tend to be left partly broken.
- Since configuration (e.g. number of analog inputs, PWM outputs) is fixed, the participants often find difficulty connecting sensors or actuators (e.g. the participant wants to connect a full color LED that requires three PWM outputs, but the module has only two PWM outputs). It is possible to solve this issue using proper external components (e.g. a multiplexor), but not so easy for a participant who does not have enough knowledge about electronics.
- To handle small signals from sensors (e.g. accelerometer), the participants have to add an external amplifier. It's difficult to design a proper amplifier circuit for a participant who has no experience.
- The price of the I/O board was reasonable as a commercial product, but it was a little bit expensive for students. Most of them became interested in the I/O board, but non purchased one.

As a result, the quality of the final projects (a prototype of an interface or an installation) was not satisfactory: Only one of the six students could create a prototype of an installation, the others remained at experimental level.

Based on the experience of the workshop, we started development of an environment (GAINER) for both education and actual interfaces and installations. Around that time, we decided that PSoC microcontrollers from Cypress Semiconductor would be a key component in our new I/O board[2]. The PSoC microcontroller is a mixed signal array that has configurable analog and digital blocks[3]. This flexibility allows the user to make their own configuration within the limitations of the hardware, and can change from one configuration to another on the fly. And the analog blocks have programmable gain amplifiers. This microcontroller is the key component of our I/O board.

2. CONCEPTS

Key concepts of the GAINER are as follows:

- The user starts with bare components and a solderless breadboard. This combination is used in real applications.
- The user builds their own I/O module from components by soldering to acquire basic techniques of electronics work and keep the cost of the I/O board as low as possible.
- The user can replace a broken microcontroller themselves.
- The user can choose from various configurations to suit their needs.
- The user can easily utilize a programmable gain amplifier to amplify small signals from sensors.
- The user can easily handle an I/O module with both graphical (i.e. Max/MSP[4]) and code-based (i.e. Processing[5]) programming environment .

- All I/O ports are directly connected to the I/O pins of a microcontroller. The user can enhance capabilities of I/O ports with the use of 'bridge' modules if needed.
- Open source hardware and software: Advanced users can modify existing hardware to create a new one for their project.

3. RELATED WORKS

As related works, Wiring[6], Arduino[7] and The CREATE USB Interface[8] have been proposed. GAINER different from these projects in terms of concept and implementation, but basically the same with regard to orientation. Verplank et al reported about a course on controllers at Stanford University[10], and D'Arcangelo reported about a cource on musical controllers at New York University[11]. Additionally, Lehrman et al. has proposed that creating digital musical instruments is effective in bridging the perennial gap between the arts and the sciences[9].

Wiring is an open project initiated by Barragán et al. that is a programming environment and I/O board. Arduino is a sister project to Wiring. The programming environments build on Processing, and an I/O board equips an AVR microcontroller from Atmel. Both I/O boards can be used to develop stand-alone interactive objects, or can be connected to software on a PC (e.g. Processing, Max/MSP).

The CREATE USB Interface is an I/O board project by Overholt. The I/O board equips a PIC microcontroller with built-in USB function from Microchip Technology. The user can purchase an I/O board at low cost, or purchase components and build themselves.

For Wiring, the user can purchase an I/O board at low cost, and for Arduino and the CREATE USB Interface, the user can purchase an I/O board at low cost, or purchase components and build themselves.

4. THE GAINER ENVIRONMENT

The GAINER environment consists of the following parts: A GAINER I/O module, 'bridge' modules (if needed), software libraries for programming environment on a PC (i.e. Max/MSP and/or Processing). Figure 1 shows the relationships between these components.

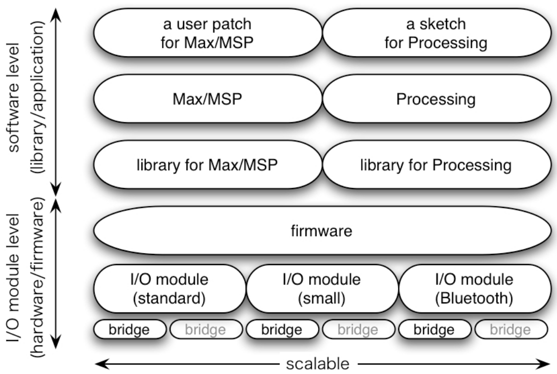


Figure 1: The structure of the GAINER environment. A user can combine a proper I/O module and a proper software platform as needed.

As mentioned in the key concepts, the user can choose from various configurations. Table 1 shows all configurations of GAINER. The first six configurations are versatile. For example, if the user chooses C4, they can connect eight potentiometers as input devices and eight LEDs with brightness control as output devices. On the other hand, the last two configurations are for specific purposes. For example, if the user chooses C8, the person can utilize four capacitive sensing switches by just connecting four electrodes and four resistors, and furthermore, the person can use an additional four digital inputs and eight digital outputs.

Table 1: All possible configurations of the GAINER I/O module. 'ain' stands for analog inputs, 'din' for digital inputs, 'aout' for PWM pseudo analog outputs and 'dout' for digital outputs.

config	ain	din	aout	dout	caption
C1	4	4	4	4	default configuration
C2	8	0	4	4	
C3	4	4	8	0	
C4	8	0	8	0	
C5	0	16	0	0	
C6	0	0	0	16	
C7	0	8	0	8	capacitive sensing
C8	0	0	8	8	matrix LED control

4.1 GAINER I/O module

4.1.1 Hardware: I/O module

Figure 2 shows the actual GAINER I/O module. The key components are a PSoC mixed-signal microcontroller (Cypress CY8C29466) and a USB-to-UART bridge (FTDI FT232RL). Except for these key components, the rest are standard and common components (i.e. a USB connector, LEDs, capacitors, resistors and so on). From a PC side, an I/O module appears as a serial port (38400bps, 1 stop bit, non-parity, no flow control).

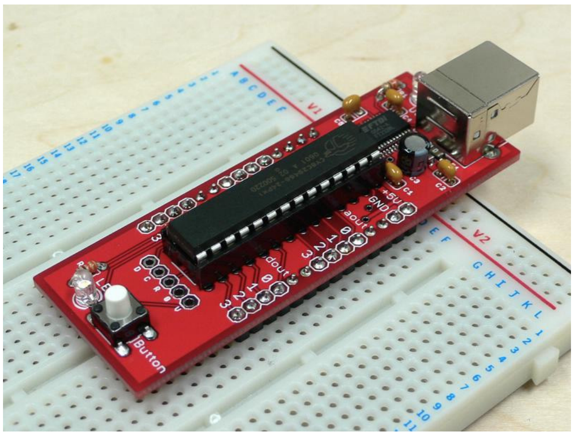


Figure 2: The GAINER I/O module. The module is placed on a breadboard with some components, and connected to a PC via a USB cable.

4.1.2 Hardware: 'bridge' modules

Figure 3 shows an actual 'bridge' I/O module to be combined with an I/O module. The right module is the 'powered output bridge' for expanding the capability of the output ports (utilizing field emission transistors). For one or two ports, the user can substantialize the same function through use of discrete components on a breadboard. But it becomes difficult as the number grows since the space of a breadboard is limited. The user can create complicated (i.e. realistic) circuts on a breadboard through the use of 'bridge' modules.

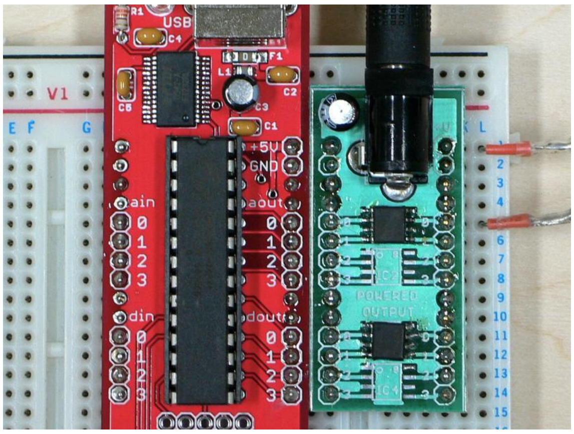


Figure 3: An examples of 'bridge' module: On the right is the 'powered output bridge.'

4.1.3 *Firmware*

While the hardware connection between the microcontrollers and the I/O ports is fixed, the configuration within the microcontroller is reconfigurable by the firmware. Figure 4 shows the internal configurations of 1 and 8. We designed specific hardware configuration and firmware for each. As shown, these internal configurations are totally different except for a UART module for communication. From the standpoint of the user, choosing or changing a configuration is a matter of simply selecting a 'patcher' (in the case of Max/MSP) or supplying an argument (in the case of Processing). Analog inputs are 8bit resolution, about 300sps speed (available in C1, C2, C3 and C4). Digital inputs are pulled-down internally. Analog outputs are pseudo analog (PWM) outputs. Both analog and digital outputs are set to 'strong' drive mode.

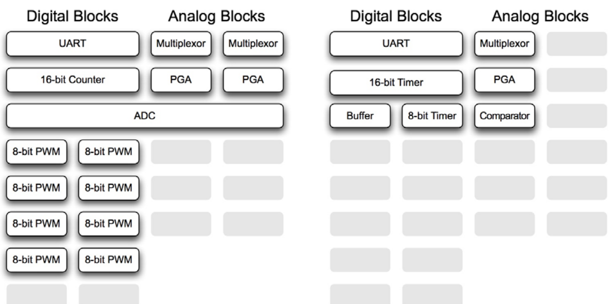


Figure 4: Examples of configurations: The left one is configuration 1 and the right one is configuration 8. The user can change from one configuration to another on the fly.

4.2 Software libraries

Figure 5 and 6 shows an example of a software library for Max/MSP. Currently, the software libraries for Max/MSP are provided as a 'patcher' (e.g. gainer.io.c1.pat) and a help patch (e.g. gainer.io.c1.help) for each configuration. As a start point, the user can use a help patch, or create their patch from scratch.

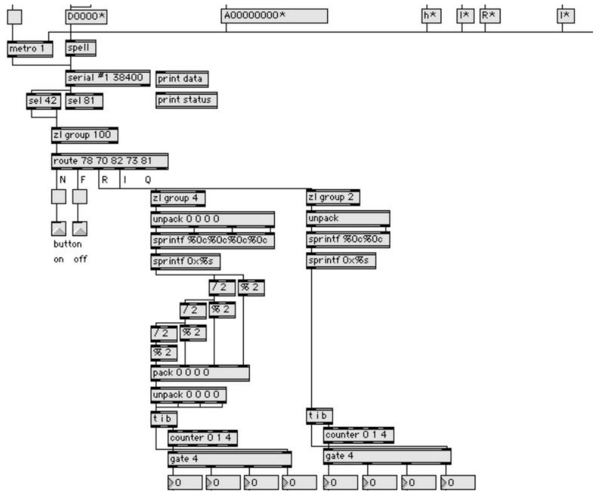


Figure 5: An example of a software library for Max/MSP. The library encapsulates low messages between a PC and an I/O board, representing them as higher level messages.

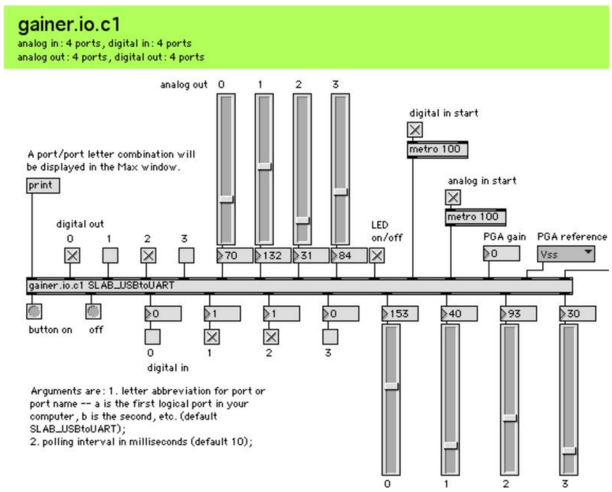
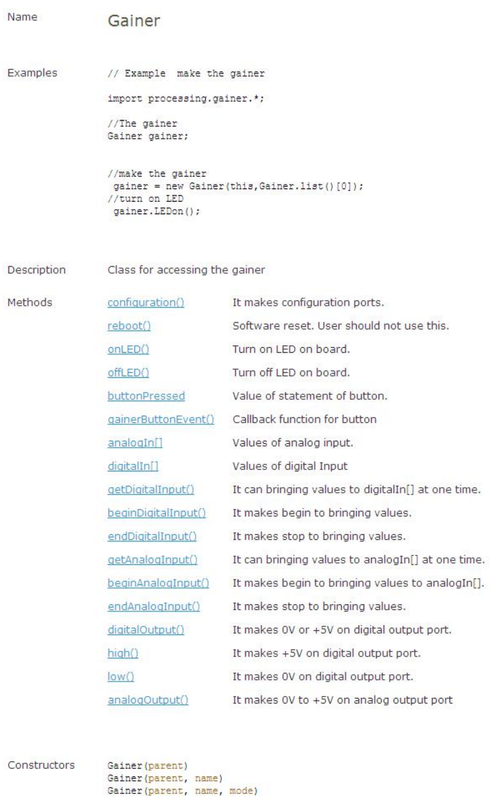


Figure 6: An example of a help patch. All connections in a configuration are displayed in a help patch. The user can start creating their own patch from a help patch.

Figure 7 shows an example of a 'sketch' in Processing environment, and figure 8 shows an example of a reference document of a class for Processing. All messaging from/to a GAINER I/O module are implemented as methods of the 'Gainer' class. A user can easily communicate with a GAINER I/O module via an instance of the Gainer class (e.g. gainer.setHigh(0)).

Figure 7: An example of a 'sketch.' Since all I/O module functions are represented as methods of the 'Gainer' class, a user doesn't have to deal low messages from/to an I/O module.



Name	Gainer	

Examples	// Example make the gainer
	import processing.gainer.*;
	//The gainer
	Gainer gainer;
	//make the gainer
	gainer = new Gainer(this,Gainer.list()[0]);
	//turn on LED
	gainer.LEDon();	

Description	Class for accessing the gainer	
Methods	configuration()	It makes configuration ports.
	reboot()	Software reset. User should not use this.
	onLED()	Turn on LED on board.
	offLED()	Turn off LED on board.
	buttonPressed	Value of statement of button.
	gainerButtonEvent()	Callback function for button
	analogIn[]	Values of analog input.
	digitalIn[]	Values of digital Input
	getDigitalInput()	It can bringing values to digitalIn[] at one time.
	beginDigitalInput()	It makes begin to bringing values.
	endDigitalInput()	It makes stop to bringing values.
	getAnalogInput()	It can bringing values to analogIn[] at one time.
	beginAnalogInput()	It makes begin to bringing values to analogIn[].
	endAnalogInput()	It makes stop to bringing values.
	digitalOutput()	It makes 0V or +5V on digital output port.
	high()	It makes +5V on digital output port.
	low()	It makes 0V on digital output port.
	analogOutput()	It makes 0V to +5V on analog output port
Constructors	Gainer(parent) Gainer(parent, name) Gainer(parent, name, mode)	

Figure 8: An example of a reference document. A user can easily open reference documents to see detailed descriptions.

5. CURRICULUM

We held a workshop on physical computing from November 8th to December 5th at IAMAS in 2005. IAMAS consists of a specialized training college (International Academy of Media Arts and Sciences) and a graduate school (Institute of Advanced Media Arts and Sciences). The participants numbered ten (eight from the specialized training college and two from the graduate school). All of them were interested in creating interfaces and/or installations. All of them had a certain degree of experience of programming (e.g. Max/MSP and/or Java), but none of them had knowledge about electronics. We had three hours per day, two days (on Mondays and Thursdays) per week.

The schedule of the workshop was as follows:

- week 1, day 1: Basic electronics
  - How to use a breadboard?
  - Ohm's law
  - Turn on a LED with a resistor
  - Ordering components of your I/O module
- week 1, day 2: Building your I/O module
  - How to do soldering?
  - Building your I/O module
  - Testing the I/O module
- week 2, day 1: How to handle outputs?
  - How to use software libraries for Max/MSP?
  - How to connect a LED?
  - How to connect a SSR?
  - How to connect a R/C servo motor?
- week 2, day 2: How to handle inputs?
  - How to connect a switch?
  - How to connect a potentiometer?
  - How to connect a CdS?
  - How to connect an accelerometer?
- week 3, day 1: How to process data in Max/MSP?
  - How to do scaling in Max/MSP?
  - How to do data processing in Max/MSP?
  - How do you map incoming data to outputs?
- week 3, day 2: What is Processing?
  - Basic introduction about Processing
  - How to use software libraries for Processing?
  - How to handle inputs?
  - How to handle outputs?
  - How to process data in Processing?
- week 4, day 1: Plan presentation
  - Show a plan for the final presentation
  - Discussion with lecturers
  - Order components and materials
- week 4, day 2: Building
  - Build a prototype
- week 5, day 1: Final presentation
  - Present one's prototype
  - Discussion
  - Closing

Figure 9 shows a scene of the second day of the first week of the workshop.

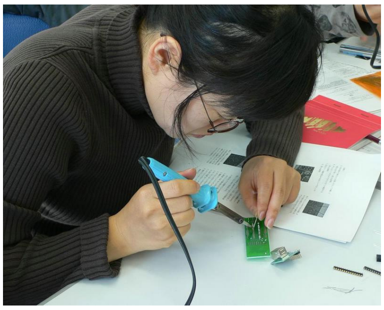


Figure 9: A student is soldering to assemble her own I/O module. Most participants had some soldering experience in junior high.

6. RESULTS

6.1 Winter 2004 student projects

At the final presentation, all participants presented as described in Table 2. Although the preparation period was short (about one week), most students presented a working prototype as shown in Figure 10. Additionally, a few students actually used prototypes in their musical performances or installations in the weeks following after the final presentation. With regard to configuration, eight students used the default configuration (C1), and two students used another configuration (C4) as needed.

Table 2: The breakdown of the final presentation

type	number
musical interface	6
installation	3
performance	1

Figure 10: A student is presenting his prototype to all participants.

Figures 11, 12 and 13 show examples of the final projects in musical interface category. The first one is a musical interface consisting of RGB color sensors and a full color LED. The RGB color sensors on the bottom face recognizes a color, then PC side software turns the color information into sounds. An I/O module is used to handle the sensors and the LED. The second one is a musical interface consisting of pressure sensors and a CCD camera-based computer vision. A performer plays the instrument with their hands, and changes parameters by pressing on the sensors. An I/O module is used to handle pressure sensors. The third one is a musical interface consisting of CdS sensors and LEDs under a half-mirror. A performer plays the instrument with a bulb light. When a sensor detects the light, a corresponding LED is turned on and a sound is played for a short while. An I/O module is used to handle sensors and LEDs.

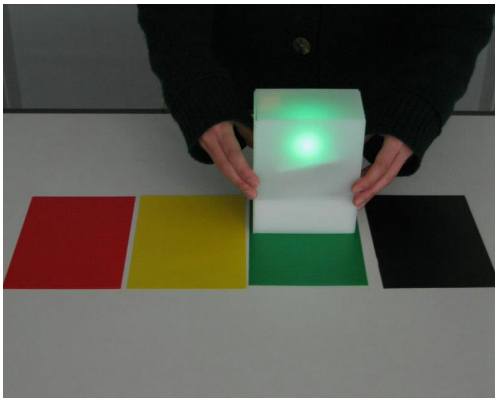


Figure 11: Input is color information from a RGB color sensor. Output is a light from a RGB color LED and corresponding sounds from a PC.

Figure 12: Input is color information from a CCD camera and pressure information from pressure sensors. Output is visual feedback on a projected screen and corresponding sounds from a PC.

Figure 13: Input is brightness information from CdS sensors. Output is lights from white LEDs and corresponding sounds from a PC.

6.2 The results of questionnaires

We asked students to answer questionnaires several times throughout the workshop. First of all, the result of "Was the workshop of interest to you?" was 4.44 (5 was "very well" whereas 1 "not well.", SD = 0.96). According to this result, the workshop was of interest to most of the students.

Secondly, the result of "Did you easily assemble your I/O module?" was 3.33 (SD = 0.80). According to this result, the degree of difficulty was thought to be reasonable.

And the result of "Did you become interested in the GAINER after assembling by yourself?" was 3.78 (SD = 1.03). Except for two students, the score was 4 or 5. The two students who scored 2 to this question gave supplemental answers to the question as follows: "Since no interesting real applications are presented currently, I'm not very interested." This was one of the points of reflection about the curriculum of the workshop.

Q1 in Table 3 shows understanding of electronics before and after the workshop. According to the result, the understanding of electronics seems to have been deepened through the workshop.

Q2 in Table 3 shows understanding of programming before and after the workshop. According to the result, there was no change about the understanding of programming. We hoped that the understanding of programming would increase as it did in electronics, but it didn't. We think that there was not enough time for most of the students to understand both electronics and programming simultaneously.

Q3 in Table 3 shows interests in creating works with electronics. According to the result, the score is mostly same before and after the workshop. Before the workshop, the students were highly motivated. But during the workshop, they experienced many difficulties in creating their final projects. In spite of the difficulties, they remained highly motivated throughout the workshop. We think that the workshop was meaningful for the students.

Table 3: An excerpt of questions: Q1 was "How well do you understand electronics?" Q2 was "How well do you understand programming?" Q3 was "Do you want to create works with electronics?" The numbers shown in parentheses are SD values.

	before	after
Q1	1.67 (0.47)	2.78 (0.91)
Q2	2.72 (0.85)	2.72 (0.85)
Q3	4.44 (0.68)	4.33 (0.81)

7. CONCLUSIONS AND FUTURE WORK

According to the results of the workshop, we achieved some positive results with utilizing the GAINER environment. During the workshop, we couldn't explore possibilities of the reconfiguration in deep. According to the questionnaire after the workshop, 66% of the students wanted to attend an advanced workshop. We have no detailed plans for an advanced workshop, but we would like to explore possibilities of the reconfiguration in the workshop. In regard to scalability, we would like to provide a wider range of I/O modules and 'bridge' modules as follows:

- Documentation in multiple languages (currently in single languages only).
- Smaller I/O modules to be embedded into a small device (to expand scalability).
- Stand-alone capability (both firmware side and development tool side).
- Wireless connection capability (e.g. Bluetooth, partially tested).
- More software libraries (e.g. PureData, Adobe Flash and so on).

In regard to the effectiveness of the GAINER environment and the workshop, it's hard to do control experiments. So we want to keep holding the workshop for a few years to examine its effectiveness and to improve the environment.

8. ACKNOWLEDGMENTS

This research has been a part of the 'Programmable Device Project' (PDP)[12] at IAMAS. The authors wish to acknowledge of the assistance of Masayuki Akamatsu for suggestions as a media artist, Takahiro Kobayashi for technical advice, Kazuki Saita for reviewing the documentations and all members of the PDP and workshop attendees for cooperative suggestions. The source code of the firmware, the hardware and software libraries are available at the following location: http://gainer.sourceforge.net/

9. REFERENCES

[1] O'Sullivan, D. and Igoe T. Physical Computing. Muska LipmanPremier-Trade, 2004.

[2] Seguine, D. Just add sensor - integrating analog and digital signal conditioning in a programmable system on chip. Sensors, 2002. *In Proceedings of IEEE* Volume 1, 12-14, p.665–668.

[3] Ashby R. Designers Guide to the Cypress PSoC. Newnes, 2005.

[4] Cycling '74. Max/MSP. Available at: http://www.cycling74.com/

[5] Reas, C. and Fry, B. Processing: a learning environment for creating interactive Web graphics. *In Proceedings of the SIGGRAPH 2003.*

[6] Wiring. Available at: http://wiring.org.co/

[7] Arduino. Available at: http://arduino.berlios.de/

[8] Overholt, D. The CREATE USB Interface - where art meets electronics. Available at: http://www.create.ucsb.edu/˜dano/CUI/

[9] Lehrman, P. and Ryan, T. Bridging the Gap Between Art and Science Education Through Teaching Electronic Musical Instrument Design. *In Proceedings of New Interfaces for Musical Expression* (NIME05) (Vancouver, Canada, May 26–28). p.136–139

[10] Verplank, B. A Course on Controllers. *In Proceedings of New Interfaces for Musical Expression* (NIME01) (Seattle, USA, April 1–2)

[11] D'Arcangelo G. Creating a Context for Musical Innovation: A NIME Curriculum. *In Proceedings of New Interfaces for Musical Expression* (NIME02) (Dublin, Ireland, May 24–26)

[12] Programmable Device Project. http://www.iamas.ac.jp/project/pdp/

Hyper-shaku (Border-crossing): Towards the Multi-modal Gesture-controlled Hyper-Instrument

Kirsty Beilharz
Key Centre of Design Computing & Cognition
University of Sydney
+61 2 9351 4031
kirsty@arch.usyd.edu.au

Joanne Jakovich
Key Centre of Design Computing & Cognition
University of Sydney
+61 2 9351 2053
joanne@jakovich.net

Sam Ferguson
Acoustics Research Laboratory
Faculty of Architecture
University of Sydney
samferguson@ihug.com.au

ABSTRACT
Hyper-shaku (Border-Crossing) is an interactive sensor environment that uses motion sensors to trigger immediate responses and generative processes augmenting the Japanese bamboo *shakuhachi* in both the auditory and visual domain. The latter differentiates this process from many hyper-instruments by building a performance of visual design as well as electronic music on top of the acoustic performance. It utilizes a combination of computer vision and wireless sensing technologies conflated from preceding works. This paper outlines the use of gesture in these preparatory sound and audio-visual performative, installation and sonification works, leading to a description of the *Hyper-shaku* environment integrating sonification and generative elements.

Keywords
Gesture-controllers, sonification, hyper-instrument

1. INTRODUCTION
Background developments contributing to the *Hyper-shaku* project are intelligent sensor environments (sensate spaces) and gesture-controller interactive audio-visual works. This paper describes the gesture-controlled audio-visual hyper-instrument and previous interactive gestural works that have led to it. Human movement, social and gesture data used as the foundation for sonification and visualization are shown, an alternative to the common sonification process of analyzing and representing abstract, non-contextual data. In contrast, gesture-data and room-data are user-centric, contextual, situated and experiential. In this way, ambient display and installation art influences conflate in informative data-driven aesthetic displays (bimodal audio and visual). The responsive/intelligent room using pressure and motion sensing demonstrate environmental data used as the basis for sonification. The sensor technologies are essentially like embedded, passive gesture captors that track mobility in the environment, not worn and portable sensors. Ambient display in architectural spaces provides interesting information about the inhabitants and activities of a location in the socially reflective experience. Several of the works discussed implement generative structures integrated with information representation for interactive installation. Generative algorithmic structures provide a representation with a consistent mapping scheme and transforming, evolving display that is intended to enhance sustainable participation and motivation over longer periods of time. The example works from performance to installation in the first part of this paper have shaped the technology integration for the hyper-instrument: wireless gesture-controllers, computer vision motion triggering and real-time generative displays in Max/MSP + Jitter.

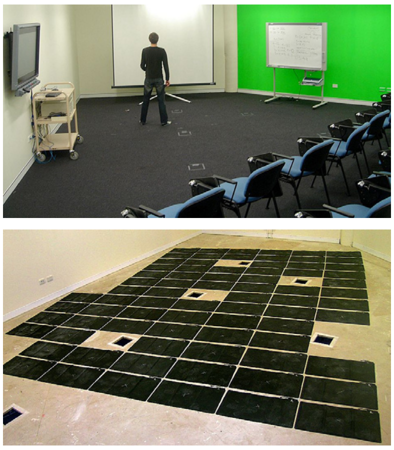
The second part of this paper is concerned with development of a gesture-controlled hyper-instrument (system). A key feature, distinguishing this from other hyper-instruments, is transforming the acoustic instrument into a hyper-instrument capable of both augmented audio and visual display. The system aims to control not only gesture-response events but also to trigger generative design processes affected by movement. Hence, the proposed system is a performance "environment" for multiple related compositions. It can be used to augment purpose-composed notated music for *Hyper-shaku* or in an improvisatory audio-visual performance context.

2. FORMATIVE WORKS
Gestural interaction, sonification and generative display in the following works influence the design of *Hyper-shaku*. Responsive/reactive spaces are discussed, followed by works that respond to gestural interpretation of space in performative works. This section considers mapping correlations between spatial activity and auditory display, in order that gestures can be understood by the interface-user and the audience.

2.1 Audio-visual Responsive Environments: Reactive Space
Other audio-visual responsive spaces, such as the Golan Levin's work, *Eyesweb,* and other systems for movement capture using computer vision establish the concept of pervasive and responsive display is socio-spatial contexts. Tod Machover's Hyperinstruments group at MIT Media Lab also addresses visual feedback in instrument design while Andy Hunt's MIDAS programming environment and other work examines auditory and visual mapping of gestures that has contributed to the formation of this approach.

Emergent Energy developed in the University of Sydney, Key Centre of Design Computing and Cognition's sensate lab (Figures 1, 2 & 3) demonstrates the way in which socio-spatial behaviours are mapped onto a computational process of sonification and visualization. Beilharz, Vande Moere and Scott's *Emergent Energy* (Figure 1) is an iterative, reflexive bi-modal (audio-visual) system of interaction in which motion, speed, number of users and position in a space (triggering pressure-sensitive floor mats) determine the growth of a visual design drawn with a Lindenmayer System (L-sys) generative algorithm [3; 7; 17; 22; 23; 24]. The design artefact is an embedded history of the movements, interactions and number of people who produced it. Sound is a spatial experience, inseparable from context [21] so it is logical to utilize 3D spatial interaction to measure activity and manipulate sound.

Figure 1. Beilharz, Vande Moere & Scott's L-system generator patch in Max/MSP + Jitter software [15] used to create branched visualizations on screen. In the corresponding sonification, the number of people relates to dynamic intensity, position to *timbre* (tone colour) and speed to frequency (pitch) [3].



Figures 2. & 3. The Sensate Lab (2 views) showing the "invisible" pressure sensitive floor mats, triggering the visual and auditory sound system and (bottom) before carpeting, networked to the Teleo (analog to digital in/out) modules for conversion to a USB interface [20].

Sonic Kung Fu by Jakovich and Beilharz (at *Sydney Esquisse* exhibition, March 2005) is a sonic art installation using colour-sensing gestural interaction with sound, in which participants wear coloured gloves to perform gestures that produce a real time responsive audio soundscape (Figure 4). A web cam receives the visual gesture information. The Max/MSP patch responds to the motion of the centre-point of a specific colour (calibrated to match the glove being worn), responding with auditory variation across a range of *x* and *y*- axis values. The immediacy and mapping of this work was intentionally as simple and intuitive as possible for recognition to invoke interaction by passers-by in a gallery setting. The result was that users spent considerable time with the "instrument" learning to understand and control its performance.

Figure 4. Gestural interaction with auditory display created in response to colour tracking of the spatial glove motion.

2.2 Gesture Mapping for Auditory (and Visual) Display: Interpreting Space

Correlating/mapping gesture to responsive representation involves the design decisions most crucial to comprehensibility and intuitive interaction [1; 2; 25]. Depending on the context, the degree to which gestures and reflected consequences have to be learned varies. In the public sphere, like in Jakovich and Beilharz's *Sonic Tai Chi* Sydney Powerhouse (Design and Technology Museum) installation, the audience is transient, covering a range of ages from children to adults and the immediacy of engagement determines the length of time a user will participate in the display. Regardless of the simplicity of mapping correlations, users seemed to naturally pay primary attention to the visual display and, when questioned, it took longer for users to understand and explain their interpretation of the relation between their movement and auditory display than both the literal and generative visual display elements.

Sonic Tai Chi uses a computer vision system (identical to the method in *Hyper-shaku)* to capture movement data to produce a visualization comprised of the interpolated real image of the user combined with random Cellular Automata and the music is a sonification of the motion left to right and up and down with pitch, spatial panning, timbre and intensity affected by user interaction. A second sonification engine produces audio particles from the position, multiplicity and intensity of the Cellular Automata that can be triggered into rapid proliferation (using the breeding metaphor of aLife) by moving the body in one horizontal direction across the room and towards stasis by moving in the opposite direction (Figure 5). This piece is designed for spatial interaction by the general public. It has its own approximately 25m-square room, rear projection, stereo speakers hidden in the walls and camera concealed below the screen.

Max/MSP + Jitter uses the Horn-Schunk method to estimate optical flow of movement captured by the web cam [16]. There are numerous possibilities of rules to govern the propagation of Cellular Automata [9; 26] but this scenario uses the original, quite simple rules for pattern formation based on John Conway's *Game of Life.*

Figure 5. ***Sonic Tai Chi*** **(BetaSpace, Sydney Powerhouse Museum, installation) uses computer vision to capture movement data that produces the vizualisation and sonification.**

The *Sensor-Cow* project uses the La Kitchen Kroonde Gamma receiver and wireless UDP[1] transmitter and gesture captors. The sensors used were acceleration, gyroscopic and bi-directional motion captors. Figure 6 shows the way in which these sensors and transmitter are attached to the calf for capturing the data. The outcome was a sonification of the calf's motion.

The highly sensitive mercury motion sensors operate between extremes of direction, registering a "bang" (signal to the sonification program) when changes in direction occur. Thus these were attached to the front legs to indicate steps as the calf walked. The acceleration sensor values were scaled to 128 distinct output values. These sensors were attached to the calf's ear and forehead, respectively, because these regions isolate significant independent gestures. The calf naturally raises and lowers its head to eat, when flicking away flies, in response to people and other animals - it is expressive and the range of motion is diverse. While naturally following whole head movements, the ear is also flicked and rotated independently producing an audibly recognizable gesture.

A distinctive *timbre* was attributed to each sensor in order to make it possible to distinguish the sounds arising from each sensor region. The rhythm, pace/acceleration and velocity of action are heard in real time. Hence the correspondence between rapid gestures and rapid sonification is literal. For both the acceleration and gyroscopic sensor, extremes of motion away from the median, drives the pitch in directional extremes away from a central pitch region. The direction of pitch, ascending and descending away from the mean, corresponds to the *x*-axis direction of motion so that changes in direction are audible and circular motions of the ear and head produce sweeping auditory gestures that reinforce the audio-visual connection between activity and sonification. The sonification was programmed in Max/MSP +Jitter using La Kitchen's Kroonde Gamma recognition [14] and CNMAT Berkeley's Open Sound Control [10].

The Music Without is concerned with exposing the motion of music. Real time computer music responds to sensors placed on the violinist's left-hand finger and forearm and the bowing arm. The gyroscopic, binary-motion and acceleration sensors convey the intensity, physicality and movement (outside forces) that performing involves. Typically, we think of the music within, of the source of musical creation being the mind (composer) and the heart (interpretation). Most reactive, responsive computational real time music systems analyze and respond to pitch, harmony and rhythm. Thus, most systems for improvisation and collaboration are responding to the musician's inner music by "listening" to the auditory outcome.

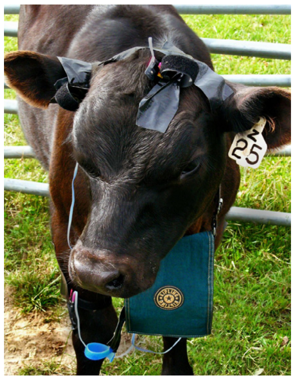


Figure 6. Bi-directional (mercury) motion sensors are attached to the calf's front legs, a gyroscopic sensor on the forehead and accelerometer on his right ear. The pouch hanging around his neck contains the radio frequency transmitter that sends the real time data to the (La Kitchen) Kroonde Gamma wireless UDP receiver [18]. It is connected by Ethernet to the computer running the data sonification with Max/MSP object-oriented programming environment.

In contrast, this system creates a response to the physical forces producing sound; hence 'the music without' is more like choreography. The "other musician" here is a sonification of the external energies creating music. The system is generating a musical response to gestures perceived by the sensor devices. It is not so much listening as feeling, or experiencing, the process of performing. This work emphasizes a different and often overlooked part of the music-creating process.

3. HYPER-SHAKU (BORDER CROSSING): AUGMENTING SPACE

Hyper-shaku is a new hyper-instrument performance environment that uses motion to trigger response events and growth of generative process in both auditory (electronic) and visual displays. Its purpose and configuration follow.

3.1 Description and Objective

Hyper-shaku (Border-Crossing) is both a digital audio-visual creative environment and a performance/composition outcome.

[1] UDP is a protocol for high speed, high precision data-acquisition.

The motivation behind this application is two-fold:

1) To develop a system of computer vision and sensors producing an augmented sound-scope and derivative visual projection (that will be applied to a prototype and continuing works); and
2) To demonstrate the prototype with an initial concert installation (by performance with a composed, notated *shakuhachi* part).

This project develops an ongoing framework of computer vision for capturing movement of a performer, together with wireless sensor information to trigger a generative computer system. The generative part of the process produces a motion-activated fabric of computer music and graphic visualization. First stages of using computer vision with web-cam sensing and Max/Jitter software patches integrating Pelletier's cv.jit 'Computer Vision for Jitter' programming objects[2] were implemented in Jakovich and Beilharz's *Sonic Kung Fu* (Sydney Esquisse Exhibition) and *Sonic Tai Chi* (Cité Internationale des Arts, Paris in September 2005 & Sydney Powerhouse BetaSpace Exhibition in November-January 2005-2006). Radio-frequency gesture controllers (similar to the proposed WiFi sensors in this project) were used in *Sensor-Cow* and *Music from Without.* Musically, the intention of this project is to utilize this technological approach in a more developed electro-acoustic musical and visual context. Rather than producing literal sonification and visualizations as in previous works, this project will use the motion data to trigger synthesis processes in real time and to control a computational generative process for sound and visual design. Former work demonstrates the use of a Lindenmayer generative system (in *Emergent Energies)* and Cellular Automata *(Sonic Tai Chi)* for evolving graphical responses to user interaction. The generative part of the process produces a motion-activated fabric of computer music and graphic visualization. *Hyper-shaku* combines an aleatoric generative process using Cellular Automata with a homeostatic process using Neural Network Oscillators. The breeding behaviour of the Cellular Automata is moderated by the large upper body movements of the player, tracked by the web cam. Within the Neural Network Oscillators, triggers in individual "neurons" (Max/MSP software model, Figure 7) instigate moments of excitement that infect other neurons [11]. Over time, the effect of one neuron in the network influencing another develops stabilizing homogeneity, as gradually the neurons resemble and emulate one another. The combination of the chaotic and vigorous process of the Cellular Automata and the stabalizing, homeostatic nature of the Neural Network Oscillators [12] provides a suitable excitement-stasis balancing structure for the production of long background transitions, behind a foreground of dynamic activity responding to the live performance. Algorithmically, the output of each neuron in the network is determined by the weighted outputs of every other neuron. The critical threshold of perturbation, beyond which reorganization is triggered, is an adjustable parameter in the Max/MSP model. This is akin to the musical effect of stable harmony so the metaphor provides a suitable excitement-stasis balancing structure for long background transitions, behind the foreground of live performance activity and relatively immediate, agile C.A. responses to the wireless captors. The fundamental units of an artificial neural network (units/nodes/neurodes) are modeled after individual neurons: its dendritic tree collects excitatory and inhibitory inputs from other neurons (the 'receives' in the Max/MSP model), and passes these messages, as voltages, on to the cell body (soma) (see Figure 7). These voltages are added to the current voltage if excitatory and subtracted if inhibitory. When the threshold is exceeded, a signal is transmitted [the 'sends' in the Max/MSP model] down an axon to synapses that connect the tree to dendrites of other neurons [13].

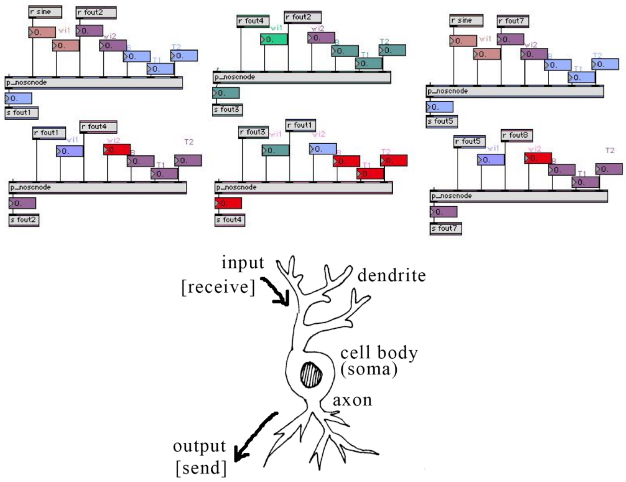
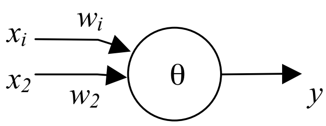


Figure 7. The Max/MSP Neural Oscillator Network patch, here showing the first 6 nodes, each sending and receiving information between nodes [12] that will be used as a stabilizing influence affected by large camera-tracked gestures. The Max/MSP patch is modeled on individual neurons: dendrites receive impulses and when the critical threshold is reached in the cell body (soma), output is sent to other nodes in the neural network.

Neurons (also called a linear threshold device or a threshold logic unit) can be modeled formally (mathematically) (Figure 8):



x_i -- the inputs
w_i -- the weights (synaptic strengths)
θ -- the threshold
y -- the output

$$y\,(t+1) = \begin{cases} 1 & \text{if} \quad \sum_i w_i x_i\,(t) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Figure 8. A symbolic simplification of a neuron with a formal model of threshold.

The new media technologies (the software patch and methodology) from this project will be applied and adapted to a series of future works, each unique because it is a responsive interactive system, a synergy of notated, performed music and sound and visual material generated from the performer's gestures. The author's chamber concerto for *shakuhachi* and

[2] Jean-Marc Pelletier (IAMAS)
http://www.iamas.ac.jp/~jovan02/cv/

ensemble, *The White Face of the Geisha,* performed by Yoshikazu Iwamoto with Ensemble Recherche, Freiburg (2000) and Jeffrey Lependorf's article, 'Contemporary Notation for the *Shakuhachi:* A Primer for Composers', in *Perspectives of New Music* [19] provide some background in idiomatic techniques, notation and articulations.

The physical nature of playing the *shakuhachi* makes it especially suitable for motion triggering since pitch inflection is achieved by angling the chin relative to the instrument and dramatic *atari* (articulation) attacks and head vibrato are part of the ornamentation approach to pitch production, in addition to fingering and upper body movement typical performing an instrument. Traditional live music-processing approaches analyze and synthesize real time musical response from the musical (audio) content of a performer. The approach of this project, in contrast, focuses on the **gestural/spatial** and theatrical nature of *shakuhachi* performance. The whole system is an "environment" - a hyper-*shakuhachi,* augmenting the sound scope from traditional sounds of the bamboo end-blown Japanese flute to include computer-generated music and visual images for a single-performer holistic presentation.

The reason for this project is multi-fold: to stimulate the interest in a traditional instrument; to augment its capability into the multidisciplinary, trans-medial realm of electronic music as well as physical, acoustic sound; to re-invigorate interest in traditional instruments amongst Japanese and other audiences with listening tendencies moving towards Western or technologically-enhanced listening. There is more interest in the traditional Japanese instrument in the U.S.A. and Australia (with its great cultural inheritance of *shakuhachi* players like Riley Lee, James Franklin, Andrew McGregor) than in Japan [8]. Hybridization with technologies and a new approach bringing its attention to a new and possibly younger audience potentially contributes to a new role for the instrument. In addition, traditional repertoire is extremely ancient and there is not very much contemporary repertory or performance context for this instrument, hence a multimedia environment positions it in a contemporary performance context.

This project will develop the method and a prototype piece. The method is transferable to other instruments (though specialized here for the *shakuhachi)* as well as further improvisation and composed performance pieces. The prototype will be developed with a notated *shakuhachi* part to demonstrate the development for the first performance exposure but the infrastructure will also contribute to subsequent creative work in the field of gesture-controlled hyper-instruments by the composer.

3.2 Technical Configuration

Method & configuration are illustrated in figure 9. A continuation of the web cam computer vision system based on earlier works and Pelletier's Computer Vision Jitter objects are used to capture visual data about the *shakuhachi* performer (using luminosity tracking). Wi-Fi (wirelessly transmitting) sensors[3] capture and convey motion data from the performance in real time. Both data inputs will be processed in Max/MSP + Jitter (visual programming environment) to produce both real time auditory augmentation and visualization of the performance energies with some additional generative design (building and transforming over time). Cellular Automata (for random material), L-system (for tree-like growth) generative algorithms will be applied to visual display in Jitter, along with the musically stabilizing Neural Oscillator (NOSC) Network for sound only. The electronic augmentation is two-fold: comprising event-response immediate reaction events activated by the different motion captors and isolated captors trigger successive events in the aleatoric (C.A.) and generative (L-system) hyper environment. Signals emanating from distinct captors are mapped to individual *timbres* in order to distinguish subtle effects of isolated gestures. A second sonification engine, similar to the approach demonstrated in *Sonic Tai Chi,* will display an auditory representation corresponding to the C.A. visual generative design. The sonic material for the sound synthesis will be real-time samples captured from the live *shakuhachi* performance, in which the granularity and synthesis parameters are determined/affected by degree of motion (calibrated according to individual captor types). Sound is played back alongside the performer with adequate stereo separation to optimize spatial panning in response to horizontal motion detected by the Computer Vision and the visual display is projected onto an on-stage screen. The notated *shakuhachi* part for a prototype performance is fully composed.

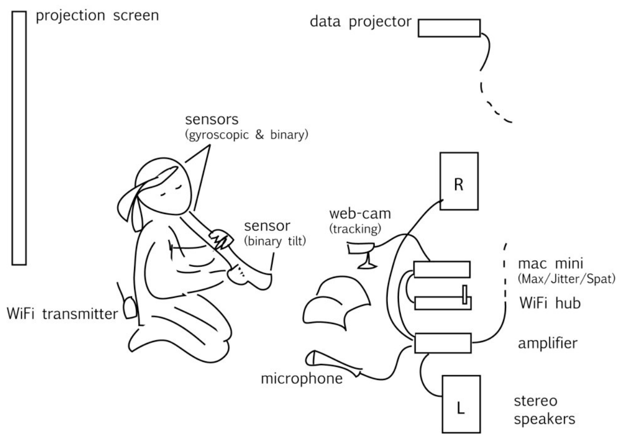


Figure 9. Configuration for performance. Visual and motion data is captured by wireless sensors and web cam producing augmented real time audio and feeding a generative audio-visual design system in Max/MSP + Jitter.

4. CONCLUSION

Hyper-shaku is a confluence of technologies, sonification and generative methods in a growing body of interactive work. This paper briefly outlines the transition of gestural technologies into the hyper-instrument environment that augments the acoustic instrument in both auditory and visual domains.

5. ACKNOWLEDGEMENTS

Thanks to the University of Sydney Sentient Lab (sensate lab), Key Centre of Design Computing and Cognition and to collaborators: Andrew Vande Moere, Joanne Jakovich and Amanda Scott; exhibitors, BetaSpace (Sydney Powerhouse Museum) and *Sydney Esquisse.* Beilharz's research is supported by a University of Sydney Bridging Support Grant 2006. Jakovich is assisted by an Australian Postgraduate Award and Collaborative Research Centre for Construction Innovation Ph.D Scholarship.

[3] WiFi protocol wireless sensor transmission, e.g. Emmanuel Flety's WiSe Box

6. REFERENCES

[1] Beilharz, K. (Criteria & Aesthetics for) Mapping Social Behaviour to Real Time Generative Structures for Ambient Auditory Display (Interactive Sonification) *in* E. Edmonds and R. Gibson (eds)*, INTERACTION - Systems, Practice and Theory: A Creativity & Cognition Symposium*, Creativity and Cognition Press, The Dynamic Design Research Group, UTS, Sydney, 2004a, pp. 75-102.

[2] Beilharz, K. Interactively Determined Generative Sound Design for Sensate Environments *in Sound and Music Computing*, IRCAM, Paris, 2004b.

[3] Beilharz, K. Responsive Sensate Environments: Past and Future Directions: Designing Space as an Interface with Socio-Spatial Information *in* B. Martens and A. Brown (Ed.s)*, CAAD Futures*, Wolters Kluwer Academic Publishers, Netherlands, Vienna, 2005a, pp. 361-370.

[4] Beilharz, K. Gesture-Controlled Interaction with Aesthetic Information Sonification, *in* Pisan, Y. (Ed.) *The Second Australasian Conference on Interactive Entertainment,* Sydney, 2005b, pp. 11-18.

[5] Beilharz, K. Wireless Gesture Controllers to Affect Information Sonification, *in* Brazil, E. (Ed.) *ICAD,* Limerick, Ireland, CD-ROM proceedings July, 2005c.

[6] Beilharz, K. Architecture as the Computer Interface: 3D Gestural Interaction with Socio-Spatial Sonification, *in* Duarte, J.P., Ducla-Soares, G. and Zita Sampio, A. (Ed.s) *Digital Design: The Quest for New Paradigms - eCAADe* Proceedings, Lisbon, 2005d, pp. 763-770.

[7] Bentley, P. *Evolutionary design by computers*, Morgan Kaufmann Publishers, San Francisco, CA, U.S.A, 1999.

[8] Blasdel, C. *The Single Tone,* Printed Matter Press, Japan, 2005.

[9] Burraston, D. Structuring Cellular Automata Rule Space with Attractor Basins, *in* Edmonds, E., Brown, P. and Burraston, D. (Ed.s) *Generative Arts Practice '05: A Creativity and Cognition Symposium,* University of Technology Sydney, 2005, pp. 21-40.

[10] CNMAT Open Sound Control (OSC) http://cnmat.cnmat.berkeley.edu/OSC/Max/#downloads, 2004.

[11] Eldridge, A. Fond Punctions: Generative Processes in Live Improvised Performance, *in* Edmonds, E., Brown, P. and Burraston, D. (Ed.s) *Generative Arts Practice '05: A Creativity and Cognition Symposium,* University of Technology Sydney, 2005, pp. 41-51.

[12] Eldridge, A. NOSC (Neural Oscillator Nodes) Open-source Max/MSP [patches & abstractions], Creative Systems Lab, Evolutionary and Adaptive Systems Group, Department of Informatics, University of Sussex, U.K., 2005.

[13] Franklin, S. *Artificial Minds* MIT Press, U.S.A., 1995, this ed. 1999.

[14] Henry, C. *La Kitchen Kroonde Gamma - User Manual & Max Patches*, La Kitchen, Paris, 2004.

[15] IRCAM. Documentation & Tutorials for Max/MSP software by Cycling 74: Max/MSP is a graphical object-oriented programming environment for real time control of music, audio, and multimedia objects, capable of hardware and software integration, U.S.A., 2003. http://www.cycling74.com/products/maxmsp.html

[16] Jakovich, J. and Beilharz, K. 'Sonic Tai Chi': Auditory Gestures in Reactive Soundspaces (demo paper and installation) *in* Pisan, Y. (Ed.) *The Second Australasian Conference on Interactive Entertainment,* Sydney, 2005, p. 243.

[17] Kreveld, M. V., Overmars, M., Schwarzkopf, O., Berg, M. d. and Schwartskopf, O. *Computational Geometry: Algorithms and Applications*, Springer Verlag, 1997.

[18] LaKitchen. Kroonde Gamma 16-sensor Wireless UDP Interface, 2004. http://www.cycling74.com/products/kroonde.html 03/08/04.

[19] Lependorf, J. Contemporary Notation for Shakuhachi *in* (Henry Burnett and Hilary Tann, Ed.s) Tradition and Renewal in the Music of Japan, *Perspectives of New Music,* **27**: 2 (Summer), University of Washington, WA U.S.A., 1989.

[20] MakingThings. Teleo Modules and Sensor Devices, 2003. http://www.makingthings.com/products/products.htm 1/1/04.

[21] Pottier, L. and Stalla, O. Interpretation and Space *in* W. W. Wanderley and M. Battier (Ed.s)*, Trends in Gestural Control of Music*, IRCAM (Institute of Research in Computer and Acoustic Music) - Centre Pompidou, Paris, 2000.

[22] Skiena, S. S. *The Algorithm Design Manual*, Telos/Springer-Verlag (Book and CD-ROM edition), 1997.

[23] Skiena, S. S. Analysis of Algorithms for Music, 2003. http://www.cs.sunysb.edu/~algorith/lectures-good/index.html 21/11/03.

[24] Vande Moere, A. and Beilharz, K. Interaction Design for Smart Spaces – a Constructivist Design Studio, *in Proceedings of the 38th Australian & New Zealand Architectural Science Association (ANZAScA) Conference*, University of Tasmania, Hobart, 2004, pp. 335-342.

[25] Walker, B. N. and Kramer, G. "Mappings and Metaphors in Auditory Displays: An Experimental Assessment", *in International Conference on Auditory Display (ICAD)*, Palo Alto, California, U.S.A., 1996.

[26] Wuensche, A. & Lesser, M. The Global Dynamics of Cellular Automata: An Atlas of Basin Attraction Fields of One-Dimensional Cellular Automata. Addison-Wesley, 1992. (Available as PDF from http://www.ddlab.com)

Adapting the trombone: a suite of electro-acoustic interventions for the piece *Rouse*

Neal Farwell
University of Bristol
Department of Music, Victoria Rooms,
Queens Road, Clifton, Bristol BS8 1SA, UK
+44 (0)117 954 5028
neal.farwell@bristol.ac.uk

ABSTRACT

Three electro-acoustic systems were devised for a new trombone work, *Rouse*. This paper presents the technical systems and outlines their musical context and motivation. The uSlide measures trombone slide-extension by a minimal-hardware ultrasonic technique. An easy calibration procedure maps linear extension to the slide "positions" of the player. The eMouth is a driver that replaces the mouthpiece, with software emulation of trombone tone and algorithmic musical lines, allowing the trombone to appear to play itself. The eMute is built around a loudspeaker unit, driven so that it affects strongly the player's embouchure, allowing fine control of complex beat patterns. eMouth and eMute, under control of the uSlide, set up improvisatory worlds that are part of the composed architecture of *Rouse*.

Keywords

Trombone, electro-acoustic adaptation, mute, composition, improvisation, mapping, illusion, emulation, ultrasonic.

1. INTRODUCTION

Rouse for trombone, trombonist, and live electronics was the fruit of a commission from trombonist extraordinaire Hilary Jeffery. Jeffery's own "tromboscillator" is an improvisation environment controlled by a sensor-equipped trombone mute [1]. The further development of a mute, to be somehow electronically activated, was at the centre of the commission.

The final system has three interlinked components: the *uSlide*, an ultrasonic measure of trombone slide-position that is efficient in implementation and rugged under performance conditions; the *eMouth*, a transducer and excitation system that replaces the player's embouchure at the mouthpiece so that the trombone can seem to play itself; and the *eMute*, a loudspeaker-based device held and operated like a plunger mute, which changes the acoustics and playing characteristics of the trombone. *Rouse* has been performed in concerts of otherwise-acoustic music, and its hardware was used on Jeffery's regular concert instrument. The devices thus had to be easy to fit and remove with no damage to the instrument itself. A further goal was that the system should be low cost, and involve minimal application-specific hardware beyond the three named devices. Software was implemented in Max/MSP.

An obvious precursor is Nic Collins' "trombone-propelled electronics".[1] This uses an old trombone as armature for a new performance instrument, whose soundworld involves sampling external sources and reshaping them, often in collaborative improvisation. By contrast, *Rouse* aims to make the trombone more itself, in a surreal or hyper-real presentation of its acoustic world and playing styles.

This paper has a dual purpose. It will give a technical description of the suite of electro-acoustic interventions, and will also summarize aspects of the musical result with the intention of showing the motivation for the adaptations. The electro-acoustic ideas grew out of compositional goals, and the systems were optimized informally. Technical experiments and developments led to compositional ideas and vice versa. Further information about *Rouse*, sound examples, and screenshots of the operating software have been placed at the Sonic Arts Research Archive [2].

2. TROMBONE ACOUSTICS

The adaptations for *Rouse* engage with aspects of normal trombone acoustics, summarized below. A more detailed description is online at [3]. A number of older books on trombone pedagogy were found useful to relate acoustic principles to musical practice [4], [7], [9], [10], [13].

The vibrating aperture of the player's lips creates an oscillating air pressure. Buzzing on the mouthpiece alone, the frequency is a continuous variable set by embouchure shape and lip tension. The trombone acts as a resonator and radiator. The tube is effectively closed at the player's lips and is designed, through combination of the cylindrical section, mouthpiece and flared bell, to have strong resonances on an overtone series $2f$, $3f$, $4f$, etc. The pressure anti-node at the closed end interacts with the embouchure mechanism, forcing the oscillator onto the nearest preferred frequency. The fundamental f is not at a resonance but can be played as a pedal tone, the oscillator mode-locked by reinforcement of its overtones. The slide, a telescoping tube system, varies the actual length hence effective acoustic length of the trombone. With the slide closed, 1st position, the Bb trombone has notional fundamental Bb1. Each further numbered position extends the tube and lowers the fundamental by a semitone, to E1 at full-extension 7th position. The linear distance between successive positions increases because the tube length must increase in geometric progression. The slide is a precision system and the outer part is made of light-gauge tubing to be low in inertial mass, allowing considerable agility. The pitch range of the instrument is covered by slide-selection

[1] Seen in concert, and described in [5].

of series and embouchure-selection of overtone. The radiation characteristics of the bell vary with frequency. A mute reshapes this frequency dependence.

3. THE uSLIDE

Slide extension is measured by a time-of-flight ultrasound method using a 40KHz transmit-receive ceramic transducer pair. There are many existing implementations of ultrasonic distance measurement. The version here offered particular advantages for the trombone and in conjunction with the other hardware for *Rouse*. Mapped to slide position, the uSlide provides control information to the eMouth and eMute.

3.1 Hardware

The computer audio interface (a Hammerfall DSP Multiface) hosts the ultrasound transducers directly, at a system sample rate of 96kHz. The ultrasound transmitter is driven by the headphone output, and the receiver is wired to an analogue input – there is no hardware amplification or signal conditioning. This further simplifies an approach described by Johannes Taelman [12]. The transducers are mounted on buffers shaped from pencil erasers, and bound to the trombone tubes with double-sided eye-and-hook tape (see Figure 1). This makes a lightweight non-slip and non-abrasive mount, possibly with a shock-decoupling effect, and is very quick to attach and remove from the instrument.

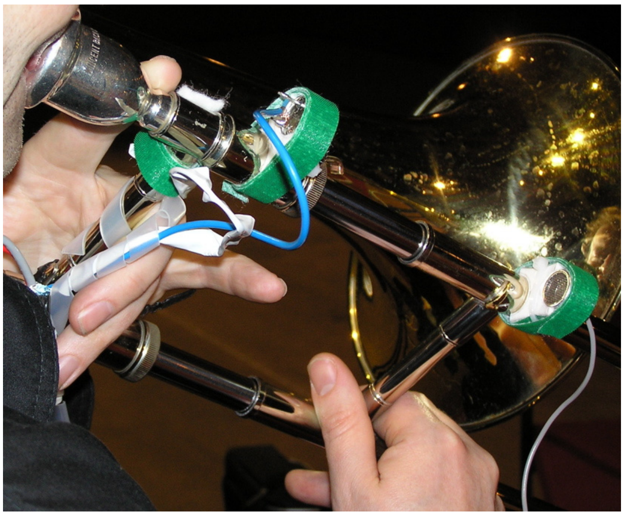


Figure 1. The uSlide transmitter (right of photo) and receiver (upper middle). The player's left forefinger is resting on the mouthpiece, next to the Trombone push-button (hidden). The nylon coil-wrap (centre left) is an easily released tether for the umbilical cable.

Measurements of the trombone spectrum (see below) suggest the trombone can produce significant energy in the 40KHz region. The transducer mounted on the outer slide will pass in front of and behind the bell edge as the slide moves. To minimise interference, the receiver is mounted on the static inner slide, where its acoustic environment is more stable, and the transmitter is fitted to the outer. An umbilical cable connects both signals to the computer, extended by a fly-lead of miniature coax in the case of the moving transmitter. The umbilical is tethered to the cross piece of the inner slide and falls loose in front of the player's body. This proves very workable in practice, the main inconvenience being that the fly-lead sometimes tangles when the instrument is first picked up. Early experiments with a miniature powered satellite, optical in one direction, ultrasonic in the other, were abandoned. Future development may revisit this idea, or explore the mechanically simpler option of a reflector mounted on the slide.

3.2 Drive and Conditioning

The uSlide uses continuous-wave (CW) transmission, with phase inversion acting as the change-of-state to be detected. The ultrasound system runs in a software shell at 96KHz – see [2] for Max/MSP patches.[2] The nominal measurement accuracy is one sample, or 3.5mm, at a refresh rate of 200Hz.

At 96KHz sampling, one cycle of 40KHz occupies 2.4 samples. An exactly repeating pattern of 12 samples generates 5 complete cycles. As far as the audio hardware is concerned, this is an ordinary signal at less than the Nyquist frequency and so is reconstructed by the DAC as a smooth 40kHz sinewave for the transmitter.

Pulsed transmission was rejected for this application. If the pulse amplitude is high, it can be audible as a click or buzz. The drive level available from the interface headphone jack was in any case limited. If the pulse amplitude is smaller, the receiver is susceptible to mis-triggering from air-born interference and from mechanical impact pinging the transducer. Both are likely in the case of the trombone, mechanical noise arising for instance from the proximity of the F-valve trigger. Time-based averaging or reasonableness-measures introduce latency. Simple pulse systems can also suffer from a "cogging" effect, when the detection threshold interacts with the received pulse envelope and distance-dependent amplitude: depending which cycle of the received waveform is registered, the measured distance can show abrupt jumps of around 8mm.[3]

For the uSlide CW implementation, a buffer is pre-filled with cycles of the 12-sample pattern, and played in a continuous loop. The buffer index provides both a trigger for synchronous inversion of the carrier at the loop start, and a measure of elapsed time since that inversion. The loop period is 5ms, more than enough for the <2ms time-of-flight at full slide extension, but short enough for acceptable latency.

The buffer is in fact two-channel, the second channel filled with a quadrature version of the first. These sine and cosine signals (without the periodic inversion) are used for synchronous demodulation of the receiver signal. The cutoff frequency of the filters in the demod chain (8-pole lowpass) can be adjusted to trade off selectivity against response time. The received signal must pass through an amplitude null at the notional phase inversion. The null is detected by thresholding the demodulated amplitude against a fraction (set empirically at 0.7) of its locally-smoothed value. This also implements AGC on the received signal, with controllable parameters.[4] When the null is detected, the buffer index is captured in a sample-and-hold, rescaled as slide position, and passed from the shell to the main software body running at 48kHz.

---

[2] The CW technique here was originated for *Rouse*, and I have not seen other examples in NIME applications. A later search suggests that related techniques are documented in the engineering and instrumentation literatures.

[3] $340ms^{-1} \times 1/(40kHz)$. I encountered this effect during work on my "funny fiddle" (1996), described in [8].

[4] Continuous-wave reception also allows (though not used in *Rouse*) the continuous direct measurement of movement velocity via Doppler shift. For the "funny fiddle", a self-zeroing digital hardware implementation was devised for bow velocity [8]. For Doppler sonar, see also [12].

In use, the AGC seems to prevent the cogging phenomenon, and the system copes readily with the unamplified transducer signal some tens of dB below full-scale. Resilience to air-borne and mechanical interference has not been systematically tested, but the uSlide has so far proved reliable in concert use.

3.3 Scaling

The ultrasonic system generates a raw linear measure of slide extension. For most purposes in *Rouse*, a measure of slide position – the musical measure natural to the player – is needed, which at integer values corresponds one-to-one to a chromatic scale of pitches identifying the overtone series. If the ultrasonic measure is x and the effective length of the trombone tube is L, then $L = L_0 + kx$ for some L_0 and k. The notional fundamental frequency $F_x = cs/L$, where s is the speed of sound in the tube, and c depends on the geometry.

For the player, it is much easier to deal in pitches than frequencies. For calibration, we select two pitches in a register that is easy to play accurately, both on the third overtone. Using the MIDI-note ←→ frequency conversion functions in Max and simplifying fractions:

$$\text{note\#} = \text{ftom}\ (F_x) = \text{ftom}\ (\ 1 / (a + bx))$$

Once the uSlide sensors are fitted, and the acoustic instrument tuned and warmed up, we play F3 (1st position, note# =53) and C3 (6th position, note# =48), register the two x values and solve the pair of equations for the variables a and b. A simple user interface hides the math. The slide position, floating-point for continuously-variable control, is then

$$POS_x = 54 - \text{ftom}\ (\ 1 / (a + bx))$$

A higher order function was considered, but this was found to be as accurate as needed. The transducers need not be precisely positioned when fitted, because variations are calibrated out.

4. THE eMOUTH

For the opening of *Rouse*, I wanted the trombone to appear to play itself. The illusion should be strong, yet also be easy to blur, and the trombonist would gradually be allowed to participate in the trombone's dream. The eMouth system provides an acoustic drive to the trombone in place of the usual mouthpiece, with constrained control from the uSlide.

4.1 Driver

In his "trombone-propelled electronics", Collins used a compression driver of the kind designed for use with an HF radiator horn in loudspeaker arrays. I concluded that available models were too bulky and massive for *Rouse*. With the eMouth fitted, it had to be possible to handle the trombone as normal; and the physical loading could not risk damage to the mouthpiece socket. I wanted also to use a wide audio spectrum.

Jeffery provided a typical mouthpiece that fitted his instrument, of 26mm inner diameter at the embouchure opening. The driver selected is a small-diameter metal diaphragm, rubber surround type. This type allows a long-throw pistonic movement of the diaphragm, producing a high sound level for the small size. Despite their widespread use inside computers and in miniature satellite loudspeakers, these drivers seemed to be unavailable for one-off purchase in the UK. I sought a model closely matching the mouthpiece in diameter: for the eMute prototype, this was a 24mm-diaphragm device cannibalized from an Apple computer. The driver is mounted to the mouthpiece using a plastic gasket, and the assembly held together by a web of cable ties. This is readily adjustable, and easy to dismantle for transport or storage. See Figure 2 and Figure 3. The connection must be airtight and somewhat rigid for all the sound to appear to come through the instrument, since this might not otherwise be the path of least resistance.

This arrangement places the diaphragm of the driver as close as possible to where the player's lips would normally be, and with minimal alteration of the mouthpiece enclosed volume. This gives a close match to the instrument's normal closed-tube acoustics, so that the tuning of the slide positions feels normal to the player and the transducer approximates the pressure drive of the buzzed lips. The transducer is powered by one channel of a small hi-fi amplifier.

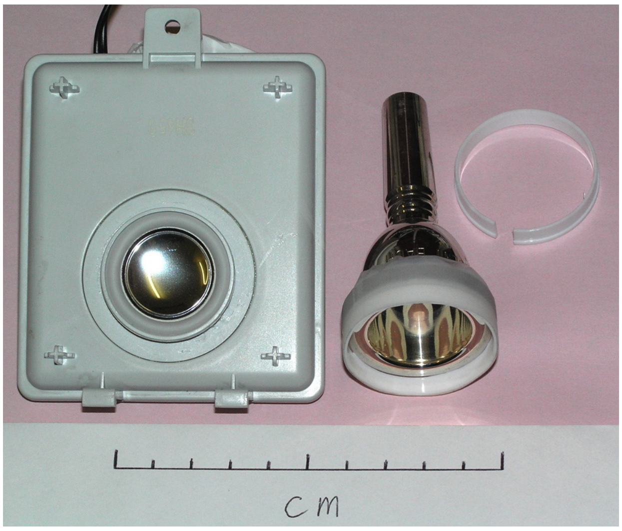


Figure 2. eMouth components. The plastic enclosure has a small port on the rear, which is here sealed closed.

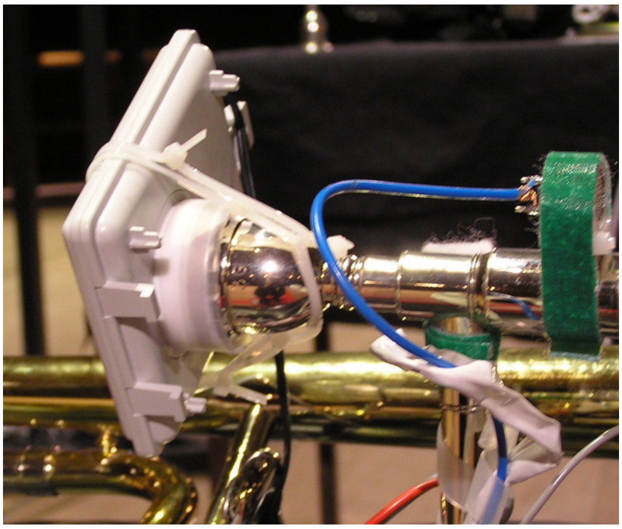


Figure 3. The eMouth assembled and fitted to the trombone.

4.2 Excitation Function

The excitation signal for the eMouth transducer is from a simple synthetic model of the trombone. Test recordings were made of the trombone played normally. An omnidirectional microphone was placed on-axis in front of the bell at around 50cm, in a dry acoustic. Spectral analysis of the recordings yielded some heuristic conclusions:

1. At a given pitch and sustained dynamic, there is an approximately straight-line relationship between partial-amplitude in dB and frequency on a linear scale.
2. The slope varies with dynamic level, flatter at *forte* hence more energy in high partials, steeper at *piano* with the high partials lost below the measurement floor.
3. The overall spectrum is subtly modulated by a formant shape, and the fundamental is typically a few dB weaker than the 2^{nd} or 3^{rd} harmonics. The fundamental, however varies relatively little with playing dynamic.
4. Short notes confirm this behaviour, with the spectral envelope tilting from steep to flat and back during the attack and release of the note.

A surprisingly simple model approximates the measured spectrum. An anti-aliased sawtooth passes through a two pole low-pass filter, a state-variable implementation at low Q. A further high-pass filter rolls off the lowest partials (and protects the driver from exaggerated cone excursions). The interaction of the 1/f amplitude-spectrum of the sawtooth and the LPF slope approximates the measured variable dB/frequency slope by varying the filter cutoff relative to the oscillator frequency.

In steady state, the timbre is plausible for the trombone, but very clearly synthetic. A much more "real" effect, reflecting the common psychoacoustic observation that instrumental timbres are characterised strongly by their transients, is gained by modulating the LPF cutoff with an envelope, linear in frequency/time but matching the timing of staccato notes. Figure 4 shows a test version of the model.

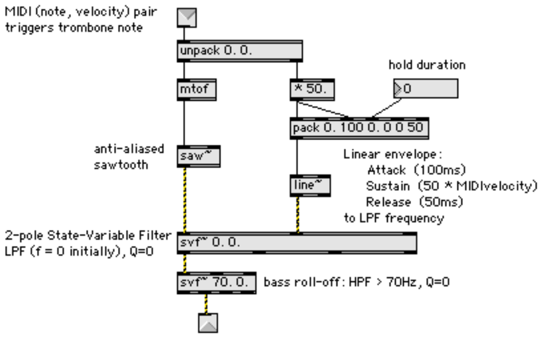


Figure 4. Simple trombone model.

5. THE eMOUTH IN CONTEXT

5.1 Illusion and Transformation

When the eMouth is fitted to the trombone body and driven with passages of notes synthesized by the basic trombone model, the effect is disconcerting. The resonances of the physical instrument overlay and enrich the synthetic timbre, and all the sound is heard to come from the bell. The instrument seems indeed to play itself. However, the sound is "not quite right". In part this is because the dynamic level available from the eMute is quieter than would be expected for the heard tone-colour – I was not prepared to destruct-test the transducer to establish a maximum level(!) – and the frequency of the eMute drive is not precisely matched to the acoustic response of the resonant system.[5]

[5] I made early experiments in matching the eMute pitch automatically to the actual trombone response; but I chose instead to use defined drive pitches, enjoying the timbral inflection created by uncorrelated movement of the slide.

The ambiguity and incompleteness of the illusion was retained because it was what I sought compositionally: an image of the not-quite trombone. To manipulate further the illusion, a "noisy trombone" extension was made to the basic model. Pink noise is fed through a comb resonator, implemented as a bank of individual resonators, whose Q can be varied to give a tone from nearly-unpitched to strongly focused in pitch, with a spectral envelope close to that of the sawtooth. When the tone is most focused, the noise source is crossfaded to the pure sawtooth. Parameters are mapped to give subjectively smooth transformation from noisy to pure in a single control variable.

Table 1. The five phases of "Dream".

P	Mapping of slide position to pitch behaviour.	Automation
1	Slide locked in 1^{st} position. *Trombone solo.* *Trombonist joins for Phase 2.*	Emergence from noise to resonance to trombone-tone. Bb horn-call figures, initially three low notes, expanding to five. The tone returns gradually to noise.
2	Fundamental moves in discrete semitone steps. New pitch takes effect at the start of the next note to be sounded.	Tone re-focuses. Range gradually expands to seven notes of the series.
3	Fundamental moves in discrete semitone steps, but pitch changes straight away, in the middle of any note already sounding.	Tone abruptly changes to noise. Tempo becomes more brisk.
4	Pitch-control becomes continuous: slide to any fundamental in the Bb – E range, and not just tempered semitones.	Noise becomes abruptly darker, then re-forms into trombone-tone.
5	Tessitura also responds to the slide position. If the slide spends more time in positions 1–3, the tessitura drifts towards upper overtones. If the slide spends more time in positions 4–6 (–7), the tessitura drifts down towards the pedal range. The **Trombone** button has an effect: while pressed, the pitch moves rapidly down towards the pedals; only shorter notes are played; and there are no rests in the eMouth phrases. Intended for the end of Phase 5, but may be used earlier if desired.	Note-speed accelerates. A continuous rushing noise gradually fades in.

5.2 Music and Mappings

The opening section of *Rouse* is called "Dream" and has five phases. At the start, the trombone is set on stage, bell facing the audience, and the player sits unobtrusively upstage. The player takes up the instrument for Phase 2. Throughout, the "note" information to the eMouth is an algorithmically generated cadenza exploring horn-call figures. The trombone automatically plays notes up and down an overtone series (including the pedal tone), and the uSlide determines which series is used. For slide positions 1 – 7, the fundamental is Bb – E (descending), just as in normal playing. Position 7 also acts as trigger for the transition to the next dream-phase. The visual component of reaching full extension may be theatricalized or underplayed. The effective resolution of the uSlide control evolves through successive phases, as do the note-speed and range of the algorithmic cadenza, and trajectories automated between noisy and pure tone. Table 1 summarizes the behaviours. The score guides the trombonist through "an improvisation within the strange dreamworld of the not-quite-trombone", moving from calm to frenetic.

5.3 Player and Audience

The sensory feedback loop from the trombonist's action – the uSlide selection of pitch-series – is incompletely closed because a heard pitch may arise from different series. The player can draw on practised knowledge, (pre-)setting the slide intuitively for a desired pitch-set. However, the controlled-random detail of pitch and rhythm, the progression of the slide mapping from under- to over-determined, and the programmed trajectories in texture, tend to undermine this security. The player has to balance the attempt at fine control with the management of larger-scale momentum. For the audience, the fact of the player's control is apparent, but its limits and frustrations are also suggested. The ambiguity and the partial redundancy of the player help set the dramatic intent of the dreamworld.

6. THE eMUTE

My initial idea for a new mute was to create a device with electronically-variable frequency dependence in its muting effect. Experiments with small loudspeakers in the trombone bell led to a different discovery, which became the core of the eMute. The eMute is a hand-held device build around a loudspeaker driver and fed by a variant of the simple synthetic trombone model. Because the driver has no housing, the sound heard from the mute is quiet compared to that from the trombone yet, with the trombone acting in reverse as an "ear trumpet", exerts a strong effect on the player's embouchure. If the played and eMute pitches differ, strong pulsation is heard in the played tone corresponding to the beat frequency, and this beat is closely felt through the embouchure.

Embouchure tracking has been investigated as a means of electronic control [6], and breath control has appeared in several commercial products. With the eMute, the effect is rather reversed. The embouchure provides haptic sensitivity to the electroacoustic intervention, while continuing its normal purpose of acoustic control. This facilitates, with a nearly normal playing technique, fine control of the beat-interval from unison through microtonal to large. Coupled with a quasi-intuitive manipulation of the eMute pitch, this proved a rich space for contemplative improvisation; and, with the eMute driven programmatically, for scored materials.

6.1 Hardware

A large driver is preferred for a strong acoustic effect, and the driver must also withstand induced vibrations from the blown trombone. However, the assembly must be light enough to be held in the hand and operated like a plunger mute, without fatigue, because this player-adjustment is essential to the improvisatory control. The driver model chosen for the prototype is a mid-range/woofer with a 12cm diaphragm and compact magnet assembly. The paper diaphragm is not ideal, because water droplets can be propelled out of the trombone bell; but it has so far withstood extended sessions without harm. A glued surround cut from 5mm foamboard matches the outer diameter (18cm) to that of a metal plunger mute favoured by Jeffery, and provides non-abrasive and circular contact with the instrument bell. The assembly balances in the hand, and a push switch (labelled $\boxed{\text{eMute}}$) is mounted at a convenient position for the player's fingers. A tethered cable connects the driver to the spare channel of the hi-fi amplifier, and the switch to a hacked USB game controller serving as computer interface. The eMute prototype is shown in Figure 5 and Figure 6.

Figure 5. eMute. The white foamboard surround has a circular cutout matching the inner diameter of the frame.

Figure 6. eMute in use.

6.2 Control mapping

Real-time change of the eMute pitch is by manipulation of slide position and the $\boxed{\text{eMute}}$ button, using a click-drag metaphor modelled on the computer mouse. Table 2 summarizes the actions. eMute pitch changes, though quiet, are sometimes audible to the audience, but this is absorbed into the improvisation.

6.3 An eMute context: "respiration"

The second section of *Rouse* is called "Deep Sleep", an extended improvisation with the eMute. For the first time, sound is heard from two playback monitors, placed discreetly in the near stage-space of the player. Within the imagined world of the piece, these create a "respiration field" extending the body. In this section, the monitors quietly play a shifting overtone field related to the eMute pitch.

A synthesis engine periodically triggers overlapping sine-tone voices each with a slow attack-release envelope. The frequency of each tone is a controlled-random exact multiple of the current eMute frequency, whether or not the eMute is currently sounding. At the start, the eMute pitch is preset but the device itself is silent. Overtones sound while the player changes from eMouth to mouthpiece and takes up the eMute. As the eMute pitch subsequently changes, new overtone triggers relate always to the current pitch. The result is a disembodied sound-bed with no abrupt transitions. Yet the overtone field always suggests, probably unconsciously for the audience,[6] the eMute pitch and hence anchors the blown pitch: there is a “rightness” to the played pitches.

There are three further influences on the overtone field. Patterns of rapid slide movement (whether or not they are changing the eMute pitch, and whether or not the instrument or eMute are sounding) accelerate the synthesis retriggering to produce flurries in the overtones. Over time, the tessitura of the field migrates. Lastly, once the trombone begins playing, a small random frequency deviation is added to each sine voice, less than or equal to a maximum that gradually increases from zero. This gradually increases the likelihood and intensity of beats perceived in the respiration field, a mirror of those created by the trombone and eMute.

Table 2. eMute control mappings.

single-click
Toggle the eMute on/off (i.e. sounding / non-sounding). *If the eMute is “off”, the actions below still change the (silent) pitch. If “on”, the change is heard immediately.*
single-click-drag
Slide moves the eMute pitch up/down the present overtone series (as performed by *embouchure* in normal playing). Motion is relative to the initial pitch when eMute is pressed. ***Example:*** *the eMute pitch is currently in the pedal range. Move the slide to 7th position; click and hold eMute while bringing the slide into 1st position; release eMute. The eMute pitch will ripple up to the 7th harmonic. To raise it further, repeat (go to 7th, click-drag upwards, release).*
double-click-drag
Change the fundamental pitch for the eMute, according to the slide position when the second click on eMute is released. Slide positions 1 – 7 give fundamentals Bb – E, “as normal”. *A quick double-click gives near-immediate pitch change.* If the button is held – *click-release-press-hold-release* – the pitch makes a glissando. The duration of the gliss is equal to the *hold* time, and begins at the final button release.
You can play normally at the same time as changing the eMute pitch.

7. CONCLUSION AND FUTURE PLANS

The systems described above were developed for a particular composition, *Rouse*. Control mappings here have an important role in specifying the musical intent and result, by defining an improvisatory space. Choices within this space are further conditioned by composed indications in the score. The network of interactions in the trombone-trombonist-electronics system involves various different pathways. The electro-acoustic interventions set up the dramatic shape of *Rouse*, which moves from a disembodied, electronically mediated and improvisatory starting point in “Dream” and “Deep Sleep” to an acoustic and fully-scored ending via (not described here) three further sections, “Half World”, “Snooze”, and “Rouse”. I welcome enquiries from interested performers!

The electro-acoustic systems have the potential for other applications and for evolution. The uSlide system offers another option in the range of techniques for ultrasonic distance measurement. The eMouth may be strongly bound to *Rouse* by its visual theatre, but future plans include a higher sound level, matching of excitation to acoustic-system spectrum, and many possibilities for emulation and dissolution of the trombone tone; hence, perhaps, setting new allusive contexts. The eMute/uSlide system can be made self-standing as a tool purely for improvisation. A line of exploration for the eMute is to use the transducer simultaneously as driver and as microphone, to investigate muting and synthesis behaviours steered by the trombone’s acoustic output. eMouth and eMute may be readily adapted to other brass instruments, perhaps in ensemble.

8. ACKNOWLEDGMENTS

My thanks to Hilary Jeffery for the invitation to make this piece, for supplying the mouthpiece used for the prototype eMouth, and for his patience, skill and musical insights in our experiments and rehearsal.

9. REFERENCES

[1] http://www.mti.dmu.ac.uk/~jrich/kreepa/tromb.html

[2] http://www.sara.uea.ac.uk/?work&id=2450

[3] http://www.phys.unsw.edu.au/~jw/brassacoustics.html

[4] Bate, P. *The trumpet and trombone : an outline of their history, development and construction*, 2nd ed. E. Benn, London, 1978.

[5] Collins, N. Low Brass: The Evolution of Trombone-Propelled Electronics. In *Leonardo Music Journal*, 1 (1991), 41-44.

[6] da Silva, A., Wanderley, M., and Scavone, G. On the Use of Flute Air Jet as A Musical Control Variable. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, BC, Canada, 105-108.

[7] Dempster, S. *The modern trombone: a definition of its idioms*. UC Press, Berkeley/London, 1979.

[8] Farwell, N. J. *being what I am, doing what I do: manifesto of a composer*. PhD Thesis, University of East Anglia, Norwich, UK, 2001, 131-133.

[9] Gregory, R. *The Trombone. The Instrument and its Music.* Faber, London, 1973.

[10] Kleinhammer, E. *The Art of Trombone Playing*. Summy-Birchard, Evanston, IL, 1963.

[11] Pierce, J. Introduction to Pitch Perception. In Cook, P.R. *Music, Cognition, and Computerized Sound*, MIT Press, MA, 2001, 58-63.

[12] Taelman, J. A low-cost sonar for unobtrusive man-machine interfacing. In *Proceedings of the 2002 International Conference on New Interfaces for Musical Expression (NIME02)*, Dublin, 92-95.

[13] Wick, D. *Trombone Technique*, 2nd ed. OUP, Oxford, 1984.

[6] The perceptual mechanism infers a fundamental pitch [11], though the effect is subtle here.

The Augmented Djembe Drum – Sculpting Rhythms

Teemu Maki-Patola
M.Sc, researcher
Laboratory of Telecommunications Software and Multimedia, Helsinki University of Technology
+358 9 451 5849
tmakipat@tml.hut.fi

Perttu Hämäläinen
M.Sc, M.A., researcher
Laboratory of Telecommunications Software and Multimedia, Helsinki University of Technology
+358 50 596 7735
pjhamala@tml.hut.fi

Aki Kanerva
Research assistant
Laboratory of Telecommunications Software and Multimedia, Helsinki University of Technology
+358 50 544 8673
aki.kanerva@iki.fi

ABSTRACT

In this paper, we present an augmented djembe drum created by mounting a webcam inside a regular djembe. By moving your hands in the vicinity of the drum membrane, you can sculpt computer-generated rhythm patterns that imitate real-life djembe rhythms. You can also play the djembe for real, treating the patterns as interactive accompaniment. A computer vision system infers the 3d location of the performer's hands by tracking their shadows on the drum membrane. The individual drum hits of the patterns are automatic, but the player has direct control over their loudness, tempo and timbre. We explain the design and implementation of the instrument and share our design experiences. We also present qualitative results from testing the instrument with amateur musicians and experienced drummers.

Keywords

Augmented musical instrument, computer vision, drum, tactile feedback.

1. BACKGROUND

Gestural interfaces for sound control have been created since the Theremin in 1919 [21]. Their popularity has increased much more in the past two decades, thanks to quickly evolving technology.

Computer vision technology based on web cameras and inexpensive sensors (e.g. Phidgets [17]) greatly reduce the effort needed to use gestures as an input medium. Complementing input devices, several software tools, such as EyesWeb, Pure Data and Open Sound Control, [2], [14], [19], are available to help with creating interface prototypes rapidly. These tools handle functions from extracting control features from a webcam feed to routing the data to sound models, and also hosting the sound models themselves. However, moving from prototype to final product cannot often be done with these tools, as will be discussed later.

Thanks to increase in both processing power and research attention, computer vision is becoming more popular as an interaction method. One of its advantages is the lack of any wires or control devices, allowing the performer freedom of expression. Computer vision may be used for shape and feature recognition, and it offers higher precision in tasks such as evaluating color, object proportions, wavelengths or features that humans do not perceive.

On the topic of practical computer vision for user interfaces, there is a growing body of research [4], [5], [6], [11]. Practical systems often make simplifying assumptions, for example, that the user performs or at least tries to perform only motions that are relevant in a given application context [4], [5]. Markers, such as coloured gloves, can also allow the user to be tracked as 2d colour blobs instead of a fully articulated 3d skeleton [9], [13].

An article by Paradiso [15] and a book edited by Wanderley and Battier [23] offer a good introduction to existing gestural musical interfaces. Many prototypes have been created, and a few commercial gestural controllers exist as well [7], [22].

Computer vision or magnetic tracking is also used in some tangible interfaces to locate specific objects and buttons directly from a control surface [8], [16]. The instrument installations Music Table [20] and Augmented Groove [18] present controls layered on top of real objects, such as cards or LP records, visible on screen. Many traditional instruments have been expanded with additional sensors [15], [23]. However, the added control is often used to alter the existing sound through filters or effects. We have augmented a real musical instrument by adding a completely new way to play. The player can play the drum normally, use only the augmented playing method, or even play both versions at the same time.

The augmented djembe is different from our previous instruments based on computer vision [13]. The control features are not extracted directly from an image of an object or body part, but from lighting changes caused by body parts. The physical drum itself remains unaltered, because the technology is hidden.

One of our goals in building new instruments is to make the experience of playing available for a larger audience by reducing the amount of practice needed to play instruments. The augmented djembe follows this ideal as well, because the augmented playing method can be used even by beginners to create advanced results. The amount of control can then be gradually increased, such as hitting the drum every now and then, all the while staying in rhythm. This way, the augmented playing method supports the user's learning and makes practicing more rewarding.

2. AUGMENTED PLAYING METHOD

The player of an augmented djembe moves his hands above and on the surface of the physical drum, without striking the drum membrane. A rhythm pattern consisting of sampled djembe hits is automatically generated and output through a loudspeaker, and the player's hand motions control the timbre of the individual strikes in the pattern. This method was devised by

NIME 06, June 4-8, 2006, Paris, France.

observing real-life djembe players. In this article, we refer to the non-augmented djembe as the 'real-life version'.

2.1 Real-life djembe

When playing the real-life djembe, the player strikes the drum alternately with left and right hands at a steady tempo and rhythm. Larger-scale dynamics are "sculpted" by modifying the loudness and timbre of the individual hits. Loudness is modified simply with strike strength, and hits can also be skipped, which we regard as changing the loudness to zero.

In addition to striking the drum membrane, the player controls the timbre of hits by striking at different locations. Striking near the centre produces a deep and dry sound, and a location near the edge results in a higher and more tonal and reverberant sound. The player uses this control to produce gradual changes in timbre over a fast pattern of strikes.

2.2 From real-life to augmented playing

Figure 1. Recording setup for analyzing hand motion during real-life djembe playing. To record hand motion, a magnetic motion sensor is attached to each hand.

We hypothesized that the djembe playing hand motions could be separated into components of different frequencies. The high frequency consists of the up-down motion of striking the drum, and the timbre and loudness modifications are seen as lower frequencies. If the high frequency was eliminated by automatically producing a pattern of strikes, the low frequencies could still be used to control the timbre and dynamics of the strikes in the pattern.

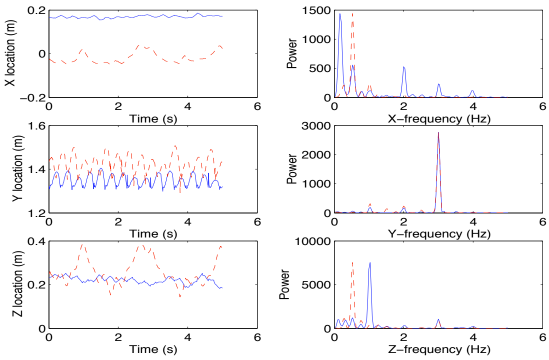


Figure 2. Motion of player hands along coordinate axes (left) and their respective frequency spectrums (right). The dashed line represents the right hand. The left side shows samples of movement data. Frequency spectra are evaluated from more comprehensive data. For example, the y-spectrum reveals that the drum is struck with alternating hands at a rate of 3 Hz or 180 bpm.

To prove this hypothesis, we recorded the motion of a djembe player's hands using a Polhemus magnetic tracker (see Figure 1). The location data was then analyzed in MatLab to extract separate movement trends.

By plotting the frequency spectrum of hand motion along coordinate axes (see Figure 2), we find out the different frequency components. Plotting vertical hand motion reveals that the membrane is struck at a rate of 3 Hz, visualized in the second row of Figure 2. The lower frequency components of dynamics in the recorded djembe pattern are 1 Hz for the left hand and 0.5 Hz for the right on the z axis, and the same 0.5 Hz for the right hand, and a 2 Hz pattern for the left on x axis, as illustrated in Figure 2. Combined, these patterns define the movement between centre and edge, which causes the most radical changes in the timbre of the sound.

3. IMPLEMENTATION

Figure 3. A web camera is mounted inside the physical djembe, facing towards the membrane. The software runs on a Windows laptop. Good speakers are also needed as the sound contains mostly low frequencies.

The augmented djembe consists of two parts: the physical djembe drum with a webcam mounted inside, and computer software. Both the sound model and the computer vision run inside the same software, a custom Windows application.

3.1 Hardware

The large viewing angle (70 degrees) of the Creative NX Ultra webcam allowed us to mount it at the narrow base of the drum, where it could be fixed securely, while still capturing the entire membrane in its field of view. The software runs on a regular Windows computer, and poses no special requirements. However, in minimizing audio latency, it is recommended to install either ASIO drivers supported by the soundcard, or the generic ASIO4All drivers [1]. Strong ambient lighting or a lamp positioned perpendicular to the drum membrane, pointing at the drum, is required.

3.2 Software

The initial prototype was constructed on the PureData [19] software, which was used for both input mapping and sound generation. Communication with the computer vision application was done through Open Sound Control [14] messages. Soon, the input mapping was moved to the custom computer vision application, leaving only the sound production for PureData. It then became apparent that a custom sampler would serve us better for further prototyping, and we abandoned the Pure Data platform.

3.3 Image analysis

Figure 4. Raw images from the web camera. The hands project blurred shadows on the surface of the drum membrane.

Figure 4 shows what the camera sees during playing, when a lamp causes hands to cast shadows on the drum membrane. The real-life djembe controls for timbre and loudness – strike position and strength – are mapped from hand location and height as seen by the camera. Based on the distance of a hand from the edge to the centre of the drum, the software selects a sample recorded from a similar position to play back. Conversely, the height of a hand is used to control the volume of the sample.

Despite this poor image data, it is possible to extract the required information in three steps.

3.3.1 Circle fitting

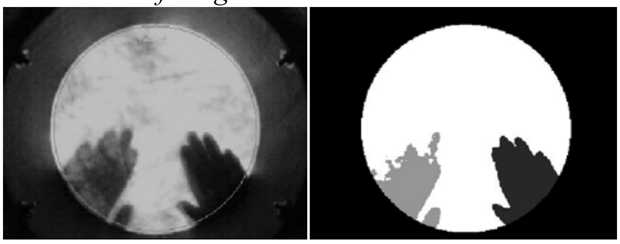


Figure 5. Pixel mask generation by fitting a circle constraining the lit area of the membrane. Note how the left hand's shadow is brighter due to the hand's distance from the surface.

First, a pixel mask is created to include only pixels inside the membrane area for further processing. This should be done every frame, because the camera is not absolutely fixed, and the djembe may move during playing. This has the added benefit of ignoring artifacts during an initialization phase, and being more tolerant of lighting changes during playing.

The translucent membrane receives more light than the inside of the drum body, so its pixels are brighter than those of the body. The membrane's pixel mask is created by fitting a circle to contain the bright pixels. We use genetic optimization to maximize a fitness function that has the radius and centre of the circle as parameters. This works well even when hand shadows are present in the image. The fitness function is:

$$f(r,x,y) = r\sum_{i=1}^{N}\left[I(x + r\cos\alpha_i, y + r\sin\alpha_i) - I(x + r'\cos\alpha_i, y + r'\sin\alpha_i)\right]$$

where r'=1.05r, I(x,y) is the intensity of the image pixel at coordinates x,y, and $\alpha_i = i2\pi/N$. Basically, fitness equals the intensity of the circle with radius r minus the intensity of a slightly larger circle with radius r'. This is maximized when the first circle is aligned at the rim of the drum. The sum of the intensities is multiplied by r to favour large circles, which speeds up the convergence of the optimization. According to our experience, it is sufficient to sample the circles at 16 points, that is, $N=16$.

3.3.2 Thresholding the membrane

The membrane of a typical djembe is not uniform in color. To remove the pixels that are not in shadow we simply threshold away the pixels that are brighter than the color variations of the membrane. As a result of the variations the final hands may become "broken", as seen in Figure 5.

3.3.3 Shadow area and average color

After thresholding, only hand shadows remain visible for further processing. The surface is divided into two halves, and for each half the amount of overlaying shadow surface and its average colour is evaluated.

Initially, we had used an angle histogram to determine whether zero, one or two hands were visible. The histogram calculated the amount of shadow mass as a function of angle from the center of the membrane. A valley in the histogram would indicate the space between hands. This approach was abandoned, because it was prone to errors when playing with spread fingers, which made it difficult to interpret clusters properly.

3.4 Sound model

The sounds of the augmented djembe were sampled from the same drum, and are played back using a software sampler. The sample matrix consists of strikes at varying distances from the membrane centre, with varying intensities. For each location and intensity pair, a number of different samples were recorded to avoid sounding too mechanical.

To further reduce monotony, each output sound consists of multiple samples recorded from nearby locations, with continuously changing weights. As a result, the sound model can produce a nearly infinite amount of different combinations even with a small selection of samples.

3.5 Control mapping

Strikes are generated for both hands at a steady tempo, which can be changed with the software's interface. Each hand strikes 8th notes alternately, resulting in a straight 16th-note pattern. To avoid the mechanical quality of strict tempo, the strike times vary randomly within 5% of the actual beat [3]. This simulates a human player, and sounds livelier.

The distance of a hand to the membrane's centre is mapped as weight in choosing samples to mix together from the sample matrix, effectively modifying the timbre.

The height of a hand (distance from membrane) is calculated from the average luminosity of the visible shadow on each membrane half. The higher the hand is, the brighter the shadow becomes, creating a penumbra. For this reason, strong ambient lighting or a lamp with dimensions, such as a fluorescent lamp, is required.

Thus, the locations of the player's hands control both the timbre and loudness in the rhythm pattern. For producing typical djembe rhythms, the user moves hands in smooth, rhythmic motions over the drum's surface. This motion resembles the low frequency motion of real-life djembe playing excluding strikes.

3.6 Visual feedback

Both the unprocessed camera image and the result of computer vision processing are shown on screen. This feedback helps the user understand how the sound is generated and sculpted.

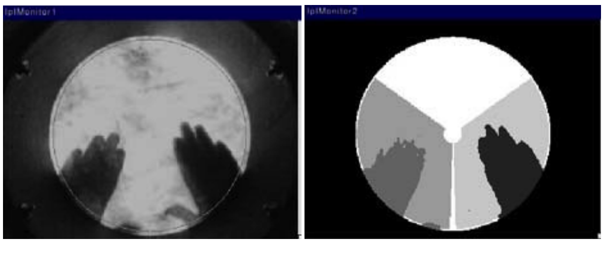


Figure 6. Visualization. Distance from centre is represented by darkening the according sector, and hand height visualized by darkening the hand itself.

4. USER TESTS

We conducted user tests to find out how people with different musical backgrounds experience the instrument, and to see what playing styles would be discovered. The informal test situation was a jam session where each user was allowed to experiment with the various features of the instrument. All comments given during test sessions were recorded.

4.1 Users

The test users consisted of six people. One was an experienced percussionist, three had a strong musical background and two were amateur musicians. Only the percussionist had previous experience of playing a djembe. One subject had experience with other hand drums, while the rest had none. Five of the subjects were male. The ages of the subjects were between 24 and 46 years.

4.2 Procedure

First, the real-life djembe was introduced to the user with a three-minute introduction video. The video taught how to hold the instrument properly, demonstrated playing and explained timbre and dynamics controls. After the video, the user was encouraged to experiment with the djembe.

Next, the augmented version was introduced by explaining its control logic with the test conductor showing simple demonstrations. The user was told that they could ask to change the tempo.

The user was then allowed to experiment with the augmented djembe for as long as they liked, which ranged with different users from half an hour to two hours. Users were encouraged to comment during the entire test.

Finally, the user was interviewed with informal questions:

1. How would you describe the playing experience?
2. What kind of playing styles did you find?
3. Were you able to achieve satisfying results?
4. Did you feel that you were in control of the instrument?
5. How did it feel to play together with the augmented djembe?
6. What would you change in the instrument?

4.3 Results

Most users were intrigued by the augmented instrument. They felt it was a good idea to have the augmented playing method attached to a real-life instrument: "Because the user interface is an actual drum, playing feels natural, like playing a real instrument".

The users quickly learned to produce patterns with little or no dynamic variations, but maintaining a complex pattern for a longer period of time turned out to be more difficult than expected. According to some users, a partial reason for this was that they did not know much about typical real-life djembe patterns. We assume they also had difficulty in knowing how the hands should be moved to produce a pattern they had heard.

It took some time for each user to find a suitable playing style. Most subjects felt this was because the control approach was fundamentally different than their mental model of playing a drum, even if it was based on drum playing. One user commented that the instrument was easier to understand once you stopped thinking of it as a drum. Still, after some practice, most subjects found ways to produce rhythms they enjoyed.

4.3.1 Playing experience

Most users felt that the instrument felt natural and behaved logically. Many noticed some latency, but felt they could adapt to it. Some users commented even without being asked that the augmented instrument was easier to approach than the real-life djembe. These users enjoyed the ability to produce decent-sounding results even with little practice.

The user with the most musical experience commented that combining real-life djembe playing with the augmented method was the most fruitful in a musical sense. He quickly learned to use the augmented patterns as a dynamic background for his own djembe playing. He felt the steady patterns allowed him to practice real-life playing, and he was able to play more freely and use embellishments when the augmented patterns took care of staying in tempo. Nevertheless, the dynamics of the patterns could be controlled enough to be musically interesting instead of acting as a monotonic metronome.

The experienced percussionist felt that the augmented play mode was frustrating when he already knew how to play the instrument. It appears that a more fruitful approach for him would have been to use the computer vision to control the parameters of effects applied to the sound of the real djembe.

4.3.2 Playing styles

Figure 7. Small motions produce drastic changes in the sound. Lifting the hand only a few centimeters halves the amplitude.

For maintaining a particular rhythmic pattern, the most effective way to play was to keep one's hands on the drum's surface. It provided an anchor for repetitive movements, and made it easy to stay in tempo. Several users felt that without touching the drum, it was difficult to repeat motions accurately.

The most successful playing styles involved rocking the whole body with the rhythm. This way, the user was immersed in the rhythm of the pattern, which was additionally helped by closing the eyes and concentrating on the aural feedback of the drum's sound. One user described the experience as being "submerged in a flow of mind."

Figure 8. Touching the drum's surface helps to repeat motions accurately. Here, the palm of the left hand stays in contact while its angle affects the size and intensity of the shadow. The right hand moves with a larger motion, controlling amplitude.

4.3.3 Problems

Users commented that the biggest problem was dropping out of rhythm without making a clear error. This seems to be the combined result of the nature of rhythm patterns and latency. The sound of a strike is based on what the camera sees at the time, which may be data that is over 50 ms old due to the camera's 30 Hz refresh rate and the delay in starting the next sound. As a result, the player needs to adapt to performing controls slightly before the sound is heard.

However, the user has little feedback of the phase of his motion processing. For example, while the user keeps on emphasizing the first hit in a pattern, his control action may happen in a good time or be almost late. Both situations appear similar to him. If his action is almost late, his time precision may fluctuate just enough to cause the control action to suddenly happen too late missing the next hit. As a result the pattern is shifted, which confuses the user who did not perceive making any errors. Some users were able to adapt to this once they understood what caused the shifting: "you try to adapt your playing to the sudden error but until you understand why it happens, fixing it may be difficult."

4.3.4 Suggestions for changes

A simple way to reduce the shift error described above would be to use a camera with faster refresh rate. We considered adapting the tempo to the player dynamically, but concluded that this would lead to both user and software adapting to each other, resulting in even more confusion.

Many subjects suggested using presets for different rhythmic patterns, as well as different sounds to make the playing less monotonic. Presets could be selected with an attached switchboard, for example. Additionally, users wanted a way to modify the tempo quickly and easily.

5. DISCUSSION

The general impression based on user tests was that our approach has potential, even though there are a few technical issues, such as reducing the latency, to tackle. The augmentation's main benefit was in making the playing more rewarding already early on. Users were able to produce interesting results quickly, and gradually increase their interaction with the real-life djembe. This made the instrument easier to approach and supported the learning process by keeping the motivation high. The automated rhythm patterns also acted as examples of real-life djembe playing.

The conducted user test supports our earlier findings [12] of tangible objects and tactile feedback improving the feel of musical interfaces. Users enjoyed having a real, physical drum as the interface.

5.1 Software Tools

Using PureData allowed us to quickly see if our approach would lead to anything. However, as the instrument developed, the software soon became cumbersome for both the input mapping and sample playback. Once the concept was verified, we moved to a custom application.

Our earlier experiences with existing software tools such as PureData are similar – they are invaluable for rapid prototyping but become unwieldy when polishing final versions. Using highly general architectures is bound to make achieving specific goals more difficult than it is with software designed to achieve exactly those goals. Naturally, making everything by yourself requires a strong programming background but if you have it, it may be a faster approach than learning to expand existing software tools for your specific needs.

6. FUTURE WORK

As suggested by the user test subjects, the instrument would benefit from several additional features. First addition should be to add tempo control, realized as three to five presets in the up-most sector of the drum surface. Through using the instruments visual feedback, the upper sector could contain even pull-down menus for choosing different rhythm patterns, sounds and play modes.

7. CONCLUSIONS

We introduced an augmented djembe drum, which features a playing method augmented with computer vision and sample-based sound generation. The augmented method allows users to concentrate on sculpting rhythm pattern timbre and dynamics without concentrating on precise tempo.

In our user tests, we discovered that the augmentation allowed the users to produce musically interesting results with less practice making the augmented djembe easier to approach. Additionally, a novel playing style was discovered, where users played the real-life djembe accompanied by dynamic automated patterns.

The application of computer vision we developed for this instrument suggests that as long as the context is known well beforehand, accurate control data can be extracted even from seemingly poor image data.

8. ACKNOWLEDGMENTS

The authors would like to thank all the users who tested the instrument for their valuable feedback. The research was partly supported by Pythagoras Graduate School funded by the Finnish Ministry of Education and the Academy of Finland.

9. REFERENCES

[1] ASIO4All Universal ASIO Driver, http://www.asio4all.com/ (Link visited Jan 2006).

[2] EyesWeb software home page, http://www.eyesweb.org/ (visited 1.2006)

[3] Dahl, S. The Playing of an Accent – Preliminary Observations from Temporal and Kinematic Analyses of Percussionists. *Journal of New Music Research* No. 3: 225-233, 2000.

[4] Freeman, W.T., Anderson, D., Beardsley, P., Dodge, C., Kage, H., Kyuma, K., Miyake, Y., Roth, M., Tanaka, K., Weissman, C., Yerazunis, W., Computer Vision for Interactive Computer Graphics, *IEEE Computer Graphics and Applications*, May-June, 1998.

[5] Freeman, W.T., Beardsley, P.A., Kage, H., Tanake, K., Kyuma, K., Weissman, C.D., Computer Vision for Computer Interaction, *SIGGRAPH Computer Graphics Newsletter, Vol. 33, No. 4*, November 1999, ACM SIGGRAPH.

[6] Hämäläinen, P., Höysniemi, J., Ilmonen, T., Lindholm, M., Nykänen, A. Martial Arts in Artificial Reality, *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'2005)*, Portland, Oregon, 2-7 April 2005, ACM Press.

[7] I-Cube website. (Visited Jan 2006) http://infusionsystems.com/catalog/index.php

[8] Jordà, S., Kaltenbrunner, M., Geiger, G. Bengina, R. The Reactable. *Proceedings of the International Computer Music Conference (ICMC05).* Barcelona, Spain, 2005.

[9] Karjalainen, M., Maki-Patola, T., Kanerva, A., Huovilainen, A. and Janis, P. Virtual Air Guitar. *Proc. AES 117th Convention*, San Francisco, CA, October 28-31, 2004.

[10] Machover, T. Instruments, Interactivity, and Inevitability. *Proceedings of the NIME International Conference*, 2002.

[11] Moeslund, Thomas B.Computer vision-based human motion capture – a survey. Aalborg: Aalborg University, Laboratory of Computer Vision and Media Technology, 1999. (LIA Report; LIA 99-02.-ISSN 0906-6233)

[12] Mäki-Patola, T. User Interface Comparison for Virtual Drums. *Proc. Int. conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, Canada, May 2005.

[13] Mäki-Patola, T., Kanerva, A., Laitinen, J., and Takala, T. Experiments with Virtual Reality Instruments. *Proc. Int. conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, Canada, May 2005.

[14] Open Sound Control project home page, http://www.cnmat.berkeley.edu/OpenSoundControl/ (visited Jan 2006)

[15] Paradiso, J. Electronic Music Interfaces: New Ways to Play. *IEEE Spectrum*, 34(12), 18-30, 1997. Later expanded as an online article, 1998. (visited Jan 2006): http://web.media.mit.edu/~joep/SpectrumWeb/SpectrumX.html

[16] Patten, J., Recht, B., Ishii, H. Audiopad: A Tag-based Interface for Musical Performance. *Proc. Int. conference on New Interfaces for Musical Expression (NIME02)*, Dublin, Ireland, 2002.

[17] Phidgets sensor hardware, http://www.phidgets.com/ (visited Jan 2006)

[18] Poypyrev, I., Berry, R., Kurumisawa, J., Billinghurst, M., Airola, C., Kato, H. Augmented Groove: Collaborative Jamming in Augmented Reality. *ACM SIGGRAPH Conference Abstracts and Applications, 2000.*

[19] Puckette, M. PD – Pure Data, http://www.crca.ucsd.edu/~msp/Pd_documentation/ (visited Jan 2006)

[20] Rodney, B., Makino, M. Hikawa, N., Suzuki, M. The Augmented Composer Project: The Music Table. *In proceedings of the International Symposium on Mixed and Augmented Reality,* Tokyo, Japan, 2003.

[21] Theremin info pages. (Visited Jan 2006) http://www.theremin.info/

[22] The Yamaha Miburi System (visited Jan 2005) http://www.spectrum.ieee.org/select/1297/miburi.html

[23] Wanderley, M., Battier, M. Eds. *Trends in Gestural Control of Music*. Ircam - Centre Pompidou - 2000.

[24] Wanderley, M. *Performer-Instrument Interaction: Applications to Gestural Control of Music*. PhD Thesis. Paris, France: Univ. Pierre et Marie Curie - Paris VI, 2001.

[25] Wright, M., Freed, A., Momeni A. Open Sound Control: State of the Art 2003. *Proceedings of the 3rd Conference on New Instruments for Musical Expression (NIME 03)*, Montreal, Canada, 2003.

Children of Grainger: Leather Instruments for Free Music

Stuart Favilla
Bent Leather Band
Hybrid Initiative
ICEPA
Victoria University, Melbourne
61+3 97301026
sfavilla@bigpond.com

Joanne Cannon
Bent Leather Band
Hybrid Initiative
ICEPA
Victoria University, Melbourne
61+3 97301026
joanne_cannon@bigpond.com

ABSTRACT
Grainger's *Free Music* remains a rich source of discovery for contemporary Australian musicians. *Free Music* represents a significant departure point for electronic musicians and instrument makers searching for new musical language, form and expression. This paper presents research undertaken by the Bent Leather Band exploring Grainger's *Free Music* ideas within a 21st century music-making context embracing live improvisation, instrument and software design. Research outcomes presented in this paper includes a range of creative works; meta-serpent wind controllers, the 4th generation of the light-harp controller, new MAX-based software engines for signal processing, control-modes and strategies for the instruments and music including Bent Leather Band's latest collection of works "Children of Grainger". This paper discusses technical issues confronting the contemporary electronic instrument builder and presents Bent Leather Band's aim to develop *playable* instruments.

1. FREE MUSIC
Since late 2003, Grainger's *Free Music* has been revisited by a number of Australian composers and experimental musicians. In all cases the Grainger Museum audio collection has revealed a surprising amount of interesting material that has redefined our previous notions of Grainger's *Free Music* experimental depth and rigor. Warren Burt's work covers the history of the *Free Music* experiments [1] and presents an appendix of the audio collection. His work rebuilding Grainger's unfinished *Electric Eye Tone Tool*; a seven part *Free Music* player machine, was supported by the ABC listening room and presented collaborative works for the *Electric Eye* by Tristram Carey, Catherine Schieve, Wang Zheng Ting and Warren himself. The Blisters ensemble; an ensemble of Australian improviser/instrument builders including; Jon Rose, Rainer Linz, Tom Fryer, Joanne Cannon and myself, were also commissioned by the ABC listening room to investigate Grainger's Free Music legacy and this created a radiophonic work "Skeleton in the Museum", which was selected for the 2004 International Karl Szcuka Pries.

The Bent Leather Band has continued to work on Grainger. We were surprised to discover the diversity of Grainger's experiments. The breadth of the audio collection was a stark contrast to the musical education we received in Australia; which sorely neglected Grainger let alone his experiments.

We first discovered Grainger's *Free Music* back in the mid 1980s, in the library at LaTrobe University. From there we found the Grainger museum; which has had a long association with experimental electronic music. Finally we began our collaboration with Garry Greenwood and through the exhibition of this work at the Grainger Museum, created the opportunity to work there with our Blisters project.

Grainger's Free Music comprises writings, recordings and the actual machines he created. "Music beyond the traditional constraints of pitch and rhythm" [4], was developed through the construction of many bizarre instrumental experiments and prototypes including; 6th tone tuned pianos fitted with player systems, air-pump powered reed organs capable of fine controlled portamenti, and large machines; such as the *kangaroo pouch*; which allowed the pitch of up to four electronic valve oscillators to be played by score of cut cardboard and paper rolls.

Grainger's explains his *Free Music* as "music using gliding tones and irregular rhythms" throughout the 1951 recordings of his experiments and instruments. The 1938 manifesto [4] explains his desire to liberate or *free* sound from the constraints of conventional pitch and rhythm.

"Existing conventional music (whether "classical"; or popular) is tied down by set scales, a tyrannical (whether metrical or irregular) rhythmic pulse that holds the whole tonal fabric in a vice-like grasp and a set of harmonic procedures (whether key-bound or atonal) that are merely habits, and certainly do not deserve to be called laws." [Grainger, 1938]

This manifesto also reveals a desire to bypass the role of a performer or interpreter of his music. It directly follows Grainger's experiences composing for and rehearsing theremin ensembles. His attempts to have his *Free Music* compositions played by musicians on theremins never achieved results to his satisfaction [5]. It is doubtful that Grainger knew about categorical perception [17] and perceptual limits on the human ear and how it would affect the performance of glides using theremins. Grainger was already using player piano technologies and the potential of piano roll devices became the hub of his Free Music activities.

As improvising live ensemble musicians, we had to consider how much Grainger's preference for paper-roll [player] sequencing was going to influence the formation of our music. As improvisers, we are not interested in a paper-roll or sequencing technique. But perhaps we share with Grainger a common desire to eliminate the "proverbial middleman" or interpreter (Cross, 1976, in Drefuss interview [6]). We do know how impressed Grainger was with improvised music generally. He brought the Ellington Band into one of his composition classes in New York [5] and Grainger's own top ten system rated Rarotongan improvised polyphony 3rd; well above Debussy and even Bach [7]. Therefore, as improvisers, we embrace *Free Music* as an opportunity to escape the rigid harmonic constraints of traditional pitch systems and also as a departure point for the development of new specialised musical instruments.

Figure 1. Detail of Grainger's Kangaroo Pouch Machine [Courtesy Grainger Musuem]

2. NEW INSTRUMENTS

The field of new interfaces for musical expression continues to expand. Musicians are offered increasing access to new technologies that can develop new instruments. Network protocols such as OSC, are beginning to purge the old MIDI language through a range of new interfaces from Kroonde and Gluion. Micro-electronics internet groups such as the MIDI Box network run by Törsten Klose, have made available cheap MIDI circuits and PIC chip-software; allowing musicians the chance to construct their own customizable interfaces. Novel controllers, mixers and DJ spatial sensor interfaces for music are available straight off the shelf in music stores.

A constant proliferation of theoretical literature regarding the development of musical instruments has also flourished over the past five years. Amongst this proliferation are ideas that challenge the relevance of traditional notions of music performance such as whether the role of virtuoso performance is valid? Choi[3], and Paine[13]: and the blurring of the traditional roles of composer, performer and listener.

Twenty years ago Jeff Pressings imagined a super-instrument [14]. It posited a human limit of up to ten independent degrees of freedom and provided the player with multiple channels of quality sensory feedback. Controllers were expected to develop high resolutions, scanning rates and sensitivities capable of performing very fine expressive control of music. Pressing's work defined ten fundamental issues relating to the design and construction of new interfaces.

Although Pressing's instrument has been achieved in part, much of today's work lies in the domain of instruments for public interaction. In stark contrast to Pressing is the work of Ulyate, [18]. His ten commandments of interactivity require "no expertise", "no thinking" and measure quality of control from immediate reactions of players or participants.

Other approaches proposed by Mulder [9], develop new instruments to existing human motor skills, rather than requiring the musician years of commitment to developing new skills. Pedro Rebelo's work applies the media theory of prosthesis to instruments [15]. Physical modeling is used in an intervention of an acoustic sound to mimic, extend or fulfill a potential of the body [acoustic instrument]. For Rebelo the player's intention and the instrument [which he defines as a point of resistance] constitutes an *acoustic threshold*.

Other interesting areas of research include: The continuing development of controllers modeled from existing acoustic instruments such as; Cormac Cannon's *EpipE* Uilleann Pipes [2] and Diana Young's *HyperPuja* [18]. Completely new novel instruments designed for a specific form of synthesis or sound generation; including Sile O'Modráin and Georg Essl's *PebbleBox* and *CrumbleBag* controllers for granular synthesis [12]; and *Blockjam* [11], a polyrhythmic sequencer interface that forms a series of interconnecting block switches. The switch's function is displayed by an LED panel and can change throughout an interaction or piece.

Something that seems to be lacking from the field overall is a development of new original music through new instruments/interfaces and although there are many new contributions made to the field in the form of new instrument prototypes, very few of these prototypes are developed to the next generation. The field has also responded to the rise of sound design over music.

Our musical instrument work has been primarily concerned with skilled ensemble performance of new sounds. Our broad research aims have been to create new music, performance and ensemble techniques and new instruments. So far our work has developed in order to embrace specific musical languages. Over the past five years, our language has specialized in beat-less, microtonal and gliding forms of sonic expression. In essence we have been playing a digital form of Grainger's *Free Music*.

3. EVOLUTION OF PLAYABILITY

Our idea of a playable instrument is one that essentially does not limit or inhibit the development of skill. The key is a balance between the instruments' expressive potential, responsiveness, quality of feedback, embodiment of the sound and the instruments' ability to provide the player with an intuitive understanding about the music being played. The instruments we were going to build had to suit the music we played and also work well together in ensemble. We defined this as *playable*, meaning

- Expressive
- Responsive
- Versatile in solo and ensemble performance
- Visceral [*naturalness*, *appropriateness,* good visual feedback]
- Palpable [allowing for skill development, an instrument you can practice for hours]
- Inspiring [intuitive, revealing new things to the player]
- An instrument that has a definitive sound or character

The focus on playability was intended to unify all aspects of controller interface design across as many possible disciplines, e.g. cybernetics, HCI, ergonomics, gesture research and skill-development. Specific areas we have focused our preliminary research on have included tactile, haptile control and expert skill development Shakel [16], the combination of dominant [attack based] gestures with ancillary or [modifying] gestures and the potential limitations of bimanual control [8].
The goals of the project were as follows:

1. To build an ensemble of new playable electronic instruments
2. Develop a new improvised music and ensemble
3. Build prototypes and develop them into mature aesthetic instruments
4. Explore the language of Percy Grainger's Free Music

The overall aim of the research was to create successful instruments, their playing techniques and ensemble music simultaneously. Our process was reflexive, sometimes beginning with a sound or process and then finding physical gestures that could effectively control them. Sometimes gestures discovered their own sounds. We investigated playability through the development of two distinct instrument projects; one investigating the potential of light sensors to trace virtual strings for a musician to play [the *Light-harp*]; and another project investigating the potential of live signal processing control of double reed [the *Meta-serpents*].

4. LIGHT-HARP

The *Light-harp* uses spotlights and lasers to trace virtual strings through space. The instrument is a MIDI controller and was originally built in collaboration with David Brown [a violin and shakuhachi maker] and Robin Whittle [a notable computer music instrument developer and designer].

Figure 2. *Light-Harp* Ancillary Controllers

After earlier models were built, using wood, fiberglass and steel as construction materials, the current *Light-harp* was made from leather by the talented Tasmanian leather artist, Garry Greenwood. This version supports an extensive array of controllers. These include an active electro-magnetic whammy-bar, a two-dimensional bamboo whammy-bar, two large wheels, breath control and two touch-sensitive strips. It is usually played with up to 5 independent dimensions of freedom. It is also boosted with a control panel of 16 assignable pots for synthesis parameter control. The instrument controls synthesizers, software-synthesisers and signal processing.

The *Light-harp's* specialized hardware allows for the threshold attenuation of light-sensors. This reduces the response time of light-sensors [less than 2msec] and makes sensing beams playable of up to 200 MIDI notes a second. This means that unlike conventional keyboards and other controllers, the *Light-harp* is capable of performing extremely dense and interesting textures not to mention glissandi. It is well suited to the performance of equally tempered microtonal tuning systems such as the sixth tone tuning system used by Grainger to approximate glides and perform "loud unisons", [tremelos] with his own butterfly piano. Within the Free Music project the *Light-harp* has referenced the butterfly piano by using piano samples as a source timbre for all sound creation.

Figure 3. *Light-harp*
Dimensions 164 x 64 x 29cm

The experience of building two previous instruments has brought about changes to the instruments dimension and shape. The neck now supports a scalloped tactile playing surface so the player can feel the sensors sitting under the fingertips. The curvature of the neck has been increased making the instrument's dimension more compact and additionally, the ancillary controllers have been grouped in accordance with bimanual control [8], emphasizing a strong-handed [in this case right-handed] role for leading attack gestures against a left handed passive modification role. Whenever possible, a breath controller is used to control dynamics or attacks.

The aesthetic design of the instrument merges elements from Indian music and 1930's valve radio equipment. Bakelite and French polished controller knobs are set against flat polished leather panels. The Indian elements include the dragon [yali] headpiece, human physiology of the pelvis [instrument base], spine & vertebrae [neck and sensors], lotus flower [the tailpiece], and the fluted trumpet end. With the exception of metal control panels, a strip of supporting metal and wooden pieces supporting the base, the instrument is constructed entirely from leather.

5. SERPENTS

The evolution of the meta-instrument controllers began with the sensor modification of simple double reed instruments. Joanne Cannon, a bassoonist and Australia's chief protagonist for the creation of an Australian electric bassoon, wanted to transport her reed playing into a signal-processing environment. The first prototype instrument used force sensitive resistors and a passive magnetic proximity sensor to track the spatial position of the instrument's bell. This instrument was interfaced via a MIDI control circuit to a laptop running Max which in turn controlled a number of effects units. The musical language we developed for this instrument made heavy use of delays, which we used to create additional parts. These techniques required fine control of delay times and more controllers were desired to independently control the multiple audio streams. The major drawback of this instrument proved to be its limited tonal production. This led to the idea of making long tubes with open holes.

The second prototype instrument we built in collaboration with instrument leather instrument maker Garry Greenwood. The *Serpentine-bassoon* is a leather meta-bassoon, with a 2.4meter conical bore. The instrument has eight open holes; which can be used to play pitches or closed with stoppers allowing for sensors to be played instead. This instrument produced a variety of timbres reminiscent of bassoons and horns. Two contact condenser microphones were used to pick up a large variety of sounds and the signal was processed using MaxMSP via a Digi002. Dials were added for fine delay time and other parameter control and three force sensitive resistors were used to control dynamic features of the signal processing such as acoustic or delay feedback etc.

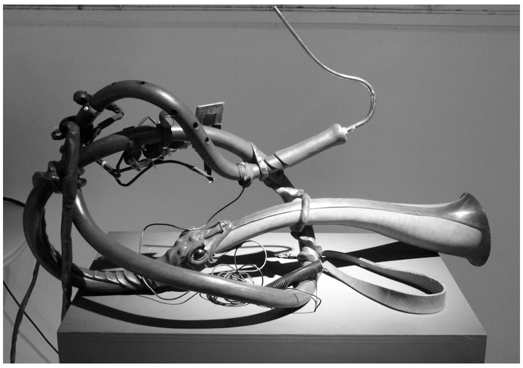


Figure 4. *Serpentine-bassoon*
Controllers attached

The third instrument, dubbed *Contra-monster*, has a 3.6meter conical bore and was built for maximum signal processing control. It has two built in condenser microphones, and 15 controllers including, three dials, one fine tuning dial, one fader, two joysticks and six small force sensitive pads; in the place of finger holes. The sensors have been positioned ergonomically for ease and effectiveness of use and the interface was completed with a small built in display for the performer. The *Contra-monster* is capable of ten simultaneous degrees of freedom.

The current instrument was built around a MIDIBox Plus PIC controller that was redesigned to make the circuit board smaller. A small panel of push buttons allows for the instruments controller mode to be changed allowing for over 760 possible assignments for the MIDI controller signals.

The aesthetics of the serpents combine the same elements adopted by the *Light-harp*. The *Serpentine-bassoon* was made as a direct relative to the *Light-harp* using the same color scheme and leather dyes. The *Contra-monster's* visual aesthetic combines 1930s Bakelite radio dials, French-polished panels and an Indian theme of a lotus or orchid design.

Figure 5. Detail of force-sensitive resistors
Contra-monster, 2005

6. *FREE MUSIC* MAPPINGS

Software is a necessary part of our process and a laptop is the host computer for practical reasons. The Laptop is effectively the live effects studio and Max software allows all of our sensors to be mapped to all the parameters we use to play our music. Our mappings are fixed, not dynamic, but we will usually switch between several mappings during a performance. The instruments however, constitute the interface between the musician and software with the laptop remaining off. Sensor mappings consists of a number of process stages including sensor adjustment [rectification], rescaling, processing [including averaging/interpolation of data]; and finally the mapping and tweaking of a specific parameter of synthesis or signal processing.

Figure 6. Joystick, C*ontra-monster*

Signal processing and synthesis techniques have developed from experiments using delay with modulating or playable delay-time. The technique is commonly associated with echoes. However, if the feedback of the signal and the delay time can be accurately controlled, tones and independent lines can be achieved in a myriad of ways.

Other associated signal processing techniques explored so far have included; pitch-shifting [coarse, tuned, continuous or modulated], extensive control of delays [to create pitched feedback tones], distortion [overdriven or boosted signals, ring-modulation, noise, clipping and unstitched wavelets], and granular treatments [streams, clouds, pitch-shifting, distortion, prolongation, and accenting], this list is not exhaustive. We developed processing techniques in MSP but also had great success hacking groups of existing VST format plug-ins including those found in cycling's Pluggo suite and other freeware plugins.

Throughout the project we were conscious of providing each instrument with its own character. This was achieved by limiting the source timbres. In the case of the serpents their source is the sounds created by the double reed anyway, but the Light-Harp, a synthesizer controller can play a huge variety of sounds. In the theme of Grainger's microtonal butterfly piano experiments, we limited the Light-Harp to using only piano samples.

Figure 6. *Contra-monster*

As our work developed our mapping strategies have grown much larger encompassing sets of over 50 parameters. These mappings are responding to the development of a reflexive approach to playing and are designed to offer a large number of possibilities to a performer. The central idea to these mappings is to create stock standard number of simultaneous sensors whose behavior can then be governed by a set of ancillary knobs or other more passive systems of control.

Figure 7. Visual Display Contra-monster

The Light-Harp for example has a standard playable set of sensors including light sensors [pitch or note/sample trigger], breath control [attack/volume], two dimensional whammy-bar [push=feedback, side to side=delay time], a force sensitive strip allows for another simultaneous control for filtering, two large dials which can also be played controls fine delay-time, modulation speed/depth and or specialized filtering parameters. These main controls are supported by a number of other controllers extending the mapping with up to 64 additional transformations such as; transposition [+-6ve], re-scaling of temperament [quartertone, sixth-tone, 7-tet, 9-tet, 23-tet, 64-tet microtonal sets], fine tuning shifts, modulation controls, envelope controls for filtering or amplitude envelopes, signal processing parameters for delays, flangers, chorusing, flanging, distortion and granular effects. Our experience has found that these larger mappings are intuitive; revealing more each time they are explored.

Intrinsic to the success of mapping gesture is the notion of *embodiment*. This remains a subjective area of research and we define *embodiment* as a convincing relationship between physical gesture and resultant sound. Convincing in this sense does not necessarily mean *realistic*. Nor do we subscribe to the research areas of *audible* gesture or *universal* musical gesture in relation to our work. We believe for example, that the tiniest movement of a fingertip is entitled to make the hugest possible sound. After all, that is a good example of what a digital instrument can do that acoustic instruments [great pipe organs the exception] do not. We also think of *embodiment* as a process. It is discovery, questioning and searching for a response in the context of an artistic discourse, an ongoing dialogue between the musician, the controller interface, the software mapping and the music.

7. FUTURE WORK

Currently the instruments' sensor implementations are limited by the small 7-bit MIDI controller resolutions. Although we have found ways around these limitations in regards to the control of audio and synthesis processing through data interpolation, averaging and smoothing, these techniques result in data hysteresis [sluggishness] and are really only a compromise.

The next stage of development will involve upgrading the instruments to OSC via Gluion interfaces. The Gluion is capable of much faster scanning rates [up to 1ms] compared to other OSC interfaces on the market. This should offer a sense of immediate control with a significant boost to resolution. The signal latency of computer processing remains a significant problem. We have also found software synthesizers to be limited in terms of polyphony and also in regards to tuning system implementation.

In conclusion, this project has created instruments, techniques and music exploring Grainger's *Free Music* i.e. music using gliding tones and irregular rhythms. We have explored and extended Grainger's ideas and legacy through the creation of new *playable* electroacoustic instruments. The research has created a folio of creative work including two finished CD albums, hours of recorded experimental work, International concert performances, exhibitions, television, radio performances and videos.

8. ACKNOWLEDGEMENTS

We dearly acknowledge the contributions of three remarkable Australians. Firstly to Garry Greenwood, who's passing in April this year was a tragic loss. Garry's incredible leatherwork will inspire Australians for years to come. An artist of rare talent we dedicate our work to Garry. We also would like to acknowledge Jeff Pressing and his support and belief in our work over the years previous to his passing. Without Jeff's support, we would never have believed in developing our first prototype. Lastly we acknowledge David S. Brown, his friendship and remarkable talent for instrument innovation, skill and artistry, have given us great inspiration from the beginning of our work in the early 1990's.

7. REFERENCES

[1] Burt, W. *Percy Grainger's Work with Music Technology Melbourne* Independent Media Centre, 2005 http://melbourne.indymedia.org/news/2005/02/87624.php

[2] Cannon, C., Hughes, S., & Ómodhráin, S., "EpipE: Exploration of the Uilleean Pipes as a Potential Controller for Computer-based Music" *In the Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03*), *Montreal, Canada*

[3] Choi, I., "A Component Model of Gestural Primitive Throughput" *In the Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03*), *Montreal, Canada*

[4] Crab, S., *The Free Music Machine* http://www.obsolete.com/120_years/machines/free_music_machine/

[5] Dorum, E., *Percy Grainger: the man behind the music* published by IC & EE Dorum, Melbourne Australia, 1986

[6] Drefus, K., *Interview with Burnett Cross*, 5.3.1976 Audio collection Grainger Museum, 1976

[7] Giles, M., & Pear, D., "Grainger's Top Ten" *Review The Centre for Studies in Australian Music* No. 15 June, 2002

[8] Kessous, L., & Arfib, D., "Bimanuality in Alternate Musical Instruments" *In the Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03*), *Montreal, Canada*

[9] Mulder, A., "Towards a choice of gestural constraints for instrumental performers" *In Trends in Gestural Control of Music* Ircam- Centre Pompidou, 2000

[10] Nelson, M., & Thom, B., "A Survey of Real-Time MIDI Performance" *In the Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04*), *Hamamatsu, Japan*

[11] Newton-Dunn, H., Nakano, H., & Gibson, J., "Block Jam: A Tangible Interface for Interactive Music" *In the Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03*), *Montreal, Canada*

[12] O'Modhráin, S., Essl, G., "PebbleBox and Crumble Bag: Tactile Interfaces for Granular Synthesis" *In the Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04*), *Hamamatsu, Japan*

[13] Paine, G., "Gesture and Music Interaction: Interactive Engagement Through Dynamic Morphology" *In the Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04*), *Hamamatsu, Japan*

[14] Pressing, J., "Cybernetic Issues in Interactive Performance Systems" *Computer Music Journal*, Vol. 14 No.1, 1990

[15] Rebelo, P., & Van Walstijn, "Designing Acoustic Thresholds" *Les journées du design sonore*, IRCAM 2004

[16] Shackel B. [1990] "Human Factors and Usability" In: Preece J. and L. Keller (Ed.) *Human-Computer Interaction: Selected Readings*. Prentice Hall 1990

[17] Siegel, J., & Siegel, W., "Categorical Perception of tonal intervals: Musicians can't tell sharp from flat." *Perception & Psychophysics*, 399-407, 1977

[18] Ulyate, R., & Bianciardi. D., "The Interactive Dance Club: Avoiding Chaos in a Multi Participant Environment." *In the Proceedings of the 2000 Conference on New Interfaces for Musical Expression,* Seattle

[19] Young, D., Essl, G., "HyperPuja: A Tibetan Singing Bowl Controller" In the Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME03), Montreal, Canada

Managing Gesture and Timbre for Analysis and Instrument Control in an Interactive Environment

William Hsu
San Francisco State University
1600 Holloway Avenue
San Francisco CA, USA
hsu@tlaloc.sfsu.edu

ABSTRACT

This paper describes recent enhancements in an interactive system designed to improvise with saxophonist John Butcher [1]. In addition to musical parameters such as pitch and loudness, our system is able to analyze timbral characteristics of the saxophone tone in real-time, and use timbral information to guide the generation of response material. We capture each saxophone gesture on the fly, extract a set of gestural and timbral contours, and store them in a repository. Improvising agents can consult the repository when generating responses. The gestural or timbral progression of a saxophone phrase can be remapped or transformed; this enables a variety of response material that also references audible contours of the original saxophone gestures. A single simple framework is used to manage gestural and timbral information extracted from analysis, and for expressive control of virtual instruments in a free improvisation context.

Keywords

Interactive music systems, timbre analysis, instrument control.

1. INTRODUCTION

Timbre is an important structural element in non-idiomatic free improvisation [2], especially in the work of saxophonists and other instrumentalists who use extended techniques. For true interactivity, the virtual instruments within a software improvisation system should be able to respond to aspects of an improviser's gestural language, including timbre, that might be perceived as significant by human improvisers.

Our interactive music system [1] is developed in close collaboration with British saxophonist John Butcher, well-known for the complex and crucial role of timbre in his sophisticated musical language [3]. The system works with timbral information, as well as more traditional musical parameters such as pitch, loudness and duration. The design goals for our system were these:

1) The system will be used in the context of free improvisation.
2) There will be minimal use of looping or sequencing, i.e., the system will behave in unpredictable ways, like an improviser.
3) The system will be responsive to timbral variations in the saxophone sound.
4) It should work with the range of Butcher's saxophone vocabulary, from extended techniques, to small close-miked sounds, to saxophone-controlled feedback through a sound system.
5) The system will not be a purely player paradigm system in the sense of Rowe [4]. That is, there will be options for a human to intervene and influence the larger shape of the system's behavior.
6) Overly obvious mappings of saxophone gesture to computer-generated gestures should be minimized.

[1] concentrated on the problems of timbral analysis and classification in our system. This paper details enhancements made in the last six months, especially in tracking, managing and coordinating high and low level gestural information. A few excerpts from our residency at STEIM, using an older version of the system from 2003, are at http://userwww.sfsu.edu/~whsu/Timbre.
We continued our work during a residency at ZKM (Karlsruhe) in May 2006, and will have new recordings for audition at NIME 2006.

We will present a selected survey of related work, and describe our system organization, focusing on the components for gesture/timbre capture and management. We will discuss issues of material generation in improvisation, and describe the framework we use for expressive timbre control of our virtual instruments. Finally, we will evaluate our experiences of the system, and discuss future directions.

2. RELATED WORK

Many previous interactive music systems work primarily with pitch and high level gestural characteristics. For example, George Lewis' *Voyager* [5] and Matt Ingalls' *Claire* [Ingalls, personal communication] both use pitch-to-MIDI converters to preprocess the input audio stream. For more examples of systems that work mostly with MIDI, see [4]. Roberto Morales' GRI [6] combines pitch with information from sensors that capture a human improviser's physical gestures on the flute.

In [4], Rowe discusses aspects of Zach Settel's piece *Punjar*, in which timbral characteristics, such as sibilance in the delivery of a vocalist, are used to influence synthesis. In [7], Cort Lippe describes his *Music for Clarinet and ISPW*, and discusses how timbre might be used to control material generation.

Ciufo's *Beginner's Mind* [8] is an improvisation system for use with unspecified instruments. It performs detailed analysis on the input audio stream, using Jehan's MSP external analyzer~ [9]. The real-time data stream from analyzer~ influences the configuration and behavior of a network of processing modules. In addition, statistics (such as the mean and standard deviation of the pitch, loudness etc) are collected for each phrase to create a *perceptual identity* for the phrase. As will be seen in Section 4. we monitor a larger set of timbral characteristics, and also track their progression over the course of a phrase or gesture. Our improvisation agents may use this information to generate responses that make references to timbral feature contours in the human improviser's performance.

3. SYSTEM ORGANIZATION

Our system, implemented in Max/MSP (www.cycling74.com), monitors real-time audio input from an improviser, extracts timbral and gestural characteristics, and uses this information to guide the generation of response material. Figure 1 shows the high-level system organization.

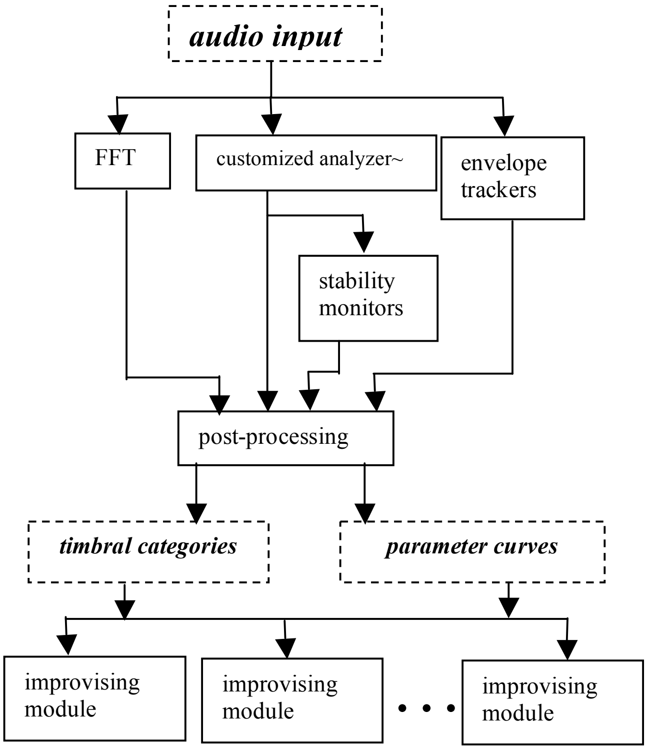


Figure 1: High-level system organization

The audio input stream (Butcher's saxophone sound) is fed into analysis modules. The raw measurements are post-processed to yield broad descriptive categories for timbre, and other performance characteristics. In addition, for each phrase/gesture, the progression of a number of timbral (and performance) parameters are tracked and stored in a repository, to be referenced for generating response material.

4. TIMBRE AND GESTURE ANALYSIS

4.1 Timbre categories

Timbral variation is often an integral component of musical gestures in improvisation. For example, a long saxophone tone might be held, with stable pitch and loudness, but acoustic roughness is slowly increased through embouchure control. An experienced human improviser would perceive and respond to this gestural variation.

Our proposed timbre classification framework attempts to reflect broad perceptual categories from a listener's perspective. [1] described our measurements and strategies for identifying specific timbral categories, which we will briefly summarize. A saxophone tone might be described as

1) *noisy* (vs. not noisy); the prominence of breath noise in a tone

2) containing *harmonic partials* (vs. inharmonic partials)

3) containing a *sharp attack* (vs. no sharp attack)

4) containing *multiphonics* (vs. no multiphonics)

5) with *flutter* (vs. no flutter); we define flutter to be a periodic fluctuation in the amplitude envelope, like a tremolo. (In [1], we confusingly called this *roughness*. Since we are now working with an additional *acoustic roughness* measure, detailed in Section 4.3, we have renamed this category.)

4.2 Gestures and parameter curves

In addition to the on-the-fly classification of the audio input stream into timbral categories, the system also monitors and records the progression of timbral and other musical parameters for each phrase or gesture.

For our purposes, we define a phrase/gesture as a sustained musical statement, possibly containing multiple note on and off events and short silences, separated from other gestures by significant intervals of silence. Each gesture is divided into a sequence of approximately 200 ms windows. For each window in a phrase, we track a set of measurements. Hence, for each gesture, we have a set of curves; each curve represents the variation of a parameter over the gesture. In addition, we track and store timestamped note on/offs through a phrase.

Figure 2 shows a block diagram of how we organize the per-gesture measurements. For each gesture, our system collects a set of parameter curves, parsed from the audio input stream during real-time performance, and stores them in a repository (within dashed outline in Figure 2). We assume a gesture begins when, after a significant (adjustable) period of silence, the amplitude of the input signal increases past a threshold. Starting from the onset of the gesture, we track loudness, brightness, noisiness, roughness, the pitch estimate and its stability/reliability at 200ms intervals. We stop recording parameters for a gesture when a significant period of silence is detected, or when a maximum gesture length is exceeded.

Loudness, brightness, and noisiness are collected using Jehan's MSP external analyzer~; the average of each parameter over a 200ms window is recorded. Hence, each parameter curve represents a list of 200ms averages of that parameter, over the progression of a specific gesture. For pitch, IRCAM's yin~ pitch estimator gives us 20ms pitch estimates and confidence. We also monitor the stability of the pitch estimate over each 200ms window.

The 200ms windowed averages are clearly imprecise; they only give a reasonable indication of input signal characteristics if the signal is fairly stable. If there are, for example, fast runs with pitch changes, the measured parameters will not be meaningful. Hence, it is useful to monitor the stability and confidence of the pitch estimate in each 200ms window, as well as the presence of multiple note on/offs, to validate the timbral, loudness and pitch measurements. Examples of some saxophone phrases and parameter curves for several of the measurements can be found at http://userwww.sfsu.edu/~whsu/Timbre.

While processing measurements from analyzer~ is quite straightforward, handling acoustic roughness entailed significantly more effort, which we will describe in the next section.

4.3 Measuring acoustic roughness

Auditory roughness describes an aural sensation associated with harsh, dissonant sounds [10]. It is one aspect of timbre, and appears to encapsulate, in a quantitative measure, characteristics of some of the timbral categories we discussed earlier, such as flutter, closely spaced inharmonic partials, and prominence of (harsher) multiphonics. Vassilakis [11] has suggested that roughness is correlated with tension/release patterns in Lebanese mijwiz and other non-Western musics; from our experience, it also appears to be a usable approximation in free improvisation.

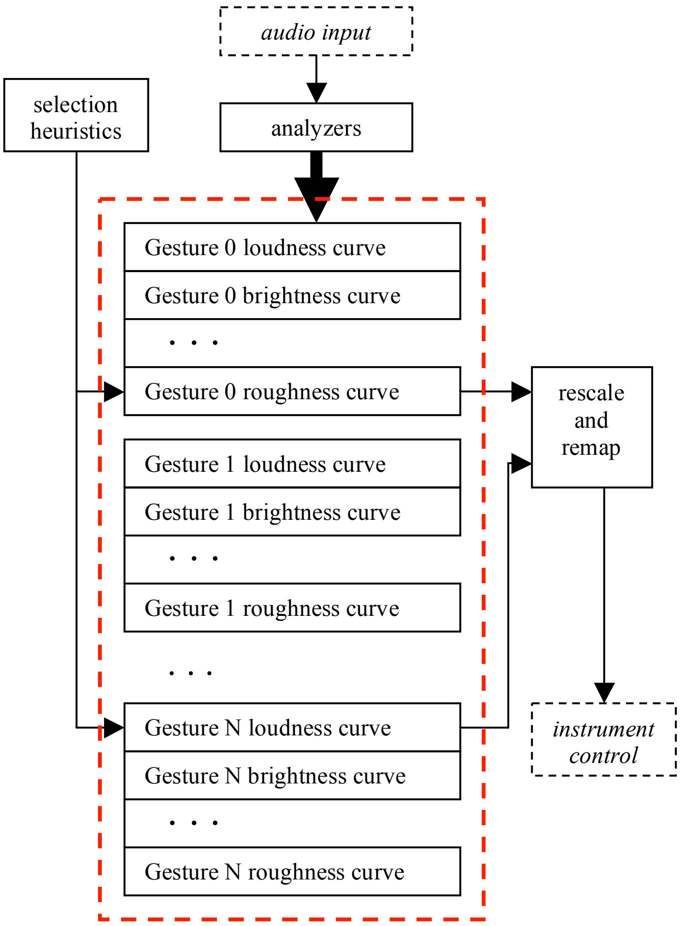


Figure 2. Parameter curve management block diagram

We based our roughness measurement on Vassilakis' model from [10], which is a refinement of an earlier model by Sethares. The basic procedure for estimating acoustic roughness from a complex tone includes these steps: 1) extract frequencies and amplitudes of sinusoidal partials from tone; 2) for each pair of partials, compute roughness contribution based on [10]; 3) sum contributions from all pairs of partials. (See http://acousticslab.com/rougness/learnmore/MoreModel.html) for more details.)

This procedure requires fairly accurate extraction of partials from the audio stream. Earlier versions of our system relied on Puckette's fiddle~ [12] for partial extraction. However, we need to separate partials that are close together in frequency, below 10 Hz. It is possible to increase the window size for fiddle~ to achieve this resolution; however, the significantly higher computation load means that the system will no longer run satisfactorily in real-time on our development platform (an Apple G4 iBook).

We switched our analysis components to Jehan's analyzer~ [9], which uses more efficient FFTs and can operate reasonably well with much larger window sizes. However, we found that some preprocessing of the signal in fiddle~, before the FFT and partial extraction, is left out in analyzer~; a possible consequence is analyzer~'s less reliable measurements observed for both partial frequencies and amplitudes.

After some investigation, we rewrote the partial extraction component in analyzer~, using an algorithm based on Smith and Serra's PARSHL [13]. Our customized analyzer~ object now uses PARSHL's parabolic partial estimation algorithm to extract fairly accurate characteristics of the 20 strongest spectral peaks. Neighboring partials should be at least three FFT bins apart, for good results. We currently resolve partials that are approximately 8 Hz apart.

Information from the partial extraction is finally fed into the roughness model equation based on [10]. Our roughness estimator reports a moving window average of the measured roughness, over a tunable number of overlapping windows (default six windows, or about 1.1 seconds).

In summary, our system is able to capture a set of timbral and musical contours for recent gestures or phrases played by the human improviser. Each gesture is essentially represented by a set of parameter curves of loudness, brightness, noisiness, roughness, pitch and pitch stability, and timestamped note on/off events; they represent the progression of these characteristics from start to finish of the gesture. The curves are stored in a repository to be accessed for instrument control.

5. RESPONSE GENERATION

5.1 Choice of materials

In free improvisation, the choice of material is fairly open, though improvisers generally avoid references to established idioms. The role of pitch tends to be downplayed or obscured; greater weight is placed on loudness, duration, and timbre. Likewise, our system emphasizes managing timbre (and loudness and duration) over pitch.

Smaller gestures with nuanced timbral variations are favored when supporting or engaging in dialog with the human improviser, over larger gestures (such as drones and thick textures) that may take up too much of the sonic space. (The latter are also possible, but should be carefully managed.) At the simplest level, gesture generation in our system involves the pseudo-random selection of a number of parameters, within tunable ranges, their rates of change, and how they might be influenced by audio input.

While we wish to avoid a delay/echo effect in our generated responses, human musicians do make references to each other's materials when improvising. Hence, as the human improviser's gesture choices change over the course of a piece, our system should be able to adjust its behavior to reflect these changes.

The tracking of timbral curves over each gesture, described in Section 4, gives us a variety of gestural materials to work with. By remapping one parameter curve from the input to a different parameter in a future generated gesture, we preserve a general impression of gestural shape and cross-referencing of materials, but without the rigid delay/echo effect that we are trying to avoid. For example, the human saxophonist may play a sequence of tones that increase gradually in roughness. The system may respond with filtered noise whose cutoff frequency increases gradually through the gesture.

5.2 Virtual instrument control

In our system, an ensemble of agents, each "playing" a virtual instrument, responds to the real-time audio input. Each agent monitors the characteristics of the saxophone sound; a combination of internal processes and external stimuli determine the material being generated and performed. Agents may act independently, or form coordinated subunits, with a user making some high level organizational and structural choices (see [1] for more details.)

Each agent/instrument has a set of predetermined gestures that it can choose from. In addition, each agent has access to the parameter curves for recent gestures that have been collected in the repository. As seen in Figure 2, when generating response material, an agent may choose to use one or more parameter curves from the repository of recent saxophone gestures (or use its predetermined gesture set). Any parameter curve can be chosen from one or more recent gestures, rescaled in an appropriate way, time-stretched or otherwise manipulated, and mapped to the same or a different timbral or gestural characteristic to create a new gesture. The same framework for representing a saxophone gesture (a set of timestamped contours of pitch, pitch stability, note on/offs, loudness, brightness, noisiness and roughness) is used to organize most aspects of a generated gesture. Hence, as the gestural language of the human improviser evolves over a performance, the gestural vocabulary available to the virtual instruments will also reflect the changing shapes of the saxophone phrases.

Many of our virtual instruments were chosen for ease of control of a range of timbral characteristics, especially brightness, noisiness and roughness. For example, for a filtered noise generator, brightness is controlled by the lowpass filter cutoff frequency, noisiness by the filter resonance, and roughness by a tremolo envelope. For a metallic-sounding comb filter excited by a noise source, brightness is controlled by varying the bandwidth of the noise source, noisiness by the feedback coefficient of the comb filter, and roughness by either detuning the harmonics of the comb filter, or with a tremolo envelope. Similarly, the brightness, noisiness and roughness of a waveguide bass clarinet can be adjusted by changing the embouchure, the mix of noise in the excitation, and a tremolo envelope for the excitation or embouchure. While not all our virtual instruments have the full range of timbral variations (see [1] for a more detailed description), a significant number of them are adaptable to working with the parameter curves from the captured gestures. Some examples of prerecorded saxophone gestures, their parameter curves, and synthesized gestures using those curves can be found at http://userwww.sfsu.edu/~whsu/Timbre.

6. EVALUATION AND FUTURE WORK

Our initial tests and experiences with the recent enhancements to the system, using recorded saxophone material, have been reasonably satisfactory. A proper evaluation is possible only with the live participation of John Butcher (or another saxophonist). We will work with this system extensively at our residency at ZKM in May, and will prepare recordings for audition at NIME 2006.

With the recent enhancements described in this paper, we seem to have found a usable approach for increasing the adaptability of the system to gestural and timbral variation in the improviser's real-time performance. A single simple framework is used to manage information from analysis for use in instrument control. Future directions include more sophisticated monitoring of both the improviser's and the generated performance, codifying tension/release patterns in performance, and role-oriented coordination of the improvising agents.

7. ACKNOWLEDGEMENTS

Thanks to Miller Puckette,Tristan Jehan and Pantelis Vassilakis for source code and detailed discussions. This work was also generously supported by STEIM, Anne LaBerge at Kraakgeluiden Werkplaats, Chris Burns (formerly) at CCRMA, and ZKM (Karlsruhe).

8. REFERENCES

[1] Hsu, W., Using timbre in a computer-based improvisation system. In *Proceedings of the ICMC* (Barcelona, Spain, Sept. 5-9, 2005).

[2] Bailey, D., *Improvisation: Its nature and practice in music*. Da Capo Press, 1993.

[3] Keenan, D., Mining echoes. In *The Wire,* November 2004. (See also http://www.johnbutcher.org.uk/)

[4] Rowe, R., *Machine Musicianship.* The MIT Press, Cambridge, Massachusetts, 2001.

[5] Lewis, G., Too Many Notes: Computers, Complexity and Culture in *Voyager*. In *Leonardo Music Journal*, Vol. 10, 2000.

[6] Morales R. et al., Combining audio and gestures for a real-time improviser. In *Proceedings of the ICMC* (Barcelona, Spain, Sept. 5-9, 2005).

[7] Lippe, C., A Composition for Clarinet and Real-Time Signal Processing: Using Max on the IRCAM Signal Processing Workstation. In *Proceedings of the 10th Italian Colloquium on Computer Music,* Milan, Italy, 1993.

[8] Ciufo, T., Beginner's mind: an environment for sonic improvisation. In *Proceedings of the ICMC* (Barcelona, Spain, Sept. 5-9, 2005).

[9] Jehan, T., Schoner, B., An Audio-Driven Perceptually Meaningful Timbre Synthesizer. In *Proceedings of the ICMC,* 2001.

[10] Vassilakis, P., Auditory roughness estimation of complex spectra – roughness degrees and dissonance ratings of harmonic intervals revisited. *Journal of Acoustical Society of America, 110(5/2).*

[11] Vassilakis, P., An improvisation on the Middle-Eastern mijwiz: auditory roughness profiles and tension/release patterns. *Journal of Acoustical Society of America 117(4/2).*

[12] Puckette, M., Apel, T., Zicarelli, D., Real-time audio analysis tools for Pd and MSP. In *Proceedings of the ICMC* (San Francisco, USA, 1998).

[13] Smith, J.O. and Serra, X., PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of the ICMC* (Champaign-Urbana, USA, 1987).

Integrated Interactive Music Performance Environment

Dr. Keith Hamel
School of Music
6361 Memorial Rd
University of British Columbia
Vancouver, B.C. Canada
1 -604-822-6308
hamel@interchange.ubc.ca

ABSTRACT

The author has designed and implemented an interactive music performance environment which allows NoteAbilityPro, a music notation and editing program, to be interfaced with Max/MSP/Jitter or Pd. Extensions to NoteAbilityPro allow control messages and notes with additional performance attributes to be embedded in a score, and sent to a network of computers running Max and/or Pd when the score is performed. Score following using the suivi.score object in Max/MSP can be used to synchronize the live performance with the score performance, thereby aligning all control messages with the live performance.

Keywords

Interactive Computer Music, Interactive Performance, Max, Pd, Score following, Music Notation, NoteAbilityPro

1. BACKGROUND

For many years composers and performers working within the field of interactive computer music have been frustrated by the lack of integration between notation software and the interactive performance software commonly used. Typically, a composer will develop an interactive environment in an application such as Max/MSP [12] or Pd [7] which responds to incoming audio signals from acoustic instruments or is triggered by MIDI messages. The composer will usually develop control mechanisms within these programs so that different kinds of processing and/or effects can take place at different times during the composition. If a score is prepared for live performers, this is done in a separate music notation application, and there is usually no direct connection between the notated score and the interactive environment. This lack of connection between the score and the interactive performance application makes simulation and rehearsal of the complete composition difficult, and makes later editing or modifications to the composition awkward since all of the software components used in the composition have to be altered separately.

The author has developed a comprehensive music notation and editing application called NoteAbilityPro [3] which runs on Macintosh computers. Recently, a number of extensions have been added to this program in order to address some of the issues cited above. While this added functionality does not solve all the challenges related to interactive computer music performance, it does provide composers with a much more integrated environment; one that better facilitates rehearsal and pre-rehearsal simulation of the composition. This software environment, called the Integrated Interactive Music Performance Environment (IIMPE), is best suited to composers who are creating interactive computer music compositions with well-defined instrumental parts rather than those working in loose improvisatory situations, though these performances can also benefit from some of the features of this environment.

2. EXTENSIONS TO NOTEABILITYPRO

At the core of IIMPE are extensions to NoteAbilityPro which allow scores to incorporate important performance data in addition to the graphical representation required for score printing and part extraction. While NoteAbilityPro has traditionally emphasized graphical flexibility rather than performance or playback accuracy, most of the recent extensions added to support IIMPE have substantially enhanced NoteAbilityPro's live performance and networking capabilities.

2.1 Embedding Max/MSP or Pd messages

NoteAbilityPro allows Max/MSP or Pd messages to be embedded in a score. These messages are entered into the score as a text class and are located in the score at specific measure and beat locations. The beat location is indicated above the text box, and the location of messages can be freely dragged around the score or copied and pasted elsewhere in the score or into other scores. The format of embedded messages replicates the format that would be used within Max or Pd (a receive name followed by data of arbitrary length and terminated with a semicolon). During score playback, these messages are sent to a receiving application (such as Max/MSP or to Pd) either as UDP network messages, or as System Exclusive MIDI messages. Up to 16 different IP address & port destinations can be specified in NoteAbilityPro and the tracks on which the messages are placed can be directed to any of these destinations. Is it possible, therefore, to direct different streams of data (such as live video controls) to different computers each running different applications. Since the data size of these control messages is very small, no significant latency issues have been encountered – even when using wireless networks.

In the example below, an excerpt of a NoteAbilityPro score shows a flute part (which is intended to be performed live) along with some of the control messages which are to be sent to another computer running Max/MSP. These messages control everything that is happening in Max/MSP over the course of the composition -- the ways in which the flute is processed, the triggering of sound files, samples, and other audio generators, panning controls, etc.

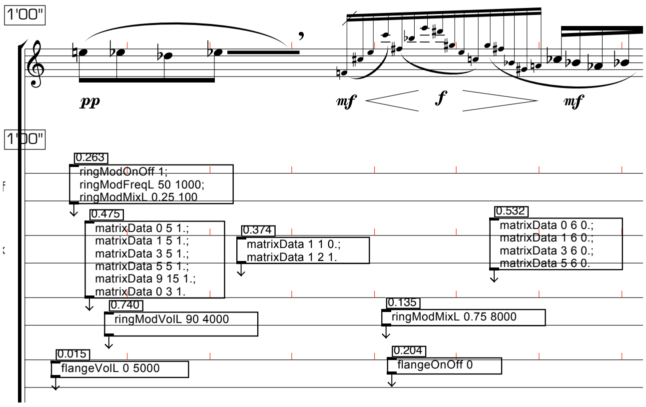


Figure 1. Control messages embedded in NoteAbilityPro

Within Max/MSP or Pd, control messages are received using netreceive [4] objects or with a SysEx parser, each of which pass the message to the corresponding receive object. Some scores may have relatively few embedded messages, while others, such as the example above, may have a single notated instrumental part along with many staves of embedded messages.

2.2 Extended Notes

In order that more complex musical data can also be sent to Max/MSP or Pd, a class of Extended Notes has been added to NoteAbilityPro. Any note entered in the score can be converted to an Extended Note and additional attributes such as microtonal pitch inflection, panning location, and FM harmonicity can be specified. While the default parameters of Extended Notes are mapped to the standard note list data structure used by audio players in the UBC Max/MSP/Jitter Toolbox [3], the data can be interpreted by Max/MSP or Pd in any way the composer desires since they are simply lists of ints and floats. The table below shows the data typically associated with an Extended Note – NoteAbilityPro provides a simple interface for entering and editing Extended Notes.

Table 1. Extended Note Data fields

Data	Value	Data	Value
Pitch	84	C2M Ratio 1	1.215
Velocity	100	C2M Ratio 2	1.0
Channel	1	Stretch Factor	3.5
Pitch Fraction	0.5	Grain Size	150
Duration (ms)	3500	Extra Int 1	0
Pan (LR)	0.28	Extra Int 2	0
Pan (FB)	0.70	Extra Float 1	0.0
Pan (BT)	0.375	Extra Float 2	0.0

2.3 Qlist and Detonate Support

NoteAbilityPro can also save scores or selected parts of scores as Max qlist or detonate objects. These functions allow NoteAbilityPro scores (which might include embedded messages and extended notes) to be embedded in MaxMSP as a qlist (with delta times between events calculated) or to be saved as a detonate object for testing or simulation purposes.

3. CONTROLLING MAX/MSP OR PD

When a score is played by NoteAbilityPro, embedded messages and extended notes are sent to the receiving applications through the designated IP addresses and ports. All tempo changes, repeats and playback settings in the score affect when and how the control messages are transmitted. It is possible, therefore, to simulate a performance of your composition by sending the instrumental outputs from NoteAbilityPro (as audio) into Max/MSP or Pd while having all the control messages and extended notes (which can be generating audio, processing the instrumental sounds, or performing audio analysis) sent as network messages at the same time. Changes to the content of messages in the score or to their placement in the score can be accomplished easily by editing the text objects or by moving the text in the score. In most cases, it is possible to create accurate simulations of the interactive composition without requiring the live instruments to be present. When the composition has been completed, the score will include all notes to played by the performers as well as all the messages and extended notes needed to control all the Max/MSP and/or Pd patches used in the interactive composition.

4. SCORE FOLLOWING

4.1 Manual Following

In the first implementation of the Integrated Interactive Music Performance Environment, score following was done manually by the composer. Score playback was started in NoteAbilityPro, and, with the help of a tempo control panel and the scroll wheel on the mouse, the live performance was loosely synchronized with the instrumental part in the score. With each new performer, a new tempo map was created during rehearsals, and then manual adjustments to the playback tempi were made during performance. While manual score following would not work with all compositions, it proved to be very successful in compositions where some flexibility in the synchronization between the performer and the embedded messages could be tolerated. These interactive performances were much better controlled and much more secure than any previous experiences the author has had with similar performances.

4.2 Automated Score Following

While manual score following suits some compositions, it certainly would not work well with all interactive music, and the fact that it requires a computer operator to be present at all rehearsals and performances is a drawback. The next stage in the development of IIMPE was to implement mechanisms for automated score following. This enables a performer to simply start the NoteAbilityPro score, have their performance tracked using a score follower which in turn adjusts the playback tempo of NoteAbilityPro so that it is aligned with the performer. Once the live performance and the notated score are synchronized, all control messages are automatically aligned to the live performance.

4.3 Score Following using suivi

Various automated score following strategies have been used over the past 20 years. The work of Vercoe [11], Puckette [6], Dannenburg [1], and more recently of Pardo and Birmingham [5], and of Orio and Schwarz [10] provides a wide range of options and approaches, yet the task of achieving accurate and reliable score following in live performance situations remains a significant challenge. While one approach works well for a certain performance situation or music, the same approach may fail miserably in a different performance situation or when following a different kind of score.

In the current implementation of the Integrated Interactive Music Performance Environment, score following using the Max/MSP objects suivi.score and suivi.score~ [8] developed at IRCAM are being tested. While this stage of development is not yet complete, the results so far are very promising. A standard midi file generated by NoteAbilityPro is imported into the suivi.score object and becomes that object's basic internal score representation. Pitch tracking is performed on the live performer (either in Max/MSP or using a Pitch to MIDI converter). The detected pitches are passed to the suivi.score object which sends a synchronization message to NoteAbilityPro through an inter-application or network message. NoteAbilityPro, usually running on a separate computer, receives this message and adjusts its score playback so that it is will be synchronized to a beat location just ahead of the beat location detected by suivi.score.

4.4 Strategies for synchronization

The complexities of score following in live performance situations have been well documented [9], and this environment falls prey to the same problems that all such systems must deal with. Players make mistakes, pitch tracking is sometimes inaccurate, and there are latency issues that must be accommodated. As well, the composer needs to ensure that the entire composition does not break down because an error is made by the performer or a miscalculation is produced by the score following software.

The basic strategy for aligning NoteAbilityPro's playback to the events tracked by the score following software is to make adjustments to NoteAbilityPro's tempo in order that the two scores are more closely synchronized at future events. Currently, there are three modes of synchronization supported by NoteAbilityPro: Fluid, Close, and Tight. Essentially, these three modes control how far in the future the actual synchronization point is. The closer the point of synchronization is in time, the more extreme the tempo changes of the NoteAbilityPro performance are likely to be. When tight synchronization is desired, the score playback tempo is adjusted so that synchronization occurs at the beat location a fraction of a beat ahead of the score following beat location. When loose synchronization is desired, less radical tempo adjustments are made and the synchronization point is further in future. When playing NoteAbility scores that mostly contain control messages, tight synchronization keeps the two score closely aligned. However, when playing scores which include MIDI tracks or other note events, loose synchronization (which result in less extreme tempo adjustments) produces more aesthetically pleasing results.

In order to synchronize the suivi score and the NoteAbilityPro score, a new playback tempo is generated for a short duration, after which the playback tempo is reset to the current tempo. If NoteAbilityPro is running behind the suivi score, the playback tempo is increased, sometimes very dramatically, for a short period of time. If NoteAbilityPro is running ahead of the suivi score, the playback tempo is slowed to allow the suivi score to catch up. The current performance tempo is also updated; it is calculated from the timing of recent synchronization points. The figure below provides a couple of simple examples of tempo adjustments where the synchronization point is 0.5 beats ahead of the beat positions tracked in the suivi score.

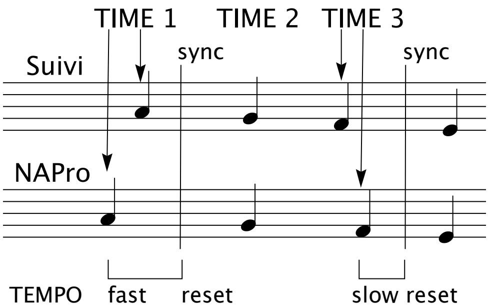


Figure 3. Tempo adjustments with sync point +0.5 beats

4.5 Strengths and Limitations

It might be argued this approach to score synchronization is flawed since the tempo of the playback score is always in flux. There are, however, some practical advantages to this approach, especially for scores having an abundance of control messages rather than timed musical events. First, the fact that the score continues to play (and can never be completely paused or jump to another location in the score) accommodates situations where the suivi gets lost or suggests a score location that it outside a reasonable time horizon – the score will continues to play at the preset tempo until we are back on track or until a cue point in the score is reached. Second, since NoteAbilityPro is able to send messages to Max/MSP at any time, it is easy to programmatically disable score following for musical passages where reliable tracking has proven to be too difficult (e.g. in music containing extended techniques or where heavy noise components are produced by the instruments, or where aleatoric or improvisatory passages are desired). During these passages, the playback score sends a message to the score follower to stop listening and then later to restart score following. The timing and score locations of these control messages can easily be changed during rehearsal.

Currently, the methods used to perform tempo adjustments to the playback score are very simple. We plan to investigate approaches that are able to anticipate tempo trajectories in a performance. We also want to test the applicability of dynamic time warping [8] to the synchronization of the two scores.

5. THE INTEGRATED SCORE

One of the positive results of the developing an interactive computer music composition using this environment is that the score contains a much more complete representation of a composition. Typically, a score created using the Integrated Interactive Music Performance Environment would include the instrumental parts (which can be extracted and given to the players or printed as a complete score and given to the conductor), modified versions of the instrumental parts which are used to generate the suivi scores (since not all the data contained in the complete notated part is necessary for score following), staves containing control messages which will be sent to other applications such as Max/MSP and Pd during performance, as well as tracks containing sound files or standard MIDI data. As well, other kinds of messages (e.g. to control other external devices) may also be included in the score. The more complete the score is, the easier it is for the composer to make changes to the composition (such as inserting or deleting measures, adjusting notes, samples, and control messages, or changing the default tempo map), and the easier it is to manage the data associated with the composition. Finally, having a more integrated score provides a simpler and more complete archive of the composers' creative work.

6. CONCLUSION AND FURTHER WORK

Although this research project is still very much in its infancy, the Integrated Interactive Music Performance Environment has already proven to be a practical and reliable system for designing, testing, simulating, and performing certain types of live interactive computer music. Ultimately, we would like to make this environment as robust and as reliable as possible, and to see it be used in a wide range of musical situations. In order to achieve these goals we will continue to develop and refine the score following system and we will design and test new mechanisms for dealing with score following errors. We also plan to invite a number of composers and composer/improvisers to work with IIMPE in order to assess whether there are stylistic or musical limitations to the system, and to develop strategies for addressing those limitations.

7. ACKNOWLEDGMENTS

I would like to acknowledge the Social Science and Humanities Research Council (SSHRC) of Canada which provided financial support for this research, and the Media and Graphics Interdisciplinary Centre (MAGIC) at UBC which provided research space and administrative support.

Thanks also to Diemo Schwarz, Norbert Schnell, and Serge Lemouton who provide valuable insights and assistance with the suivi objects, and to Bob Pritchard who helped me build the UBC Max/MSP Toolbox modules.

8. REFERENCES

[1] Dannenburg, R. An On-line Algorithm for Real-Time Accompaniment. In *Proceeding of the International Computer Music Conference,* 1984. pp. 193-198

[2] Hamel, K. NoteAbility: A Music Notation System That Combines Musical Intelligence with Graphical Flexibility. In *Proceedings of the International Computer Music Conference*, 1994.

[3] Hamel, K, and Pritchard B. *UBC Max/MSP/Jitter Toolbox.* Retrieved January, 2006 from http://www.opusonemusic.net/muset/toolbox.html.

[4] Mathes, O. netsend and netreceive. Retrieved January 2006 from http://www.akustische-kunst.org/maxmsp/.

[5] Pardo, B. and Birmingham, W. Improved Score Following for Acoustic Performances. In *Proceedings of the International Computer Music Conference,* 2002

[6] Puckette, M. Score Following Using the Sung Voice. In *Proceeding of the International Computer Music Conference,* 1995. pp. 199-200

[7] Puckette, M. Pure Data. In *Proceedings of the International Computer Music Conference*, 1996. pp. 269-272.

[8] Orio, N and Schwarz, D. Alignment of Monophonic and Polyphonic Music to a Score. *Proceedings of the International Computer Music Conference,* 2001.

[9] Orio, N. Lemouton, S. Schwarz, D. and Schnell, N. Score Following: State of the Art and New Developments. In *Proceedings of the International Conference on New Interfaces for Musical Expression* 2003.

[10] Schwarz, D., Orio, N., Schnell, N. Robust Polyphonic Midi Score following with Hidden Markov Models. In *Proceedings of the International Computer Music Conference* 2001.

[11] Vercoe, B. and Puckette, M. 1985. Synthetic Rehearsal: Training the Synthetic Performer. In *Proceedings of the International Computer Music Conference*, 1985. pp. 275-278.

[12] Zicarelli, D., *Max 4.2 Reference Manual,* Cycling '74, http://www.cycling74.com. 2003.

Learning Musical Instrument Skills Through Interactive Sonification

Sam Ferguson
Acoustics Research Laboratory
Faculty of Architecture
The University of Sydney
sferguson@arch.usyd.edu.au

ABSTRACT

Interactive visualisation methods have been used to understand real-time acoustic analysis for the purpose of learning musical skills. However, interactive sonification has not often been considered, perhaps because it is assumed the musician cannot concentrate simultaneously on two sounds – their instrument's sound, and the sonified information. However, whilst some finesse is required in designing sonification algorithms so that they interact with the musician's sound in a controlled manner, there possibly are particular advantages to adopting the sonification approach. This research reports on a suite of interactive sonification algorithms for communicating real-time acoustic analysis results to singers and instrumentalists.

Keywords

Interactive Sonification, Sonification, Sound Visualization, Music Education

1. INTRODUCTION

Research interest in using real-time acoustic analysis to provide feedback to training singers and musical instrumentalists has increased in recent times. However, with any use of acoustic analysis in real-time the communication method employed is crucial if humans are to interpret the information, as the results of acoustic analysis are both cryptic to analyse and often consist of huge numbers of results updating hundreds of times per second. It is very difficult to use this data without some form of visualisation, and many researchers are interested in developing efficient and effective methods for this purpose. Most of these efforts have been visually based and have avoided other methods of information communication.

Welch was one of the first to implement a system for providing visual feedback regarding pitch to primary school aged students of singing. He showed that there were measurable improvements possible by comparing three school classes that were taught to sing using different methods of feedback (including his own visual pitch tracking system) against each other [25]. Both he and Howard have developed this research systematically, recently culminating in the *VOXed* project [23, 24], and the visual feedback system *WinSingAD* [7].

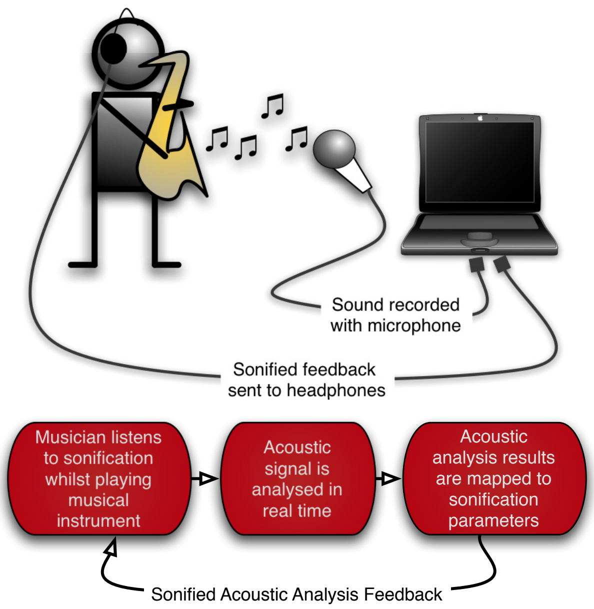


Figure 1: Acoustic analysis results can be sonified to assist with musical instrument practice.

Thorpe, van Doorn, Callaghan and Wilson have developed similar commercial software, which they call *Sing and See*.They have investigated the dynamics of computer based feedback systems extensively with encouraging results [20, 3]. Their system provides feedback concerning the acoustic features of the voice, incorporating a variety of pitch displays and both a 2-dimensional and 3-dimensional (i.e. spectrogram) representation of the magnitude spectrum of the acoustic input.

Ferguson *et al* have built a prototype musical sound visualisation that does not use typical acoustic displays [4]. It incorporates principles of information visualisation, and attempts to present the most relevant aspects of the musical sound with the greatest immediacy by using familiar metaphors. It is aimed at instrumentalists as well as singers.

These feedback systems are designed for a very specific situa-

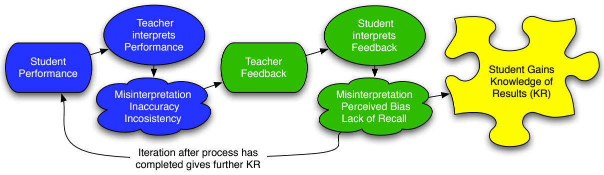


Figure 2: The dynamics of tuition can be investigated in terms of an iterative feedback loop.

tion, and for a particular style of student. The features that are highlighted are mostly 'technical' instrumental skills, and may be more important for specific styles of music (e.g. classical, orchestral, jazz). A successful modern student requires extensive technical control of their instrument in order for them to approach the nuances of musical interpretation exceptionally.

In this research I have drawn upon my knowledge of both acoustic and psychoacoustic research, as well as my experience teaching and being taught musical instruments (mostly woodwind). By attempting to solve some of my own problems maybe a useful approach to musical tuition can be created. It is my hope that the frustrations felt by Rostvall and West [17] can be to some degree alleviated by employing alternative and complementary methods in musical teaching and practise.

2. ITERATIVE FEEDBACK LOOPS

A common way for beginning students to learn a musical instrument is to engage a teacher to explain the specifics of how to physically play the instrument, whilst also often attaining basic musical understanding. A combination of demonstrations and verbal explanation provide a very flexible method of communication allowing the student gain a level of proficiency with their instrument, as well as with general musical skills. However, when the student is attempting to gain higher levels of skill the teacher usually takes a slightly different role, providing verbal feedback to the student about aural and musical impressions of their sound, such as the intonation, sound quality and rhythmic elements. Whilst this is the primary method of musical tuition in use today, there are several difficulties associated with this model of tuition, some of which may be ameliorated to some extent by the complementary use of acoustic analysis. One of the major difficulties is the small amount of time in which they may receive feedback from their teacher. Other difficulties can be seen by analysing the iterative feedback loop that is often present in tuition more closely. This iterative feedback loop incorporates four main parts: firstly the student performs while the master listens; secondly the master interprets the performance and looks for opportunities to improve the student's performance; thirdly the master provides their feedback verbally (predominantly) and the student listens to the feedback; and lastly the student attempts to interpret the master's feedback. The process iterates when the student attempts an improved performance based on the feedback received. This iterative process can incorporate errors, due to partial misinterpretation, incorrect or inconsistent judgement, inconsistency of physical and acoustic conditions, or perceived bias, at each of these stages (Figure 2).

A distinction needs to be made at this stage, because most of the skills we are interested in are made up of various sub-skills. For instance, in attaining fine control of pitch there are two skills necessary: an aural ability to discern pitches, and the psychomotor skills to produce the pitch discerned as correct [3]. The two skills are interdependent, and as such it can be difficult to attain adequate physical ability without adequate aural ability. This aural ability relies on feedback, as the student can not initially discern how the pitch they have produced compares with the pitch they were aiming for [9]. The term 'Knowledge of Results' (KR) has been used to describe the understanding the student receives from the teacher about the sound result they are actually achieving [7]. Feedback is essential to produce KR and the time delay between performance and feedback is crucial to the effectiveness of feedback. In a traditional tuition situation a student's performance for their teacher can be comparatively long, with the teacher only providing feedback after the performance is finished. This feedback must somehow be associated with the student's memory of their performance and the body positions used to achieve that particular part of the performance [22]. If the student succeeds in relating the feedback with performance, and then goes on to repeat the process as intended, the feedback loop still only iterates relatively slowly, with the appropriate length passage needing to be performed each time. Of course, real-time feedback loops iterate as quickly as the performer can perceive the information, and thus there is also little recall involved in associating body and muscle positions with sound results.

3. SONIFICATION

Visualisation systems are widespread; we are often systematically taught to understand numerical information in a visual manner within the primary and secondary school system. However, despite auditory methods for communicating information being shown to function well in various roles, they remain relatively unused. According to Walker and Kramer [21] techniques that use sound to convey information include the following:

Alerts and notifications: are simple sounds designed to alert a user to refocus their attention on some object or event.

Auditory Icons: which are the auditory equivalent of visual icons, and represent their target with sounds that the target produces.

Earcons: are sounds that represent a larger range of messages and meanings, and represent their target metaphorically, possibly with a melody or symbolic sound that is learnt over time.

Sonification: is the use of non-speech audio for information display. The data is 'mapped' to a parameter of sound and therefore the sound changes along a particular axis to represent changes in the data , thus 'sonifying' the data.

It is this final method that is most important for this research.

We are assisted in understanding these issues by the field of information theory. Moles provides an important primer regarding its application to sound and acoustic signals [10]. Moles has defined *channels* as a '...material system which conveys a message from transmitter to a receiver...' He goes on to define *natural channels* as being those channels related to a sense organ, which in our case is clearly the *auditory channel*. If we seek to transmit sonified information using the auditory channel, it seems strange to also attempt to receive information on the same channel. It is possible that the two forms of sound may interact in unwanted and arbitrary manners. This is a possible reason for avoidance of sonification in favour of visualisation. However, this is not a confounding problem when treated with a little finesse, and there are advantages of using sonification as a data communication method for this particular circumstance. For instance, it has been shown that instrumentally trained subjects are more attuned to the parameters often used for sonification [13]. Also, as sonification does not require the monitoring of a visual source, and only interjects aurally when a problem needs to be indicated, it is applicable in 'eyes busy' situations. One such situation is the practice of a piece of music. The most compelling reason for exploring sonification as a feedback mechanism is the possibility that musicians concentrate strongly on the sound they are producing when practising critically. Asking a musician to concentrate on both a visual source to identify auditory problems may disconnect the perception of the two, rendering a musician reliant on visual feedback to hear problems in their sound. Experiments that compare feedback transmitted using various natural channels may clarify whether this is the case.

3.1 Data Sources

For sonification to occur there needs to be data sources to sonify. This data comes primarily from the musician themselves, in the form of an audio waveform. The data is then processed using several acoustic analysis algorithms to provide relevant information about the musical sample. By developing these sonifications in Max/MSP [16] we are able to exploit Jehan's 'analyzer∼' [8] (which incorporates real-time pitch and loudness models), based on the 'fiddle∼' and 'bonk∼' algorithms developed by Puckette [15]. Primarily these algorithms provide us with:

- The time of attack of the note.
- The fundamental frequency of the note played.
- An estimation of the loudness of the note (using a basic psychoacoustical model).

It is by filtering and relating these data sources musically that we can provide more relevant information to sonify. Below we outline these filtering processes.

4. SONIFICATION 'STUDIES'

To teach the student innate musical and aural skills it seems most efficient to focus on one specific task. A reactive feedback system, responding to user's musical sound input with suggestions for correction of one *specific* musical parameter allows the student to focus on improvement. In a typical musical tuition framework we would term these programs 'studies' as they target a specific musical skill. These 'studies' respond intuitively to the note the instrumentalist is playing. They are designed to be either pleasurable or totally silent when the target parameter is within acceptable boundaries, and to intrude with sonified information when the user steps outside these boundaries. These sonifications are explained in further detail below.

4.1 Fine Pitch

Fine pitch control is the ability of the musician to play precisely in tune. It is a skill consisting of many sub-skills, such as noticing and estimating the pitch error, and the use of muscles to correct this pitch error. Over time players will develop strong skills at predicting or 'pitching' notes based on a flexible aural model which can attune itself to the external variables around it (key centre, environmental variables and other players with less precise tuning). Another knowledge base that is developed is an intimate understanding of the notes on the instrument, specifically the natural tuning of the instrument, and the difficulties associated with tuning these particular notes.

In this context chromatic electronic tuners that use visual meters have been a useful form of feedback employed. However, there are also multiple sonification solutions to this problem:

- Playing the 'correct' note for a comparison.
- Playing a sound when the note is out of tune.
- Mixing the sound returned to the instrumentalist with the correct note and/or distortion to produce beat frequencies.

A mix of all three of these options provides the sonification system most likely to be useful to the musician. When the musician plays a note slightly lower or higher than the target note the system responds by playing the *closest correct note*. Amplitude modulation is exhibited at a rate equal to the difference between the two wave's frequency. The comparison note's gain is controlled by the deviation from the closest correct note. The amount of distortion employed is also related to this measurement. Guitarists often use distortion to increase the audibility of beats between two notes when tuning. In this situation, information appears from the interaction of the sonification and the sound source the information being sonified is extracted from.

Of course this sonification design necessarily implies a method of determining correct notes, and for simplicity and flexibility we use the MIDI note scale. This is only relevant for rough tuning tasks, as the equal temperament system on which MIDI note names are based is not always used in instrumental performance. Musicians often gravitate towards the just intonation scheme if they are in a context that does not include an equal tempered instrument (e.g. a piano). A just intonation tuner would be a sophisticated tool for developing a flexible and rigorous understanding of fine pitch manipulation, however it would possibly require additional user input regarding the key the piece of music is being played in.

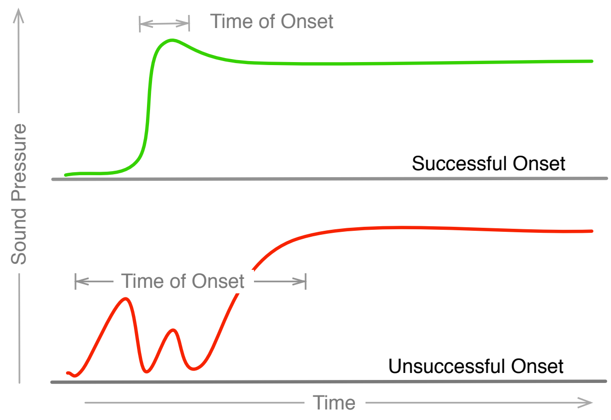


Figure 3: Successful and unsuccessful onsets can be distinguished by the time taken for the level to stabilise.

4.2 Note Onset

One important skill that can be much harder for certain instrumentalists than for others is achieving a clear note onset. Whilst pianists can be assured of a relatively precise onset, members of the woodwind and brass families (due to the acoustics of their instruments) need to practice carefully to achieve a successful onset. Furthermore their success also depends on the range and dynamic level of the note being attempted. It is useful to have some feedback as to how well they articulated a given note, in order to develop a high 'strike rate' for this task.

A clear onset of musical sound is a change from the lack of presence of a sound to the presence of sound in a short space of time. A simple algorithm for determining whether an onset is clear is to measure the time it takes to get from no sound to a stable sound pressure level (Figure 3). The influence of background noise would depend on the path length from the instrumental source to the microphone receiver, and also on the level at which the sound is stabilised at. The instrument acoustics usually do not cause a major problem when the dynamic is fairly loud and/or the instrument is playing in a comfortable range, only when there is a lower level of air pressure involved. Thus the average level used is very important in understanding the difficulty of achieving a satisfactory onset. However, accurate measurement of the amount of sound produced by an instrument requires a calibrated microphone and a constant distance between instrument and the microphone. If this precaution is taken we can provide a measurement of the sound pressure level at which the musician can consistently articulate.

Sonification algorithm presents a melodic sound representing either a successful or unsuccessful onset. A user-defined parameter would decide at what range the onset time is acceptable. A melody, based on the pentatonic scale closest to the pitch played by the musician, would rise by a number of scalar values (between 2 and 5) that indicates by how much the onset time was less than the acceptable onset time. Similarly it would fall related the amount the onset was greater than acceptable onset time. This allows the student to gauge their progress, whilst also controlling the standard they apply to themselves.

It is also useful to sonify the loudness of the sound produced, so there is an extra note added after a short pause at the end of the melody. This uses a general categorisation of the loudness level into integers analogous to the familiar *fortissimo* to *piano*, and is sonified using scalar values of the pentatonic scale again, but with a different timbre and a longer envelope. These sources of information provide students with ample feedback to gauge their ability at producing clear note onsets at different dynamic levels.

4.3 Rhythm

The passage of time has been sonified for centuries, through the use of clocks, bells and other timekeeping devices. A metronome is a timekeeper designed to help students develop innate rhythmic abilities. However, a metronome does not allow much understanding about the magnitude of the correctness in time to be imparted, rather the student learns only whether they sound 'in' or 'out' of time. Indeed, if the student is in time, and the attack of their sound occurs simultaneously with the 'tick' of the metronome and it is much more difficult to hear, due to grouping or masking effects present when two sounds onsets occur simultaneously. Thus the student often alters their rhythm to be before or after the beat to be sure they are listening to the metronome and keeping a correct pace. It could be argued that the reliance on an external source demotes a musician's personal understanding of the beat.

An alternative method of sonifying the beat is to measure the difference between the time of the attack and the correct time as determined based on the beat. The difference is the data source we wish to sonify. Again taking the approach that the user only needs to know about problems they may wish to fix, we design an algorithm that is silent when there is no rhythmic mismatch. This promotes the internalisation of this musical attribute, rather than the reliance on an external source.

This sonification algorithm uses a continuously pulsed unpitched sound to denote inaccurate rhythmic placement. Pulsing increases perceptual prominence, which is important for sounds of such short intervals. This sound is created either; after the attack of a note that is early compared to the correct time; or after the correct time has passed until an attack is detected (as per Figure 4). The length and loudness of the sound is related on the amount of rhythmic inaccuracy, and serves to provide feedback that can be used for understanding not only the occurrence of inaccuracy, but also the extent.

4.4 Loudness Control

Crescendos and decrescendos (increase and decrease in loudness level respectively) are a seemingly simple, yet often quite crucial musical concept. The loudness of sound is of great importance in

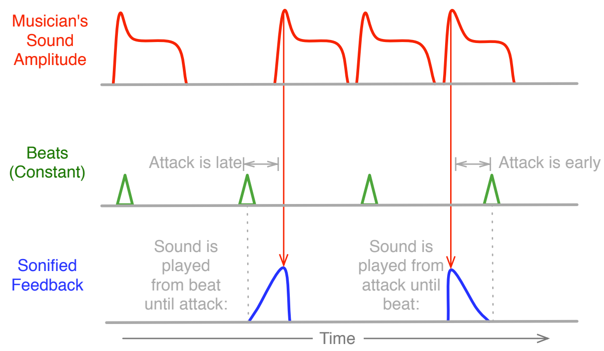


Figure 4: Method for sonification of rhythmic inaccuracy.

producing emotional responses to musical performances, as Schubert has shown [18]. Often teachers spend a lot of their time impressing on students the necessity for rapid changes in loudness to produce adequate excitement for the student's intended audience.

Increasing loudness at a defined rate is only simple before consideration of the difficulties inherent in estimating the likely perception of what 'loudness' means and how to measure it. Loudness as a perceptual sensation has been discussed by many authors [5, 19], and has also been modeled with a degree of usefulness [26, 12, 6, 8]. Furthermore, adaptation and fatigue often affect a listener in judgements of loudness [11], and loudness is strongly affected by stimulus frequency and spectral content [14]. It is generally reasonable to assume that by using a loudness model we more closely understand the percept that would be created for a listener, than if we use only sound pressure level.

Once we have decided on a scale for the loudness of a sound we can decide on what to measure regarding loudness. One of the most important abilities of a musician developing control over loudness is to achieve precise control of a single long note. This develops muscle control, respiration and endurance and is often a very important warm-up procedure for players of wind instruments. Another important ability is control of loudness whilst changing note, especially across registers. Following the mastering of static loudness levels, crescendos with constant (often quite slow) change of loudness level are often also attempted.

Thus sonification of data from a loudness model is an alternative method for developing an understanding of loudness. An algorithm that provides very clear representation is to play a short pulsing sound when the loudness exceeds boundaries around a specified loudness level. This pulsing sound represents the extent the loudness is exceeding the limit by the pulse frequency. A separate pulse sound is used for exceeding upper and lower loudness boundaries. This pulsing sound is clearly quite intrusive, but only occurs when the boundaries are overstepped, and thus the motivation to control loudness is increased.

This algorithm can be applied equally to stationary loudness levels or changing loudness levels. All that is needed to apply this algorithm to changing loudness is to calculate the change in loudness over time and compare this with a user-defined change speed. Then the algorithm needs a user-defined start point, most easily provided by using the loudness calculated at the start of a long note, the attack of which is detected using an algorithm like the previously mentioned 'bonk~'. In this way, students can easily and immediately perceive an accurate description of their ability at controlling both constant and changing loudness.

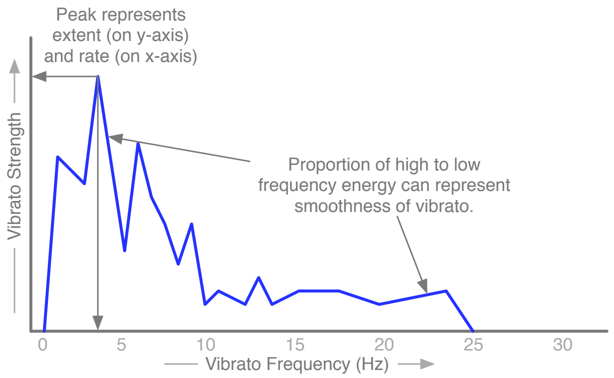


Figure 5: Examining the Fourier transform of a train of pitch values provides information about characteristics of the musician's vibrato.

4.5 Loudness and Legato

Legato, the musical impression of smoothness is highly prized in musical performance. Many factors can affect legato, but at least in woodwind playing one of the chief problems is synchronising depressing multiple fingers upon keys, and maintaining adequate air pressure. Practising musicians need to learn how listen for these gaps, and how to gauge which gaps are the most problematic. Some type of measurement of the extent of this gap may aid the user.

The use of the loudness sonification algorithm described previously is obviously an option, excepting that the parameters need to be redesigned. This situation is obviously different; the aim is not to maintain constant loudness, but only to not let loudness drop suddenly to low levels for short periods of time. Devising a strategy based on user-definition of the period of time that qualifies as a 'gap' allows the user to set a boundary condition that will trigger an earcon sound to be played.

4.6 Vibrato

Vibrato can be subdivided into three further attributes. Vibrato *Rate*, *Extent* and *Smoothness* can be defined in terms based on the vibrato spectrum. We may define the vibrato spectrum as the fourier transform of the pitch values sampled over a period of time, commonly one second or thereabouts.

In a 'vibrato spectrum' we may expect a peak at the fundamental frequency of the vibrato, somewhere in the range between 2-10 Hz, dependent on the singer, the note and the part of the note being sung. Vibrato rate changes dynamically and expressively, and Bretos and Sundberg have described how the vibrato rate tends to rise exponentially towards the end of important notes in classical arias [1]. The vibrato 'smoothness' can be estimated from the comparison of high frequency with low frequency energy in the vibrato spectrum (see Figure 5). This is based on the assumption that the smoothness of vibrato is related to sinusoidal (as opposed to harmonically complex) modulation at the vibrato rate. Whilst it is possible to use a ratio of the sum of energy below and above an arbitrary midpoint, a more appropriate method is to calculate the spectral centroid. The centroid gives us a frequency which can be considered the 'centre of gravity' of the spectrum, and can be calculated as follows:

$$C = \frac{\sum_{n=1}^{n} f_n a_n}{\sum_{n=1}^{n} a_n} \quad (1)$$

where f is the centre frequency of a bin, a the amplitude, and n the number of frequency bins in the spectrum. The vibrato spectral centroid is mapped to the amplitude of a sine tone an octave below the current pitch. As the vibrato spectrum centroid decreases towards the fundamental frequency, and the 'smoothness' of the vibrato increases, the amplitude of the sine tone increases. This may give the impression of the sound being 'reinforced', and perhaps also an increase in the size in the sound [2], and will thereby provide user feedback regarding this performance aspect.

The vibrato rate is sonified by using a mapping to the harmonic series. A categorical binning of the range between 2 and 10 Hz (avoiding the DC component and its rolloff) is first undertaken, mapping the vibrato rate to a series of integers, which are then multiplied by the current pitch, and then played back to the musician using a low-amplitude sine tone. A cross-fade is used for changes between these pitches to avoid arbitrary boundaries, and to avoid startling the user.

For different instruments vibrato and tremelo are often terms that are interchangeable, and refer to both cyclical pitch modulation and cyclical amplitude modulation. This sonification and analysis system can be applied to either pitch or amplitude trains.

5. IMPLEMENTATION, PRACTICALITIES

Use of the feedback system requires a computer, microphone and headphones. The software is reliant on the *Max/MSP* environment [16], and also requires a small number of freely available digital signal processing extensions developed by Jehan [8].

Acoustically, a room with low background noise, a medium to large volume and a reasonably low reverberation time is optimal. The avoidance of unwanted acoustic effects is important, as they could strongly affect some of the algorithms, especially those that deal with time. An optimal microphone position would be quite close to the instrument, but must be certain not to preference particular notes on instruments whose radiation pattern depends on the note played (e.g. woodwind).

6. EVALUATION

It is clear both theoretically [22, 9], and from qualitative studies [3] that there is great promise in visual feedback systems for musicians. However, little research has been carried out into acoustic analysis feedback systems that do not employ visual displays, and as far as the author knows this is the first example of an interactive sonification feedback system for this purpose. The existence of an alternative method of feedback allows research into comparison between methods.

Any feedback communication system is only as good as the feedback it is intending to give. If there are errors in the feedback presented to students, they will either learn bad practices or start to ignore the unreliable feedback being presented. Whilst this implementation intuitively attempts to only display information that is useful for musical instrument students, some method of evaluating the relevance of particular acoustic information would be helpful to maximise the applicability of acoustic analysis to musical practice.

This system introduces another layer additional to traditional sonification algorithms; the use of a single sensory channel for both data source and sonified feedback requires that interaction be controlled to avoid arbitrary interaction. This possible interaction brings the possibility of planned or unplanned emergent sonification.

This system for feedback can be taken at face value, as a system that seeks to 'lay down groundrules' to train a musician towards ideals of technical ability. However, it can also be approached with a less orthodox viewpoint. The sonification methods, if subverted, could lead to interesting and original results. Conversely, the algorithms used to derive the data sources could be altered towards usefulness for various purposes. As they are based within traditional musical practice they may be less abstract than other data sources sometimes used for sonification and may lend themselves to interactive or improvisational musical performances.

Whilst feedback does seem to be useful, it is important to investigate the usage of these systems in a tuition situation with user studies and quantitative methods. This will give some idea of the applicability of this mode of tuition.

7. CONCLUSION, RESEARCH AGENDA

We have described an interactive sonification system for supplying feedback to a musician to guide their instrumental practice and tuition. The system acts to provide the musician knowledge of their results, in turn associating their actions directly with their musical results and thereby developing muscle memory and technical musical skills.

Further research could focus on optimising this approach to music education, as well as a demonstration of its efficacy (or otherwise) in a teaching situation. Comparison and partnership with visual methods would be interesting for its multi-modal implications. Alternatively, exploration of the creative reuse and alteration of both sonification algorithm and data source derivation could be attempted. Most importantly though, methods for accessing musical parameters that musicians respond to and manipulate need to be collected for feedback systems to be effective.

8. ACKNOWLEDGEMENTS

This research was carried out with the assistance of an Australian Postgraduate Award and a Departmental Top-up Scholarship. The author acknowledges discussions with Andrew Johnston, Shigeki Amitani, Andrew Vande Moere, Densil Cabrera and his instrumental teachers Sharman Pretty, Mark Walton, and Josef Hanic.

9. REFERENCES

[1] J. Bretos and J. Sundberg. Measurements of vibrato parameters in long sustained crescendo notes as sung by ten sopranos. *Journal of Voice*, 17(3):343–352, 2003.

[2] D. Cabrera and S. Tilley. Vertical localization and image size effects in loudspeaker reproduction. In *Proceedings of the AES 24th International Conference on Multichannel Audio*, Banff, Canada, 2003.

[3] J. Callaghan, W. Thorpe, and J. v. Doorn. The science of singing and seeing. In F. Zimmer, editor, *Proceedings of the Conference of Interdisciplinary Musicology*, Graz, Austria, 2004.

[4] S. Ferguson, A. Vande Moere, and D. Cabrera. Seeing sound: Real-time sound visualisation in visual feedback loops used for training musicians. In *Proceedings of the Tenth Symposium on Information Visualisation*, London, UK, 2005. IEEE.

[5] H. Fletcher and W. Munson. Loudness, its definition, measurement, and calculation. *Journal of the Acoustical Society of America*, 5:82–108, 1933.

[6] B. R. Glasberg and B. C. J. Moore. A model of loudness applicable to time-varying sounds . *Journal of the Audio Engineering Society*, 50(5):331–342, 2002.

[7] D. M. Howard, G. F. Welch, J. Brereton, E. Himonides, M. DeCosta, J. Williams, and A. W. Howard. WinSingAD: A real-time display for the singing studio . *Logopedics Phoniatrics Vocology*, 29(3):135–144, 2004.

[8] T. Jehan. *Perceptual Synthesis Engine: An Audio-Driven Timbre Generator*. Masters Thesis, Massachusetts Institute Of Technology, 2001.

[9] A. E. Kemp. *The Musical Temperament. Psychology and Personality of Musicians.* . Oxford University Press, Oxford, 1996.

[10] A. A. Moles. *Information Theory and Esthetic Perception*. University of Illinois Press, Urbana, Ill., 1966.

[11] B. C. J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, San Diego, California; London, 1997.

[12] B. C. J. Moore, B. R. Glasberg, and T. Baer. A model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society*, 45(4):224–240, 1997.

[13] J. G. Neuhoff, R. Knight, and J. Wayand. Pitch change, sonification, and musical expertise: Which way is up? In *Proceedings of the International Conference on Auditory Display*, Kyoto, Japan, 2002.

[14] J. G. Neuhoff, G. Kramer, and J. Wayand. Pitch and loudness interact in auditory displays: Can the data get lost in the map? *Journal of Experimental Psychology*, 8(1):17–25, 2002.

[15] M. Puckette, T. Apel, and D. Zicarelli. Real-time audio analysis tools for PD and MSP. In *Proceedings of the International Computer Music Conference*, University of Michigan, Ann Arbor, 1998.

[16] M. Puckette and D. Zicarelli. *Max/MSP*. Cycling 74/IRCAM, version 4.5, 1990-2006.

[17] A.-L. Rostvall and T. West. Analysis of interaction and learning in instrumental teaching. *Music Education Research*, 5(3):213–226, 2003.

[18] E. Schubert. Modeling perceived emotion with continuous musical features. *Music Perception*, 21(4):561–585, 2004.

[19] S. S. Stevens. The measurement of loudness. *Journal of the Acoustical Society of America*, 27(5):815–829, 1955.

[20] C. W. Thorpe, J. Callaghan, and J. van Doorn. Visual feedback of acoustic voice features: New tools for the teaching of singing. *Australian Voice*, 5:32–39, 1999.

[21] B. N. Walker and G. Kramer. Ecological psychoacoustics and auditory displays: Hearing, grouping and meaning making. In J. G. Neuhoff, editor, *Ecological Psychoacoustics*. Academic Press, New york, 2004.

[22] G. F. Welch. A schema theory of how children learn to sing in tune. *Psychology of Music*, 13(1):3–18, 1985.

[23] G. F. Welch, E. Himonides, D. M. Howard, and J. Brereton. VOXed: Technology as a meaningful teaching aid in the singing studio. In *Proceedings of the Conference of Interdisciplinary Musicology*, Graz, Austria, 2004.

[24] G. F. Welch, D. M. Howard, E. Himonides, and J. Brereton. Real-time feedback in the singing studio: An innovatory action-research project using new voice technology. *Music Education Research*, 7(2):225–249, 2005.

[25] G. F. Welch, C. Rush, and D. M. Howard. The SINGAD (SINGing Assessment and Development) system: First applications in the classroom. *Proceedings of the Institute of Acoustics*, 10(2):179–185, 1988.

[26] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. Springer, Berlin; New York, 1999.

Recent Developments in Violin-related Digital Musical Instruments: Where Are We and Where Are We Going?

Cornelius Poepel
Department of Music
The University of Birmingham
Edgbaston
Birmingham B15TT, UK
cp@khm.de

Dan Overholt
CREATE, UC Santa Barbara
Center for Research in Electronic Art Technology
Media Arts and Technology Program
Santa Barbara, California 93106
dano@create.ucsb.edu

ABSTRACT
In this paper, some of the more recent developments in musical instruments related to the violin family are described, and analyzed according to several criteria adapted from other publications. While it is impossible to cover all such developments, we have tried to sample a variety of instruments from the last decade or so, with a greater focus on those published in the computer music literature. Experiences in the field of string players focusing on such developments are presented. Conclusions are drawn in which further research into violin-related digital instruments for string players may benefit from the presented criteria as well as the experiences.

Keywords
Violin, viola, cello, bass, digital, electronic, synthesis, controller.

1. INTRODUCTION
A violin-related digital musical instrument can be either a *physical instrument* that incorporates or mimics a bowed string instrument, a *software instrument* that is controlled by or mimics bowed string instruments, or quite possibly a combination of these two things. Figure 1 shows a few past and present developments in violin-related interfaces for electronic music.

In the case of *physical interfaces*, some new instruments can be played in exactly the same way as a traditional violin, while others require the performer to learn non-traditional gestural techniques. For example, Max Mathew's electric violin, which is one of the first developments in violin-family digital instruments, is played with traditional violin technique, while other developments such as Dan Trueman's BoSSA [15] are designed around an entirely new set of gestures (albeit borrowing from traditional technique as we discuss later). One example of a hybrid interface that combines these elements and brings together both traditional and non-traditional gestural techniques is the Overtone Violin [10].

A *software instrument* can be a synthesis algorithm or a processing algorithm, or a combination of the two. Synthesis algorithms are generated entirely from scratch in the computer and are typically controlled by parameter updates from an external controller with sensors that detect the performer's gestures. Processing algorithms take input from an external audio source (such as the strings of an electric violin) and either modify it with reverb, delay, etc. or use it as a stimulus for modulation of filters or other signal processing algorithms. One example of a violin-related synthesis algorithm is the use of physical modeling to simulate the Helmholtz motion of a bowed violin string [5]. An example of an algorithm that uses the audio from a viola's strings as stimulus is Audio Signal Driven Sound Synthesis [11].

A wide variety of violin-related digital musical instruments have been invented – the intention here is not to survey all known developments, but to outline the criteria that make a given development suitable or unsuitable for a specific purpose. A development that is weak in a certain area can often be strengthened by combining it with another technique to produce a hybrid.

2. LOOKING BACK
In order to get a view on where we are, it is helpful to take a look into the past. Surveying the last 10 years of developments, we analyzed the proceedings of the International Computer Music Conference ICMC (1995-2005) and the Proceedings of the conference on New Interfaces for Musical Expression NIME (2001-2005). We found 28 (18 ICMC and 10 NIME) articles that were related to the violin family.

Figure 1. Clockwise from top left: Max Mathews playing his electric violin, Neal Farwell's Funny Fiddle, Dan Trueman's BoSSA, 2 generations of Tod Machover's Hyperviolins, Chris Chafe's Celleto, and Suguru Goto's Superpolm MIDI violin.

The violin is often said to be an instrument with a high potential for musical expression and a high quality of sound. We wanted to investigate if the reason for the works presented in these publications were related to this expressivity or quality, and on which component of a digital instrument they were focused.

According to our statistics, 84% of the developments we looked at mentioned the bowed string instrument family in their reasoning when considering expressivity or sound quality. By separating the papers into the four categories of controller, synthesis, processing, and complete instrument we found the following: The majority (11 publications, 61% of the ICMC publications) focus on sound synthesis, wherein the majority of these (8) are related to physical modeling. The largest class of developments discussed in the NIME proceedings were complete instruments (6 instruments, 60% of the violin related NIME publications). Four of the 28 publications focused mainly on the development of a controller; two from ICMC and two from NIME. The remainder of the papers were divided among processing (one publication) and others (one publication on violin pitch tracking and two publications on software to do analysis of bow strokes).

While the majority of researchers said they would like to use the expressivity or quality of a bowed stringed instrument, it was interesting to see how expressivity or quality was defined, where in the instrument or playing process they thought it was located, and how it was assimilated into the development of a digital instrument. Except in one case, there were no publications that explicitly defined musical expression or expressivity of a violin.

Regarding synthesis-related research, the quality of the instrument is mostly thought to be located in the physically describable behavior of the instrument. The majority of the publications locate the expressivity in a fixed set of player gestures and playing parameters. These are: right hand—bow speed, bow pressure, bow position, and left hand—finger position and finger pressure. With respect to the connection between the gestures and resulting sound of a traditional instrument there is no major belief found which sound-conditions have to be met in order to keep an acceptable instrumental coupling and synthesized specific sound.

Concerning mapping, it was found that violin related publications directly addressing this topic are rare. One-to-one mapping was found predominantly, and one-to-many only occasionally. Many-to-one mappings were not explicitly found in our search, and only two publications addressed the mapping issue in a broader fashion, e.g. in the development of an “advanced and intelligent mapping interface” [3].

Most researchers were concentrating on scientific goals in the published literature. 11 developments (4 presented at ICMC, and 7 at NIME) were said to be ready for stage use. It was found that over 50% of the developments had the goal to sound like a traditional bowed stringed instrument, that over 50% of the developments had the goal to give the player the "feel" of a traditional bowed stringed instrument, and that over 75% of the developments behaved (in terms of articulation) similar to a traditional bowed stringed instrument.

In general, the evaluation of results and capabilities of the developments in terms of expressivity or quality were done by the authors. Six publications out of the 28 showed that external evaluation was involved, with two of the six using empirical methods to evaluate the work. This might be said to reflect the goals of the developments, which more often than not fell into the categories of personal use or research.

Additionally, there have been many important developments that to our knowledge have never been published. Some of these are quite significant in the field, and are discussed here—a few them that the authors have knowledge of include Max Mathew’s electric violin, Chris Chafe’s Celletto, Neal Farwell’s “Funny Fiddle”, Jon Rose’s MIDI bow, and Peter Beyl’s IR-violin.

One of the earliest examples of an experimental bowed string instrument is Max Mathew’s electric violin, which used a piezo-ceramic bimorph pickup system to capture the vibrations of the strings. The same pickup technology was used on Chris Chafe’s Celletto, which he began building in 1988 [2]. The Celletto is an ongoing and pioneering project, and has evolved through a series of embodiments in many performances. Along the way, various sensors have been used to capture the gestures of playing, such as strain gauge sensors and an accelerometer on the cello bow. In one context, a Buchla Lightning controller was used to track the bow, an elegant solution to the problems associated with developing custom sensor devices.

Neal Farwell’s “Funny Fiddle” instrument was used in his composition *Gipsy Fugue* in 1996 and is also still undergoing development. Jon Rose’s MIDI bow was developed at STEIM in Amsterdam, and incorporates sonar sensors to allow a violinist to lift the bow from the string and continue to play with the bow alone. Peter Beyl’s IR-violin is an altered violin with infrared transmitters and receivers as sensors in place of the strings.

3. NEW VIOLIN-RELATED INTERFACES: THREE TOPICS TO THINK ABOUT

To be sure, any new musical instrument must consider the three areas of human interfaces, sound generation, and the mapping of data between these input and output systems. Here we uncover some of the concerns that arise when designing, developing, and performing with new violin-family digital instruments. Examples of specific instruments are given where applicable, and – inspired by David A. Jaffe’s article “Ten Criteria for Evaluating Digital Synthesis Techniques” [6] — criteria for estimating effectiveness in performing scenarios proposed. We assume that the purpose of a new violin-family instrument is that of performance (other contexts for their use, such as individual or institutional research or personal enjoyment are beyond the scope of this paper). These recommendations are given as our personal opinion of how the criteria can ideally be met.

3.1 Human Interfaces – Gestural Controllers and Sensor Technologies

Although technical issues such as sensor resolution, latency of transmission, and wireless capability all have impacts on new interfaces, we will not focus on these engineering problems here. Instead, we start by looking at the gestures enabled by the interface, and how they allow a performer to extend or enhance the playability of a violin-related digital musical instrument.

3.1.1 How Intuitive are the Gestures?

When designing a new interface, one decision that needs to be made is what type of gestures are to be captured—the answer can fall into two different categories. Some musicians are interested mainly in using the gestures they have already developed through years of practice on traditional violin-family instruments, while others would prefer novel gestures to be available as control inputs. In either category, developers should consider how 'natural' a gesture feels when designing new human interfaces and sensor technologies. In the first case, sensors should be used to capture traditional gestures with as much accuracy and precision as possible, and in the case of non-traditional gestures the interface should use a sensor system that allows for gestures that are in some way related to the traditional playing motions of violin-family instruments. Of course an instrument easily becomes "something else" (no longer a violin-related instrument) if this relationship is broken.

Suguru Goto's Superpolm violin [3] is an interface that requires alternate gestures to be used as performance input, by substituting electronic sensors for strings and synthesis algorithms for acoustics. The instrument is equipped with touch-strip sensors on the fingerboard and a bow that works as a resistor ladder pressed against a voltage sensor on the bridge, plus a chin squeeze sensor for an added dimension of control. While it is impossible to use traditional playing techniques on the Superpolm (since it doesn't have strings), the gestures it requires are closely related to those of a traditional violin. Given this correlation, the Superpolm is a good example of an interface that employs non-traditional gestural input, and the use of a pressure sensor under the chin rest seems to be a natural fit for added expressivity. Although chin pressure does nothing on a traditional instrument, it could be argued that squeezing the violin harder or softer is an intuitive method of input as it relates to the overall player's effort.

Another interface that captures non-traditional yet violin-like gestures is Dan Trueman's Bowed-Sensor-Speaker-Array, or BoSSA [15]. This instrument includes elements of both the violin's physical performance interface and its resonating body, yet eliminates both the body and the strings. It replaces the body with a "spatial filtering audio diffuser", a spherical speaker designed with multiple drivers to eliminate the directionality associated with normal loudspeakers, and multiple sensors mounted on a moveable fingerboard in place of the strings. At first glance, most of the gestures (except bowing) associated with playing the BoSSA might seem to be counter-intuitive to a traditional violinist. However, as the developer is himself a violinist, the motions necessary to control the instrument have been carefully designed to overlap with several aspects of violinistic gestures.

In both traditional and non-traditional gestural interfaces, it is the authors' opinion that those looking to extend bowed instruments should expect to spend some time learning a new set of gestures if they are to have an impact in far-reaching ways; it just helps this process if such new gestures are put forth by the instrument developer in an intuitive manner.

3.1.2 How Perceptible are the Gestures?

Gestures should cause an understandable change in the sound for the performer to best grasp an instrument's playability. A gesture that causes a difficult-to-predict change in the sound may be interesting at first, but it can drive a performer crazy if they are trying to control such a sound in front of a live audience. On another level, the actions of the performer should have clear consequences in order for the interaction to be perceived by the audience. Preserving some sense of mystery in the performance is also important though, and may be accomplished partly through a composition but also via the design of the instrument itself.

A recent development that focuses on capturing traditional bowing gestures is the Ircam augmented bow [13] developed by Emmanuel Fléty. This system uses a coin-cell battery to power the electronic sensors mounted on a violin bow, and a radio link transmits the data to a receiver that communicates with the computer via OSC. The augmented bow can be used as a research tool to investigate the perception of bowing gestures as received by the computer, or as a live performance interface on stage. Gaining a better understanding of musical gestures such as those used in traditional bowing technique is an important step to perceptible gestures, and interfaces such as these greatly improve this by providing a high-resolution link to the digital world.

There have been several other developments involving the capture and perception of traditional violin family gestures, such as those from the Hyperinstruments group at the MIT Media Lab. The Hypercello [8] as developed by Joe Paradiso and Neil Gershenfeld was based on a RAAD electric cello, and had an extensive array of sensors to catch as much detail as possible. The left hand finger position, finger pressure, and right hand bow position were all detected through the development of custom sensors. The Hyperbow Controller [16] by Diana Young is the most recent of the MIT developments, and also uses a wireless transmitter on the bow along with strain gauge sensors to gather data showing the changes in bowing pressure over time.

Regarding the audience perception of violin-family instruments, public knowledge has accumulated to come to expect certain things from something that looks or is played like a violin – this common perception allows new developments that use traditional gestures to break the expectation, surprising the audience with previously unheard sounds. However, for non-traditional gestural interfaces there is no common reference as a key to comprehension for the audience, which puts the responsibility of helping an audience understand what is happening on the developer and performer of such instruments. Therefore, a novel instrument should carefully consider the perceptibility of its gestures both to the performer and to the audience.

3.1.3 How Physical/Powerful are the Gestures?

Making an obvious physical gesture should have a significant audible effect. Electronic technology allows even a tiny motion to have a huge outcome, however it is important to take into account the dramatic effects of a gesture in the design of a new instrument. As such, the performance interface should attempt to provide a vehicle for expressive communication with an audience. Human effort should be incorporated into the design where possible in order to bring out the inherent relationship between instrument and performer. Obviously, if a tiny gesture causes a big sound the performer may have difficulty controlling the instrument. In addition, the consideration of effort will have an impact on the music, lending it a "human feeling", as more exertion is required for some musical ideas than others.

Finally, the choice of whether or not to leave behind the core elements (strings, horsehair, rosin) of traditional bowed-string instruments when developing a new interface is crucial. The history, convention, and institution that comes with traditional instruments may or may not be desirable for a certain development, but dropping these core elements leaves behind a powerful interface for the trained musician. If the instrument has strings that are still playable in the traditional sense, then a single gesture can be made more powerful by simultaneously controlling the sound of the strings along with the digitally generated sound. Hybrid instruments such as Curtis Bahn's Sensor Bass [1] and the Overtone Violin can provide a way of bridging the gap between the world of acoustic instruments and the new possibilities offered by computer music, as they incorporate multi-parametric control along side the traditional instruments interface.

3.2 Sound Generation – Analysis, Synthesis and Manipulation

The method of analysis, synthesis or sound manipulation used in a violin-family instrument has a vital effect on its playability, and many researchers have identified and experimented with signal processing algorithms for this purpose.

3.2.1 How Well-Behaved is the Algorithm?

A frequent approach to controlling a synthesized tone with a violin-family instrument is to use a pitch and amplitude-tracking algorithm. While these trackers tend to be fairly well-behaved in some situations (for example when used with an electric guitar), the violin family of instruments can induce errors in many such algorithms. Bowed-string traits such as 'fuzzy' note-starts and indefinite pitches can be problematic for a tracker, and may result in incorrect pitch estimates. A synthesis algorithm that is fed this wrong information will then produce audible artifacts, an effect that can render them undesirable and displeasing to the performer and audience.

Clearly, synthesis and processing algorithms should attempt to avoid such artifacts. Work in this area has been done by Tristan Jehan, who has developed an enhanced version of Miller Puckette's *fiddle~* Max-object called *analyzer~* [7] that tries to avoid these problems, and also estimates loudness, brightness, and noisiness in the incoming signal. While it is imperative for some synthesis algorithms to know the pitch a performer is attempting to play, other processing techniques do not need this information at all, and therefore may be better choices in many situations. A simple example is a pitch-shifting algorithm, which manipulates the incoming sound directly, modifying the A/D input in either the time or frequency domain. This comes with its own challenges such as formant preservation, etc. but there is no external limit caused by errors in pitch tracking or loudness as to how well-behaved such synthesis algorithms can be. Developers should take these concerns into account when designing or choosing algorithms to use with a new instrument.

3.2.2 How Realistic/Unique is the Sounds Identity?

One development in synthesis that is related to the violin family is Bernd Schoner's Digital Stradivarius project [14], based on the mathematical technique of cluster-weighted modeling. This method concentrates on the simulation of acoustic phenomena, thereby attempting to emulate an actual violin. The potential of this approach is evident in the types of parameters the synthesis algorithm has—bow pressure and speed that are applied to the mathematical model. Input from a controller then is easily mapped and can provide results closer to real world instruments. However, there are cases where exotic (non-violin) synthesis algorithms are desirable as well, and many performers would like to take advantage of sounds that have a more unique identity. In the final analysis, the appropriateness of a given sound depends on the musical task at hand.

3.3 Mapping – Sensor Inputs to Synthesis Parameters

It should always be remembered that the physical input device and the synthesis algorithm are only pieces of the whole instrument, and one must take into account the importance of the mapping as well. This stage can in fact "make or break" an instrument.

3.3.1 How Rich is the Mapping Methodology?

This topic concerns whether the controller inputs and synthesis parameters map in an intuitive manner to musical attributes like musical dynamics and articulation, or whether they are just mathematical variables with very little correlation to real-world perceptual or musical experience. Mapping is heavily interconnected with both sensor inputs and synthesis parameters, in that limiting factors can arise from both sides. For example, an instrument without a sensor for bow position could not directly control a physical model that expects this as a parameter, and a sample-playback-based synthesis engine would not respond in complex ways to bow sensor input. Evidently, every instrument uses some kind of mapping methodology in order to connect performer inputs to sound outputs, but there can be many levels of richness and variety in treating the problem.

Camille Goudeseune has developed a system that uses a SpacePad motion tracker to map various synthesis techniques to the position of a violin in 3-dimensional space [4]. His examples proceed from very simple extensions of standard violin technique up to much richer demonstrations of what is possible when multiple layers of mappings are placed between the performer's physical input and the system's sonic output. The simplest uses the violin's position in space to control the position of its sound source in stereo. One of the more sophisticated mappings controls Hammond Organ additive synthesis by "letting an automatic timbre rover explore a few thousand times of the instrument, with instructions to choose a dozen timbres that differed enough from each other to adequately represent the whole space" (the definition of difference comes from a psychoacoustic model of the human hearing system). This timbre rover is a tool for setting up a very rich mapping methodology that allows for a wide variety of sounds.

3.3.2 What is the Widest Range of Expression?

It is the authors' view that new violin-related musical instruments should focus on deepening sensitivity to the control of micro-gesture that a well-trained violinist possesses. This can be accomplished in part through mapping by scaling control values and making the most of the available sensitivity of a given physical interface. Also, physical interfaces can incorporate very high-resolution sensors and force feedback – a technique Charles Nichols researched for his dissertation at CCRMA [9]. His development, the vBow, is a virtual bow controller designed to accurately sense the motions of a bowing gesture while providing haptic feedback in the form of tactile simulations of detents, elasticity, and barriers produced by electronic motors.

4. EXPERIENCES

In addition to the view of past developments, it is important to mention some personal experiences that we think give inside views on the contributions necessary to construct digital bowed stringed instruments.

4.1 Traditional String Players

Until now, we have focused mainly on developments that have culminated in research tools for the authors, however the next section deals with more general questions about the direction we are headed when it comes to string players throughout the world. As pointed out in [16] the study of the player – instrument interaction is important. One field to collect experiences for basic principles of violin playing is the field of violin pedagogy.

It is a known circumstance that the change from one violin teacher to another can cause the need to change playing techniques and gestures, sometimes beginning even from the basics again. Such changes very often come along with a different overall view on "how music should be performed in order to be of high quality". According to our experience, all the differences cannot be explained with the simple statement that one way is simply a bad one while another is a good one. We conclude that different answers to the question, "What are the main needs of a string player in order to do what is necessary for playing music if we build a new instrument?" may not be too big a surprise. Asking people around us, we indeed got many different answers on this question as presented in section 4.2.

Testing different versions of digital violas [11], it was interesting to see what each subject did with the sonic artifacts (e.g. wrong pitch detections) the systems were producing from time to time. While most string players tried to avoid these, there were some instrumentalists who started to play with it and mentioned this might be a good opportunity to special kinds of sound production. However, this is not a primary feature of the instrument, but it may or may not become a feature primary by the definition of the player.

4.2 >hot_strings SIG<

Confronted with the fact that the knowledge about the fascinating new developments was rare in the world of more traditional string players, the first author started to do workshops presenting these developments to them. In 2004 this lead to a community of people interested in general kinds of inventions to the family of bowed stringed instruments called >hot_strings SIG<. The SIG includes professionals from performers, instrument builders, researchers and composers based in Europe. Meetings occur twice a year with the goal of sharing knowledge, presenting and discussing new instruments, repertoire, research results and aesthetic positions.

Presenting some of the recent developments discussed here (using movies, sound examples, summaries of publications and demos) the assumption was that this would generate a lot of interest from the SIG members, since they are definitely interested in extending their expressivity and open for new sounds. However the feedback received was smaller than expected.

This result raised questions about the subjective reasons for such reluctance, and the follow-up question of what the needs and wishes of the SIG members were if they could have digitally extended versions of string instruments. The reasons mentioned for this lack of enthusiasm were different from person to person. Here is a selection of the statements:

- Some developments were not felt to be extensions of traditional bowed stringed instruments. By listening to sound examples and watching movies the most developments were estimated as to be interesting from a scientific point of view, but in terms of sound and expressivity estimated more as a reduction than an extension compared to existing instruments.

- Reducing the right hand input to the bow parameters of position, pressure, and speed was said not to cover playing techniques like pizzicato, col legno, and a lot of techniques used in contemporary music.

- In order to get the sound qualities of a bowed stringed instrument the resonances of the body (or in this case sound synthesis) have to resonate back to the strings. Otherwise there will always be a different playability and sound characteristic that may keep a lot of players unsatisfied.

Discussing the needs an extended bowed stringed instrument should have brought different answers. Asking further it became clear that the reasons came from different aesthetic points and different experiences with computer-based stringed instruments. Statements found here were:

- New instruments should not only have a string specific-playability, they should also have a string specific sound, different from the one of traditional instruments but within a specific range that enables the player to use the known gesture-input and sound-feedback loop. An important thing is to be able to create sensuality with the instrument.

- I want to make new music with the new instrument and therefore I need it to sound very different from a traditional violin. Keeping a basic set of similar gestures is necessary for me, however an instrument with a different feel and sometimes unconventional reaction is quite ok. A Zeta® violin with additional bow tracking methods would fit my needs.

- New Instruments have to be flexible and extendable in the playing parameters that can be tracked, the mapping and the methods of sound synthesis. Building an instrument is an act of composition and includes an aesthetic point of view since it has to be defined what kinds of gestures and sounds are more or less important.

- If we want to expand the violin with an electric/synthesizer violin, than the instrument has to be able to deal with the complexity of the player. The players string specific ability has to be assignable to the extended new instrument. Everything in terms of bow position, bow speed, more than that, everything that is done in nuances of sound has to be transmittable.

While the aspect of timbre plays an important role when a string player tests traditional instruments, it became obvious in discussions that the timbral color palette a violin maker may want to offer to the players is thought of in a completely different way than a synthesizer developer might. A violin may sound for a synthesist always like a violin and therefore be boring in richness of sounds. A violin player however, can have a completely different opinion in this regard, since all the colors of sound she or he is controlling fall within this space and are sufficient to give a full range of expressivity.

5. DISCUSSION

Facing the evolution of developments in recent years, we still see a challenge to make the areas of new interfaces and synthesis more rewarding for the broader world of string players. Regarding the evaluation methods in papers and looking at the opinions presented in section 4.2, we think it is necessary to obtain a better feedback loop between developers and string players (as long as the development is focusing on that area).

Taking the “top-down” approach, many of the developments analyzed in section 2 first define what the needs of a player are, then design the instrument, and finally evaluate whether the development has met the original criteria. Regarding the different needs articulated by musicians, different aesthetical positions, and different understandings of how to play bowed string instruments, we might try to avoid a view on string players through the glasses of an objective fixed average player-instrument interaction. We would instead like to discuss a “bottom-up” approach that is oriented at the needs of individual musicians as an apparent alternative. With respect to the criteria proposed in section 3 we feel there may be an effective way to build some basic digital instruments and then work to enhance them from the “ground up”, while incorporating feedback from instrumentalists. Of course we also expect that while our criteria is at present our best estimate of important design considerations, it may include presumptions which will have to be corrected according to individual positions that have yet to be found.

6. CONCLUSION

With respect to the fact that we find in the world of string players communities with similar aesthetic positions, we conjecture that over time some sets of criteria will arise that are specifically relevant for those communities. According to our own experiences and to the statements presented in section 4, we hypothesize that a set of three basic digital instruments in the violin family could look like this:

- Playing without a defined and fixed parameter-set but reduced to ASDSS-sounds: an instrument like the eviola presented in [12].

- Playing with predefined playing parameters able to use any known synthesis method: a Zeta® (or similar) type instrument expanded with a bow tracking system e.g. the Ircam Bow[13].

- Playing with the methods mentioned above and with new gestures: an instrument like the Overtone Violin[10].

With this approach we will be able to study the instrument-qualities within the not yet known quality-criteria of the players, their playing-style and their aesthetics. These three instruments as proposed will surely not be the only or even the main ones used in the future, but we see in this way a possibility to bridge the gap between the fascinating and powerful possibilities the digital age has brought to us and the culturally powerful community of string players who are seeking to enhance their musical language.

We have seen that many different approaches to violin-related instruments have occurred in last decade or two. While they all contribute to the whole, to a certain extent they tend to be idiosyncratic developments that have goals focused primarily on individual use. This could be partly because of the nature of doing research into new technologies, or possibly because we are simply in a transitional period in the history of violin interfaces, and the territory that lies ahead may lead towards developments that stick around longer. Call it a “new renaissance”, if you will — when digital instruments transpire to allow a new generation of virtuosi to emerge by providing the right affordances to performers and becoming practical enough to make it outside of the research labs. For any instrument to survive the test of time, it must be accessible (available throughout the world), have a repertoire (even the violin itself would not have survived without this), and most importantly it must be inspiring to future generations of performers and composers!

7. ACKNOWLEDGMENTS

We have attempted to present a balanced view of the instruments discussed. Nevertheless, our bias toward hybrid instruments cannot be hidden. We apologize to those researchers whose developments were not explicitly cited in this article. The choice was biased by the instruments with which we were most familiar, having seen, played, or read about them in various articles.

8. REFERENCES

[1] Bahn, C and Trueman, D. Interface, Electronic Chamber Ensemble *Proceedings of the CHI 01 Workshop New Interfaces for Musical Expression (NIME01),* Seattle, USA, 2001

[2] Chafe, C. Personal Communication, 2006

[3] Goto, S. and Takahiko, S. The Case Study of Application of Advanced Gesture Interface and Mapping Interface, - Virtual Musical Instrument "Le SuperPolm" and Gesture Controller "BodySuit" *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, Hamamatsu, Japan, 2004, 207-208

[4] Goudeseune, C. Resonant Processing of Instrumental Sound Controlled by Spatial Position *Proceedings of the CHI 01 Workshop New Interfaces for Musical Expression (NIME01),* Seattle, USA, 2001

[5] Jaffe, D.A. and Smith, J. O. Performance Expression in Commuted Waveguide Synthesis of Bowed Strings. In *Proceedings of the 1995 International Computer Music Conference,* Tokyo, Japan 1995, 343-346.

[6] Jaffe, David A. Ten Criteria for Evaluating Synthesis Techniques, *Computer Music Journal* Vol. 19(1), MIT Press, Spring 1995, 76-87

[7] Jehan, T and Schoner, B. An Audio-Driven Perceptually Meaningful Timbre Synthesizer *Proceedings of the 2001 International Computer Music Conference,* Bejing, China 2001, 381-388.

[8] Machover, T. Hyperinstruments: A progress report 1987 – 1991. Technical report, MIT Media Laboratory.

[9] Nichols, C. The vBow: A Development of a Virtual Violin Bow Haptic Human-Computer Interface *Proceedings of the 2002 Conference on New Interfaces for Musical Expression* Dublin, Ireland, 2002, 29-32

[10] Overholt, D. The Overtone Violin: A New Computer Music Instrument, *Proceedings of the 2005 International Computer Music Conference,* Barcelona, Spain 2005, 604-607.

[11] Poepel, C. On Interface Expressivity: A Player-Based Study, *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, Vancouver, Canada, 2005, 228-231

[12] Poepel, C. and Dannenberg R. B. Audio Signal Driven Sound Synthesis, *Proceedings of the 2005 International Computer Music Conference,* Barcelona, Spain 2005, 391-394

[13] Rasamimanana Nicolas, Fléty Emmanuel, Bevilacqua Frédéric, *Gesture Analysis of Violin Bow Strokes*, GW 2005, Lecture Notes in Artificial Intelligence 3881, S. Gibet, N. Courty, and J.-F. Kamp (Eds.), 2006, 145–155.

[14] Schoner, B. Cooper, C. and Gershenfeld N. Cluster-Weighted Sampling for Syntheis and Cross-Synthesis of Violin Family Instruments *Proceedings of the 2000 International Computer Music Conference,* Berlin, Germany 2000, 376-379

[15] Trueman, D. and Cook, P. BoSSA: The Deconstructed Violin Reconstructed, *Proceedings of the 1999 International Computer Music Conference,* Bejing, China 1999, 232-239

[16] Young, D. S. New Frontiers of Expression Through Real-Time Dynamics Measurements of Violin Bows, Master’s Thesis, M.I.T. 2001

Composing for Hyperbow: A Collaboration Between MIT and the Royal Academy of Music

Diana Young
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02142, USA
diana@media.mit.edu

Patrick Nunn
Royal Academy of Muisc
Marylebone Road
London NW1 5HT, UK
patrick@patricknunn.com

Artem Vassiliev
Royal Academy of Muisc
Marylebone Road
London NW1 5HT, UK
avassiliev@hotmail.com

ABSTRACT
In this paper we present progress of an ongoing collaboration between researchers at the MIT Media Laboratory and the Royal Academy of Music (RAM). The aim of this project is to further explore the expressive musical potential of the Hyperbow, a custom music controller first designed for use in violin performance. Through the creation of new repertoire, we hope to stimulate the evolution of this interface, advancing its usability and refining its capabilities. In preparation for this work, the Hyperbow system has been adapted for cello (acoustic and electric) performance. The structure of our collaboration is described, and two of the pieces currently in progress are presented. Feedback from the performers is also discussed, as well as future plans.

Keywords
Cello, bow, controller, electroacoustic music, composition.

1. INTRODUCTION

1.1 Background and Motivation
As the field of new music interfaces grows, great achievements are being made in the creation of controllers, as well as in composition and performance techniques featuring them. Alternative controllers, such as The Hands of Waisvisz [18], have been demonstrated to have truly virtuosic capabilities. Controllers inspired by traditional acoustic instruments, such as Tarabella's Imaginary Piano [16], Burtner's MetaSax [1], Scavone's Pipe [14], and the Hyper-Flute of Palacio-Quintin [9], extend the sonic possibilities, performance techniques, and metaphors of their counterparts. New string interfaces are a subset of these [2-4, 6-8, 11-12, 17].

All of the above interfaces represent many different approaches to sensor design, mapping, composition, and performance, yielding great musical and intellectual rewards. However, it is quite difficult to find cases of new music interfaces that have the benefits of a large associated repertoire and large group of dedicated players. Of course, these are not simple features to attain.

Often new interfaces are designed primarily for the use of the designer alone, or for a small number of select musicians. But even when the desire to disseminate is great, numerous practical issues, such as the robustness of the hardware and software involved, often interfere with the goal to make controllers available to more users. And, of course, it is often difficult to decide on the appropriate time in a new controller's development for feedback from a larger audience and user group, as the urge to iterate is ever-present.

Our controller, the Hyperbow [19], is one of the many that lack two of the basic characteristics, a large repertoire and large group of players, possessed by well-established instruments. The Hyperbow was built for use in real-time violin performance applications, such as required in Tod Machover's *Toy Symphony*, in which the Hyperbow made its debut. Given the rigorous demands bowed string players make on their instruments, as well as the impositions of a touring musical work, great care was taken to design the system with a high level of playability, usability, and robustness.

The Hyperbow performed well, achieving its task of an expressive new music controller that enables traditional violin technique. The Hyperbow system was successfully used in four public performances of Tod Machover's *ToySymphony* (2001-2002) by two different violinists. Later, it was featured in a performance of Michael Alcorn's *Crossing the Threshold* for the opening of SARC (2004).

However, although the Hyperbow had been used in performance by several players (who produced invaluable feedback concerning their experiences), it still cannot be considered in the same class as well-established acoustic and electric instruments. If the Hyperbow is ever to achieve the unqualified status of a real music instrument, it must not only provide wonderful new sonic possibilities, but must also be associated with a significant repertoire and have many players. Additionally, it must be easily maintained by these players, and ultimately, have its own performance practice.

In order to achieve these goals, it is essential that the Hyperbow be placed in the hands of more composers and performers. It is this belief that is the core motivation of this collaborative work.

2. HYPERBOW REVISIONS
The first Hyperbow system [20], a descendent of the original MIT Hyperstring Project [6], was designed to capture elements of violin bowing gesture for use in real-time performance applications. Installed on a commercial carbon fiber violin bow and electric violin, it features accelerometers on the frog, force sensors (composed of strain gauges) mounted on the bow stick, and an electric field position sensor that includes an antenna mounted behind the bridge of the violin. This last component of the sensor system is an adaptation of the position sensor first used in the Hypercello project [10].

The Hyperbow system is battery-powered and transmits its sensor data wirelessly via an RF communication module. An external electronics board receives the data and sends it to the computer via the serial/USB port.

In January 2005, when this collaboration officially began, the Hyperbow technology was already several years old. Though we could have significantly redesigned the hardware in preparation for this new work, we opted not to do this. Instead, we decided to postpone a major revision to a later date when the process could benefit from exposure to more composers and performers.

However, because of the different range of bowing movement required by the cello and the fact that we would be using acoustic cellos, some small but critical revisions of the existing hardware were required. Specifically, the electric field position sensor was adapted to produce a signal with greater amplitude emitting from the bow (so that it could be detected by the original receiver from the greater bow-bridge distance of the cello).

When used for violin performance, the position antenna was mounted directly on the solid body of a Jensen electric violin by means of a simple screw and right angle bracket. Obviously, such a scheme was not possible for our work with acoustic cello, so the antenna was fixed to a threaded rod, which was then mounted on the underside of the tailpiece by means of a plastic clamp. (This was a variation of an arrangement used in the Digital Stradivarius controller [15].)

Figure 1:Hyperbow and acoustic acoustic cello (© Roberto Aimi).

Interestingly, although the antenna mounting just described was functional, the musicians determined that it was easier and more convenient to attach the antenna to the strings with tape (behind the bridge). Also, they sometimes found it useful to shift the location of the antenna with respect to the center of the bridge, as pictured in Figure 1. These improvisations arose as the participants experimented with the Hyperbow and were taken as positive indications of increasing comfort with and ownership of the technology.

3. A NEW COLLABORATION

3.1 Structure

This project includes researchers from the MIT Media Laboratory, whose primary role is that of interface design, and composers and cellists from the Royal Academy of Music.

Two Hyperbows for cello were built at MIT and then transferred to the RAM's permanent instrument collection. Though we were interested in an equal exchange of information and ideas, one of our intentions was for the artists at RAM to create, rehearse, and perform without any need for outside technology support, and to be able to freely develop their own individual work processes.

3.2 Schedule

This collaboration began in January 2005, at which time the collaborators from MIT traveled to London for a week long workshop. Two Hyperbows (slightly revised, as described below) were presented to the colleagues at RAM, and the week began by imparting technical knowledge of the Hyperbow system and related software such as Max/MSP.

After these introductory exercises, each composer/cellist team began the work of creating the first compositional sketches of the project. These evolved throughout the week, and on the last working day we made a presentation to the RAM community.

The composers continued to independently develop their pieces through the spring of 2005, and in June 2005, the group met again, this time in Boston. After another week of concentrated work together, we presented the progress at MIT.

In November 2005 during the Association Européenne des Conservatoires Congress 2005 at Birmingham Conservatoire, the first performance for the outside community was given of two of the works in development, Patrick Nunn's *Gaia Sketches*, and Artem Vassiliev's *MODES*. These two pieces were performed again a month later at RAM for the third research seminar on the Hyperbow collaboration entitled "New Tools, New Uses", and are described below by their composers.

4. COMPOSING FOR HYPERBOW

As described above, there have been four public presentations of new works for Hyperbow and cello to date.

During the last two events (Novemeber and December 2005), two new compositions were performed. *Gaia Sketches*, by Patrick Nunn, incorporates the Hyperbow with acoustic cello, while *MODES*, by Artem Vassiliev, features an electric cello. Below, the composers discuss their works.

4.1 *Gaia Sketches*, by Patrick Nunn

4.1.1 Approach

As part of my own research, my intention was to explore ways of extending the timbre of the acoustic cello in a manner that would feel natural to the player without the need to change their existing technique. By applying a method of direct mapping of bow gesture data to control parameters, the incoming audio signal from the acoustic cello could be coloured and transformed.

This particular approach stems from a desire to create a closer relationship between the performance gesture and audio processing. This method of design permits the player to actively feel and control the electronic processing, rather like an extension of his/her own instrument.

The decision to use acoustic cello as opposed to the electric cello was made on aesthetic and conceptual grounds. The greater range of sonorities and resulting range of expression obtained from the acoustic cello provided a wider contrast to the sounds produced after the signal was processed. In addition, the score requires the cellists sound to be localized to their position in performance without the need for separate amplification.

In order to obtain a clean audio signal from the acoustic cello and to eliminate maximum external audio pollution, a Fisherman C-100 cello pickup was attached to the bridge

4.1.2 Concept
Gaia Sketches were inspired by Rachel Rosenthal's poem titled *Gaia mon amour* – a passionate portrayal of humanities infliction upon the Earth and the retaliation of the spirit of Earth through environmental events [13]. The cellist (representing the human spirit) is positioned at the centre point between four surrounding speakers (representing Gaia – Mother Earth).

In the early stages of testing, small composition sketches were written to test the effectiveness of the seven parameters when mapped onto audio processes. These sketches were developed with the addition of feedback delays, reverberations and twelve sequentially triggered samples that act as an accompanying landscape to the seven-minute composition.

Gaia sketches comprises essentially of a series of statements constructed from variants of a four-note motif. Each statement is subjected to transforming timbral states and colourations that explore the interaction between the players bowing gesture and the possibilities inherent in the chosen mapping configurations.

The accompanying samples serve as an additional layer of sonorities that encompass and surround the soloist. A series of sequential modes (triggered by pitch recognition) change the configuration of colouration of the incoming audio signal and is further added to the accompanying samples.

4.1.3 Challenges
The interface for *Gaia Sketches* was programmed in Max/MSP and allows for initial calibration of the bow. Visual references are given to incoming bow data and pitch values of the acoustic cello. Further control is given to mode selection and signal values in and out of the attached audio interface.

The mapping of raw data from the bow often produced inconsistent effects in practice due to natural variations in the cellist's gesture or variations between different performers. A calibration procedure was introduced before each performance which scales the extremes of incoming gesture data to a set of maximum and minimum values.

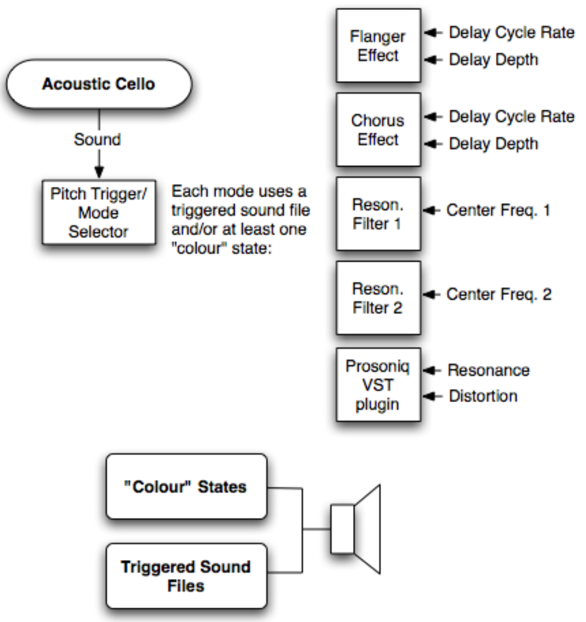


Figure 2: *Gaia Sketches* uses pitch recognition to trigger combinations of "colour" states and stored sound files.

Although the Hyperbow controller offers seven parameters of gesture data, the challenge was to find combinations of bow parameters that were responsive enough for the player to experience effectively for real-time manipulation of audio processing.

Five coloration states transform the incoming audio signal from the acoustic cello through a combination of bow gesture to sound processing parameter mappings. These include two comb filters, two resonant filters and a Prosoniq Northpole VST plugin.

The two comb filters are independently set with different feedforward and feedback coefficients to produce both flanger and chorus type colourations of the signal. The bow velocity, obtained from the rate of change in bow position between frog and tip data, is directly mapped to the rate of the filter. The downward force bow parameter is directly mapped to the depth of the effect.

The first of the resonant filters directly maps the z-acceleration (normal to the string) of the bow to the centre frequency of the filter, exciting the sound during passages involving techniques such as spiccato. The second maps the bow position (obtained from frog and tip values) to the centre frequency of the filter.

The final colouration is achieved through the use of a Prosoniq Northpole VST plugin in which two simultaneous mappings occur. The first directly maps the downward force data to distortion level of the incoming cello signal. The second directly maps bow velocity to the cutoff frequency creating a 'wah-wah' effect.

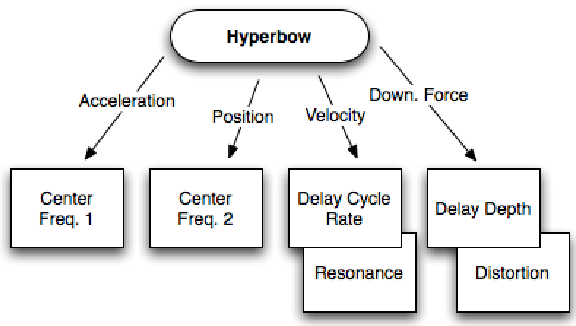


Figure 3: In *Gaia Sketches* the Hyperbow is used to control at least one parameter in each of the "colour" states.

4.1.4 Thoughts on the Outcome of Gaia Sketches
The realization of *Gaia Sketches* can be seen as an initial exploratory step in the collaborative development between designer, composer and musician. Working with new tools such as the Hyperbow generates a unique set of possibilities and challenges both musically and technically for a composer. The most common issue experienced by all composers was one of balance between the processes of learning the required new skills, design, programming, composition and testing.

Perhaps the most exciting of opportunities lies in the potential for creating an intuitive link between the performer and the sound processing through successful mapping strategies. In *Gaia Sketches*, these mappings are independent one-to-one types although the higher- level controls used in the Prosoniq Northpole VST plugin are indeed lower-level parameters that have been cross-coupled. Although these simple mappings were successful in attaining a reasonable amount of intuitive control for the performer, the range of timbral diversity was rather limited and would certainly benefit from further experimentation into more complex mapping strategies.

By far, the most fundamental part of the composition process involved a thorough understanding of the relationship between the gestural data and the perceived sound from the player's instrument. The seven parameters of the Hyperbow, although a little daunting at first, provide a composer with a multitude of gestural information about the performer. Through innovative design of sound synthesis and processing and efficient mapping schemes, it is possible to find suitable combinations that are not only musically interesting but effective in their response.

With the possibility of future performances in mind, one aim was to make the electronic processing as automated as possible. This was achieved through the use of sequential mode selection triggered by pitch recognition using Tristan Jehan's pitch~ object [5]. It is feasible that certain bow parameters could have been used but at the expense of the integrity of the composition. Similarly, the coordination of specific pitches within the composition with the triggering of electronic events can pose further challenges for the composer. However, the pitch tracking method proved to be reliable on most occasions although further improvements are required to ensure complete autonomy.

Future plans include more sketches exploring the interaction and relationships between gesture mapping and sound processing culminating in a larger work for string orchestra and cello with Hyperbow.

4.2 *MODES for electric cello, Hyperbow and computer interaction*, by Artem Vassiliev

As this is my first piece involving live electronic interaction, the proportion between computer-generated musical events and episodes of unaccompanied cello was chosen with the emphasis on the solo instrument. It is a more familiar medium for me as a composer; therefore this approach allowed me to explore the new technology without a danger for this work to become a purely technological study rather than a piece of music. It consists of three sections called *MODES*, 2 interludes, prelude and postlude. Live electronics appear only in *MODES* and their compositional function here can be seen mostly as ornamentation and distorted reflection of the main solo. The musical implication was to create an atmosphere of a meditative self-reflection, for which the timbre of an electric cello was the most productive sound source to explore. (Although initially this piece was intended for an acoustic cello, it was transformed later during our collaboration with Peter Gregson.)

While working on the electronic component of my composition, I aimed to produce a result, which can at the same time sound predictable enough to become a part of a 'stable' composition and flexibly follow all the nuances of the soloist's behavior, which inevitably varies with every new performance, on stage. This was achieved by means of Max/MSP and Logic software with inclusion of two VST plug-ins. The Max patch is a result of my collaborative work with Philippe Kocher, Mike Fabio and Patrick Nunn.

In the MODE 1 the amplitude of incoming cello sound triggers one of the three delay lines (as show in Figure 4), so an echoed version of the original sound is being transformed and distorted in three different ways.

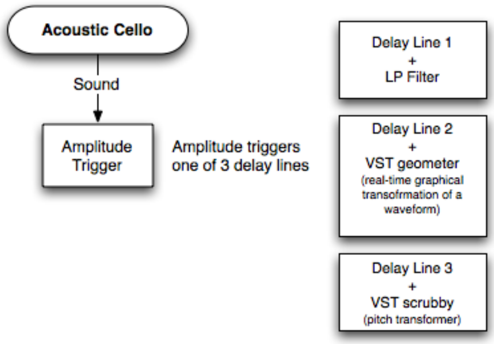


Figure 4: MODE 1 uses a variable delay. The amplitude of the cello sound triggers one of the 3 delay lines.

The next two MODES in this composition require the use of the Hyperbow. Among the multiple possibilities offered by this powerful performer's tool, the one that interested me as a composer was to use it as a flexible trigger and the controller for the electronic accompaniment in this piece. Thus, in the MODE 2 the incoming pitch from the cello is analyzed by the software (using FFT) and replicated with random octave transpositions. The pitches are also mapped according to the three predetermined harmonies. The fast acceleration of the bow switches the programme to the next chord. In order to prevent very rapid changes, the software was set to react on this parameter every five seconds. The tempo of resulting accompanying arpeggiated patterns is regulated by the bow pressure (the stronger the pressure, the slower the tempo). Transposed cello pitches are also doubled by the sine wave synthesizer, which creates an additional layer of texture in a higher register. Its amplitude envelope follows the amplitude of the incoming audio signal from the cello.

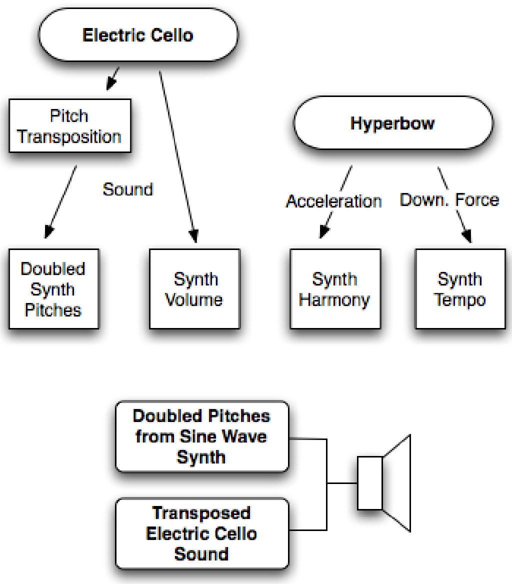
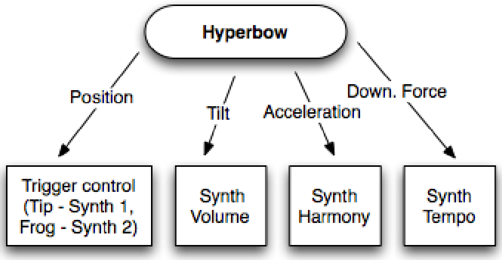


Figure 5: MODE 2 uses a sine wave synth to double transposed pitches produced by the electric cello.

The MODE 3 is similar to the MODE 2, but instead of the transposed cello sound and the sine wave synthesizer, two other physical modeling software synthesizers were used in this section. The performer can switch between them by playing either closer to the tip of the bow or to the frog. The harmony and the tempo are controlled in the same way as in

MODE 2 (fast bow movement will switch the programme to the next chord and the bow pressure determines the tempo).

Although all sound events in the electronic part of my composition are generated in realtime, they can be controlled in order to match harmonically and rhythmically the solo cello part. This flexibility is mainly achieved by use of the Hyperbow. Our practical experience demonstrated, however, that such an approach to the bow as a trigger increases responsibility and demands an additional effort from the performer. This is the one of the problems that I am going to explore further and possibly to solve in my next composition for cello with the Hyperbow and string orchestra.



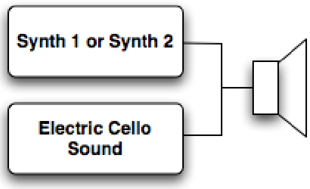


Figure 6: MODE 3 is similar to MODE 2, but uses two external software synths (instead of a sine wave synth).

As my first work for live electronics, *MODES* has achieved its goal. It gave me enough practical experience and inspiration for the new project in which the involvement of live electronics component is going to be more substantial. My first excitement of working with the Hyperbow is now leading me towards serious exploration in a broader area of compositional interests and research activities. Thus, in the new project I am going to try using only those parameters that will not interfere with interpretive ideas of the performer. In other words, I see the Hyperbow in my next piece more as a controller rather than a trigger. The ideal situation for such a performance would be when the complexity and variety of functions of the Hyperbow will become almost unnoticeable for the player, but at the same time all the events in the electronic part will be accurately synchronized with the solo instrument and the orchestra, leaving enough space for an interpretation and for an improvisation if it is required.

Apart from offering endless opportunities for an experiment, the Hyperbow is continuing to develop over the period of our ongoing collaboration. I see this fact as a great advantage for me as a composer. It motivates me to continue working on this project because certain improvements in software and hardware components can be made even according to the needs of one individual composition. Thus, the composer can be as precise as possible in his/her compositional demands. On the other hand, every new composition written specifically for the Hyperbow extends its repertoire and potentially increases chances for this outstanding invention to become one day an industrial standard in contemporary music making and performance practice.

5. PERFORMING WITH THE HYPERBOW

In this project, we are extremely fortunate to have the participation of highly skilled, classically trained, adventurous cellists. To date, four such players, Philip Sheppard, Shu-Wei-Tseng, Alexander Holladay, and Peter Gregson, have performed the various new works with Hyperbow composed throughout this collaboration.

Figure 7: Shu-Wei Tseng with Hyperbow and cello (© Roberto Aimi).

Not surprisingly, the expectations, impressions, and experiences of each individual cellist regarding the Hyperbow vary considerably. Previous experiences, such as whether or not the player has performed with an electric or amplified acoustic cello before, and the degree of familiarity with studio equipment, computers, or technology in general, are important factors. Also, the amount of time spent with the Hyperbow, in collaboration with the composers and in rehearsal, is of course critical.

Both Shu-Wei Tseng and Peter Gregson, who recently performed *Gaia Sketches* and *MODES*, respectively, took a keen interest in the technical capabilities of the Hyperbow. However, in reaction to various aspects of the experience, their impressions varied. Tseng remarks, “I was immediately put into a position where I have to be fully aware of what and how I do things. This is wonderful…” Gregson observes that producing the gesture data to produce the desired sound “came down simply to feel.” Of course, their perceived ability to control the sound output of the system was also dependent on the mapping in place.

On the issue of the weight and ergonomics of the Hyperbow, feedback also differed. One player stated that the increased weight of the Hyperbow was of no concern, while another disagreed. Interestingly, criticisms regarding the carbon fiber bow itself were also expressed, as Tseng observed that it produced sound louder than usual, and Gregson noted that he found the frog to be too low.

Though the number of cellists who have played with the Hyperbow is still quite small, the feedback gained from them has been extremely helpful. Perhaps the most encouraging piece of feedback received thus far was the suggestion by two of the cellists to conduct a training session on the maintenance and upkeep of the Hyperbow interface for the performing cellists in the group. Such a session would include instructions on when and how to change batteries, remove the electronics board to enable rehairing of the bow, debug potential problems in the operation of the electronics, etc. Imparting these skills to Hyperbow performers is essential to ensure the success of the interface, as we hope that it will one day be entirely under the care of its users and truly “gig-worthy”.

Though the number of performers who have played the Hyperbow is still quite small, the benefits we have gained from their contributions are great. As our collaboration grows and the Hyperbow develops, we will continue to enlist the essential help of these and other performers.

6. SUMMARY

We are greatly encouraged by the progress we have achieved in the past year of our Hyperbow collaboration. The successful deployment of the Hyperbow within a new community of users, performances of new compositions, and the enthusiasm of both composers and performers for this work of art and research, are all positive results.

As we continue, we plan to further increase the number of participants (composers and cellists) involved in the project, in order to create a greater repertoire for the Hyperbow and a larger body of knowledge regarding its performance applications. It is our hope that through such work we may soon be able to produce a significantly improved and refined Hyperbow that will be useful to many other musicians.

7. ACKNOWLEDGMENTS

Our thanks to Tod Machover, Simon Bainbridge, and Philip Sheppard, Mike Fabio, Philippe Kocher, and Alexander Holladay. Special thanks to Peter Gregson and Shu-Wei Tseng for the comments included above and to Roberto Aimi for providing the photographs.

8. REFERENCES

[1] Burtner, M. The Metasaxophone: Design of a New Computer Music Controller. . In *Proceedings of the 2002 International Computer Music Conference (ICMC)*. Göteborg, 2002.

[2] Chafe, C. *http://ccrma.stanford.edu/~cc/*.

[3] Goto, S. The Aesthetics and Technological Aspects of Virtual Musical Instruments: The Case of the SuperPolm MIDI Violin. *Leonardo Music Journal*, 9:115–120, 1999.

[4] Goudeseune, C. A. *A Violin Controller for Real-Time Audio Synthesis*. Tech. Rep., Integrated Systems Laboratoy, University of Illinois at Urbana-Champaign, 2001.

[5] Jehan, T. *http://web.media.mit.edu/~tristan/*.

[6] Machover, T. and Chung, J. Hyperinstruments: Musically Intelligent and Interactive Performance and Creativity Systems. In *Proceedings of the 1989 International Computer Music Conference (ICMC)*. San Francisco, 1989.

[7] Nichols, C. *The Vbow: An Expressive Musical Controller Haptic Human-Computer Interface*. PhD Thesis, Stanford University, 2003.

[8] Overholt, D. The Overtone Violin. In *Proceedings of the New Interfaces for Musical Expression (NIME) Conference*. Vancouver, 2005.

[9] Palacio-Quintin, Cléo. The Hyper-Flute. In *Proceedings of the New Interfaces for Musical Expression (NIME) Conference*. Montreal, 2003.

[10] Paradiso, J. A. and Gershenfeld, N. A. Musical Applications of Electric Field Sensing. *Computer Music Journal*, vol. 21, no. 3, pp. 69-89, 1997. .

[11] Poepel, C. Synthesized Strings for String Players. In *Proceedings of the New Interfaces for Musical Expression (NIME) Conference*. Hamamatsu, 2004.

[12] Rose, J. http://jonroseweb.com/.

[13] Rosenthal, R. and Chaudhuri, U. *Rachel's Brain and Other Storms: The Performance Scripts of Rachel Rosenthal*. Great Britain: Continuum International Publishing Group, 2001.

[14] Scavone, G. The Pipe: Explorations With Breath Control. In *Proceedings of the New Interfaces for Musical Expression (NIME) Conference*. Montreal, 2003.

[15] Schoner, B. *Probabilistic Characterization and Synthesis of Complex Driven Systems*. PhD Dissertation, MIT Media Laboratory, Cambridge, MA, 2000.

[16] Tarabella, L. and Bertini, G. Giving Expression to Multimedia Performance. In *Proceedings of the 2000 ACM Workshops on Multimedia*. Los Angeles, 2000.

[17] Trueman, D. and Cook, P. BoSSA: The Deconstructed Violin Reconstructed. In *Proceedings of the 1999 International Computer Music Conference (ICMC)*. Beijing, 1999.

[18] Waisvisz, M. The Hands, A Set of Remote MIDI-Controllers. In *Proceedings of the 1985 International Computer Music Conference (ICMC)*. San Francisco, 1985.

[19] Young, D. The Hyperbow Controller: Real-Time Dynamics Measurement of Violin Performance. *Proceedings of the New Interfaces for Musical Expression (NIME) Conference*. Dublin, 2002.

[20] Young, D. *New Frontiers of Expression Through Real-Time Dynamics Measurement of Violin Bows*. Master's Thesis, MIT Media Laboratory, Cambridge, MA, 2001

The augmented violin project: research, composition and performance report

Frédéric Bevilacqua, Nicolas Rasamimanana, Emmanuel Fléty, Serge Lemouton and Florence Baschet

IRCAM - Centre Pompidou
CNRS STMS UMR 9912
Gesture working group
1 place Igor Stravinsky
75004 Paris – France
+ 33 1 44 78 48 31

{frederic.bevilacqua, nicolas.rasamimanana, emmanuel.flety, lemouton}@ircam.fr, fbaschet@noos.fr

ABSTRACT

In this paper we present the augmented violin developed at IRCAM. This instrument is an acoustic violin with added sensing capabilities to measure the bow acceleration in realtime. We explain first the approach we developed to characterize bowing styles. Second, we describe the realtime implementation of the bowing style recognition system. Finally we describe an electro-acoustic music composition, *Bogenlied*, written for the augmented violin.

Keywords

Augmented violin, hyper-instrument, bowing styles, mapping.

1. INTRODUCTION

The augmented violin project started in 2003 at IRCAM following the interest of several composers to use violin gestures for the control of electronic processes. This project triggered the creation of an interdisciplinary working group on gesture analysis and musical interfaces. One research goal of this group is to work on the concept of "augmented instruments", i.e. acoustic instruments with added gesture sensing capabilities, which is similar to the *hyperinstruments* pioneered at MIT. We believe that such an approach is particularly fruitful for both fundamental gesture research and artistic endeavors.

Different approaches are possible with "augmented instruments". First, sensors can be utilized to add control possibilities that are not directly related to normal playing techniques. For example, various buttons can be added to the body of the instruments. In such a case, the use of sensors implies new gestures for the player. Second, sensors can be applied to capture normal playing gestures. This paper is related to this type of approach, which poses a fundamental question: to what extend can an instrumental gesture, mastered in a particular context, be used in another (or larger) context of musical expression? Such a questioning is actually very fruitful, and remains valid for any types of musical interfaces.

This paper focuses on the ongoing "augmented violin" project. We report here first a study on bowing styles characterization, and the implementation of a system performing realtime bowing style recognition. Second, we describe a music composition, *BogenLied* by Florence Baschet[1] written for the augmented violin and utilizing the analysis system.

2. RELATED WORKS

Different systems have been developed to directly or indirectly measure the violin "gesture parameters". Several systems have been developed over the years using various sensing systems [4][10][9][3]. An alternative approach is to use audio features as a trace of the performed gestures [6][2].

In most of the works, gesture data has been used as a direct control of sound filters, or as the input for physical model synthesis [8]. Fewer works have reported on interpreting the gesture data, in order to provide high-level parameters used in the mapping design. Bernd Schoner [7] used the gesture input to statistically estimate the corresponding sound features in order to drive more expressive synthesis. Chad Peiper [5] used decision trees to classify different bowing styles.

3. APPROACH

Augmented instruments are based on traditional instruments, which gestures are *a priori* defined. In the case of the violin, the different types of bow strokes form a widely accepted and formalized set of gestures. Composition for strings includes bow strokes indications such as *détaché*, *martelé*, *ponticello,* etc. On a finer level, the mastering of various bowing articulations is part of the player's skill.

Our approach has been to use this gesture "vocabulary" as a starting point to build the interaction between the player and the electronics. This is somewhat similar to the approaches of Schoner or Peiper, since we propose to build an "interpretation level" from the data stream, in order to facilitate the mapping between gestures and sounds.

Our first goal was hence to study the relationships between bowing styles and the various sensors data. Such studies allowed us then to build real-time systems that can "interpret"

[1] Performed at Dijon, France, Nov. 26 2005 Whynote Festival (Premiere), Violin: Anne Mercier (Ensemble Itinéraire); Jan. 07 2006 Maison de la Radio, Paris, France.

the low-level captured parameters into high-level parameters related to bowing styles.

This approach was found in resonance with the compositional approach of Florence Baschet: "With *BogenLied*, my aim as a composer was to focus my attention to the fine instrumentalist gestures, and to use such gesture articulations as the input of an interactive system. Precisely, *BogenLied* is an attempt to create a mixed –acoustic and electronic- sound space, resulting from a sensitive interactive relationship between the soloist and electro-acoustic system. Such interaction would be ideally similar to the type of close relationship between two musicians, as typically found in chamber music".

4. THE AUGMENTED VIOLIN

Two prototypes have been successively built. They are described below.

4.1 Prototype I

Our first prototype of the wireless bow measurement system is composed of a small electronic board with a microcontroller, two ADXL202 accelerometers from Analog Devices and a digital radio transmitter (fig. 1). The principle of the measurements is similar to the techniques developed by Joe Paradiso and Diana Young [4][10]. Our first prototype utilizes a special radio transmitter that enables collision detection on the carrier and therefore permits to share the transmission bandwidth with other bows equipped similarly. The bow module was very satisfactory. The only drawback is its relative big size due to the thickness of the radio transmitter and the batteries.

4.1.1 Bow acceleration

Accelerations are not analogously sampled but measured using PWM capture with accurate counters and timers. This technique delivers acceleration values on a range of 26000 points, which is significantly better than the use of the internal 10 bit ADC of the microcontroller. A wireless receiver is placed on an Ethersense [1] daughter board. The Ethersense sends the digital data from the accelerometers (as well as others signals) to a host computer through Open Sound Control (OSC).

4.1.2 Bow position

The bow position extraction is directly inspired from the electric field sensing measurement described by Paradiso: two different signals are emitted from the bow tip and frog. A capacitive coupling plate placed behind the bridge collects the mix of the two signals, "tip" and "frog", which relative intensities depend on the bow position [4]. An Ethersense [1] daughter board demodulates the mix of bow positioning signals. The material used to make the resistive strip placed on the bow was taken from a S-VHS tape. This tape features a homogenously distributed electrical resistance, over its whole length. Two strips of the tape were glued one on the other to protect the resistive side and to make the overall resistance in a more adequate range.

A software calibration was developed to compute the bow position based on the measurement of the two signals "tip" and "frog". The calibration was based on careful measurements for a large set of different bow positions. However, our effort to perform accurate and reliable bow position measurement was deceptive. The main difficulty with this technique is due to the fact that the two measured signals are not sufficient to determine without ambiguity the bow position. This problem especially occurs at the bow extremities where the surface of the "bow plate" diminishes. In addition, the right hand causes a drastic modification of the signals as the impedance of the body interferes with the system, causing a decrease in the signals intensities. An additional difficulty comes from bowing techniques implying significant variations of bow angle, which affects the coupling capacity between the plates.

Overall, the distance measurement was found useful only for qualitative measurement, but problematic for accurate position measurement. Other methods for position measurement are currently experimented.

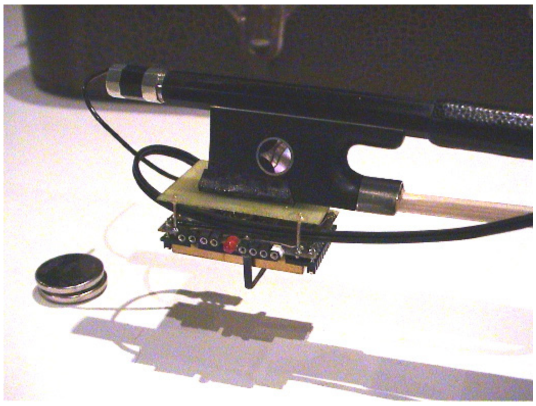


Figure 1: Prototype I.

4.2 Prototype II

A second prototype was built using a subminiature (but single frequency) radio transmitter. The custom-made batteries holder and the second accelerometer were mounted on an extension of the PCB placed on the left side of the bow frog (fig. 2). This second prototype is smaller and lighter than the prototype I (total weight of 17g). The overall thickness of this device was significantly reduced, limiting therefore the risk of scratching the violin. Soft foam was added to totally suppress any possibility of such an accident (not shown on fig 2). Moreover, it consumes less current, allowing 1h30 of continuous playing.

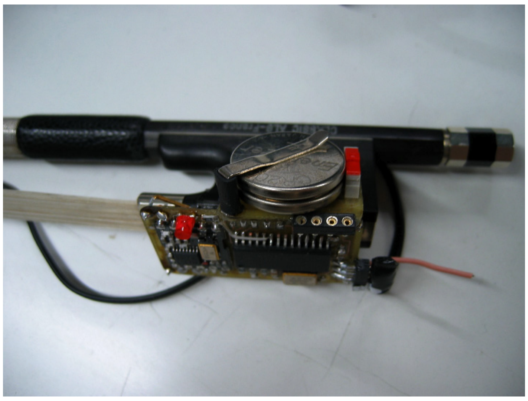


Figure 2: prototype II

5. BOW STROKE STUDIES

We focused our first studies on the following bowing styles: *détaché, martelé* and *spiccato*. After recording these bow strokes in various musical contexts (scales, musical phrases) and with different players, a complete offline analysis has been performed. This analysis has been reported in reference [11] and we summarize here only the important points.

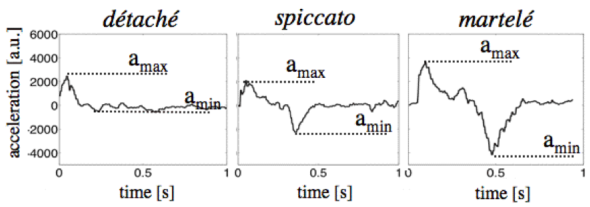


Figure 3. Typical acceleration curves for *détaché*, *martelé* and *spiccato* (from left to right respectively)

- The acceleration curve in the bowing direction shows typically two principal peaks, one positive, a_{max} and the second one negative, a_{min} (for upbow, see fig.3). The *détaché* stroke can be seen as a degenerated case where the second peak is close to zero.

- The use of two parameters a_{max} and a_{min} allows for the clustering of these bowing styles. Recognition rate using the standard *knn* method (k-nearest neighbors) leads to high recognition rates.

- The parameter $(a_{max})^2 + (a_{min})^2$ can be interpreted as a bow stroke "intensity". This parameter is correlated to the energy given to the bow by the musician. It is also correlated to dynamics when the bow is in contact with the strings. For bouncing bowing styles, such as *spiccato*, this parameter gives an indication of the gesture "intensity" rather than the loudness of the sound.

The first study was performed with two violin players, amateur and professional, and showed very consistent results. A second study was performed with a class of 12 students. The results demonstrated than the characterization we proposed remained valid over this larger set of players: the three bow strokes can still be clustered with a_{min} and a_{max}. However, idiosyncratic behaviors were also found, showing that a universal calibration might not be reliable. Interestingly, these results also showed us that this type of analysis could potentially be useful for pedagogical applications. For example, the bowing characterization could provide the students with a complementary feedback to improve bow regularity and/or dynamics. Such potential applications for pedagogy are currently studied.

6. REALTIME BOWSTROKE ANALYSIS

A real time implementation of the bow stroke characterization described in the previous section was implemented in Max/MSP, using several objects of the library MnM [12], dedicated to gesture analysis. The different steps of the gesture analysis are described in Fig 4.

A median filter is used on the x-acceleration data (main bow axis). The dataflow is then split into two different processes run in parallel.

First, a parameter related to the "intensity" of the bowing is computed from the acceleration curve. Precisely, the maximum of the absolute value of the acceleration, computed on a sliding window, is output.

Second, a distinct process allows for the segmentation and characterization of the bow strokes. The varying baseline due to the coupling between angle and acceleration is first removed, (assuming a linear offset). The segmentation is performed then in two steps. First, accelerometer peaks are determined on a sliding window. Second, a procedure sorts and labels the various peaks. One of the difficulties resides in the fact that the "maximum" acceleration a_{max} alternates between bow changes:

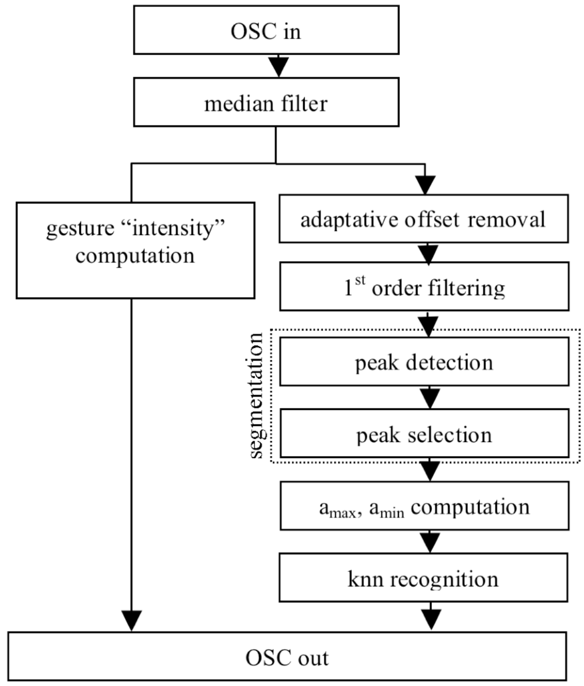


Figure 4. Schematic of the realtime gesture processing

in upbows a_{max} appears as a local maximum in the raw signal while it appears as a local minimum for downbows. Our selection and labeling procedure is based on the timing between the different peaks. Precisely, we used the fact that the time difference between a_{max} and a_{min} is typically smaller than 100 ms.

This segmenting procedure is fundamentally underdetermined for a sequence of fast notes, such as a tremolo (when using the accelerometer signal only). In such a case, the peak labeling procedure is ambiguous; different bow strokes might give rise to very similar acceleration curves. Such a problem could be solved by the combination of both audio and acceleration data.

Once the segmentation is performed, the associated a_{max} and a_{min} parameters of each bow stroke are computed. Fig 5 shows an example (as displayed in Max/MSP) of clustered points obtained in the a_{max} and a_{min} plane, with *détaché, martelé* and *spiccato* (playing scales).

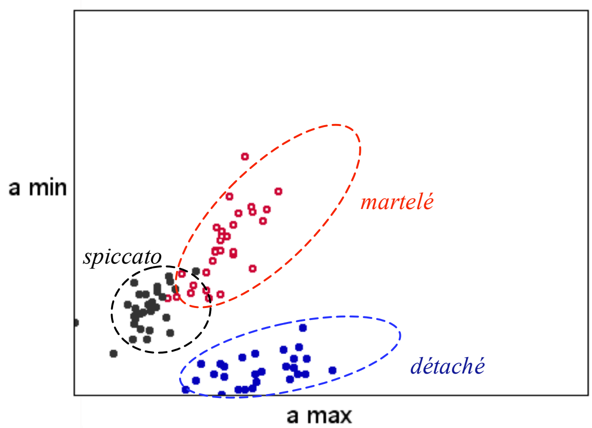


Figure 5. LCD object from Max/MSP showing the clustering on the three bow strokes (real-time): *détaché* (blue), *martelé* (red), *spiccato* (black).

Finally, a *knn* algorithm (k-nearest neighbors) is run to classify each bow stroke as *détaché*, *martelé* or *spiccato*, according to a previously recorded bow strokes database. Weights related to bowing styles are computed using the number of nearest neighbors of each class. The weight averages over several notes are also computed.

The final results, i.e. gesture intensity and bow stroke weights are sent through Open Sound Control (OSC) to the computer controlling the sound processes.

7. COMPOSITION WORK: *BOGENLIED*

7.1 Setup

The diagram (figure 6) shows the configuration used in the performance setting. Two separate computers (two Macintosh G4 PowerBooks) are running Max/MSP, and communicating using OSC. The "sound processing" computer generates the digital sound environment using the live violin sound, captured by microphone. Various parameters of the sound processing are controlled by the gesture data transmitted from the "gesture-processing" computer. The electronic sound is spatialized (using the "Ircam spat") and rendered by a hexaphonic sound diffusion system.

7.2 *Bogenlied* form.

As shown in fig. 7, the musical form of BogenLied is a simple

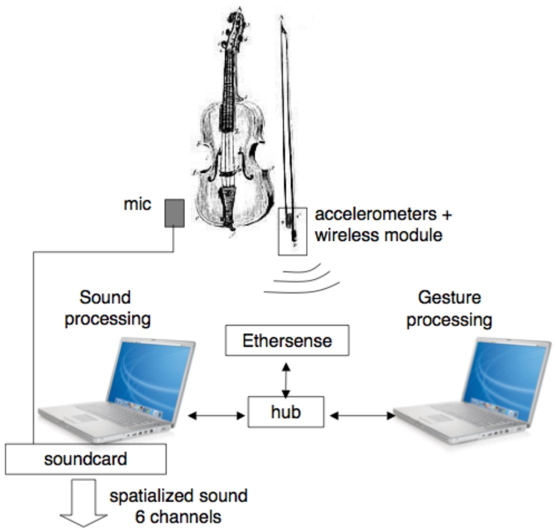


Figure 6. Setup for the piece *Bogenlied*

linear form divided into nine sections, alternatively with and without electronics. In section *II*, *IV* and *VI*, the musical writing is focused on specific articulations of bow strokes: *détaché*, *martelé*, *spiccato*. Each of these sections is associated to specific compositional materials, as well as to specific electronic sound processes (described in section 7.3).

In section *VIII*, the three bowing styles appear successively, along with their associated compositional material. In this section, the choice of the electronic processes is driven by the recognition system.

In last section (*IX*), the performer plays hybrid bow strokes, which sequence appears as a series of "mutations" from one articulation to another one. As described further below, the electronic sound is then built as a combination of the electronic processes performed in section *II*, *IV* and *VI*.

section	bowing style	sound	Gesture analysis
I		acoustic	
II	*détaché*	mixed	gesture intensity
III		acoustic	
IV	*martelé*	mixed	gesture intensity
V		acoustic	
VI	*spiccato*	mixed	gesture intensity
VII		acoustic	
VIII	*all 3 bowing styles*	mixed	gesture intensity + bow stroke recognition
IX	*hybrid*	mixed	gesture intensity + bow stroke recognition

Figure 7. Structure of *Bogenlied*

7.3 Audio processing

The Max/MSP sound patch contains two granular synthesis modules and several effects. The first granulator is used to produce drones (pedal-notes) triggered by the bow strokes. A separate drone is linked to each bowing styles: G, A, and E for *détaché*, *martelé* and *spiccato*, respectively.

The second granulator is used for real-time processing of the violin sound. The audio signal from the microphone is stored in a 5 second circular buffer~, and 100 to 200 ms grains are played back from this buffer. The granular synthesis sound is processed through several standard effects: frequency-shifter, harmonizer, filter and vocoder. Different combinations and parameterizations of these effects are used in each section. Some parameters are controlled in realtime by the gesture data, as explained next.

7.4 Gesture Mapping

The mapping between the gesture data and sound processing parameters is built during the piece as a gradual superimposition of three different mapping modalities:

a) *Continuous* mapping, applied indifferently to all bow strokes

- sections *II* , *VI*: the gesture "intensity" is linearly mapped to the grain density (number of grains played simultaneously)
- section *IV:* the gesture "intensity" is mapped to the grain transposition spread (small gestures induce small or no transposition, strong gestures cause all the grains to be randomly transposed around the played pitch).

b) S*elective* mapping using bow stroke recognition:

- sections *VIII* and *IX*: bow stroke recognition is used to select the type of sound processing. The recognition of a *détaché, martelé or spiccato* recalls the presets of section *II*, *IV* or *VI,* respectively (including the corresponding drones).

c) Mapping *mixing*, using bow stroke characterization (of hybrid bow strokes):

- section *IX*, the performer plays hybrid bow strokes. The gesture analysis computes for a given bowstroke three weights, corresponding to "likelihood" to be related to the different bowing styles (in other words a given bow stroke is considered as a mix between different bowing styles). These weights are then used to control the mixing levels of the drones associated with each bowing style. Each bow stroke is thus colored by the combination of these three pitches reflecting continuously the quality of the articulation.

8. DISCUSSION AND PERSPECTIVES

We report here developments related to our current "augmented violin" project. From the technological point of view, our prototypes were tested on several experiments and performances and found to be robust. The added weight on the bow seems manageable for the players: the various professional violinists we worked with agreed to play with such a constraint.

We described the real-time implementation of a bow stroke analysis framework that was reported previously [11]. This approach requires the segmentation of the acceleration data stream in separate bow strokes. Such a task is difficult when the acceleration signal alone is used, due to various artifacts. The algorithm we designed was generally satisfactory, but was not applicable to fast notes, such as tremolo. We are currently investigating other approaches, based for example on Hidden Markov Models, in order to overcome such limitations.

BogenLied is the first piece written for the IRCAM augmented violin. It is worth to note that this piece (composed by Florence Baschet) has been developed in the context of an interdisciplinary workgroup. The piece takes advantage of the gesture analysis we described. In particular, the mapping is expressed from elements of the musical language, in order to create an electronic environment that performers can apprehend intuitively. Our collaboration with the violinist Anne Mercier confirmed us that such an approach is very effective.

The real-time analysis was reliably used in performances of *BogenLied*. Overall, the recognition system was satisfactory. In particular, the performer was able to control the mapping easily with her own gestures, and she could feel that the system was reacting well to her gesture. A promising point was the possibility to characterize "hybrid" bow strokes by continuous parameters. This feature seems to offer pertinent information in complex musical phrases.

In the near future, other performers will experiment with this system, and we are expecting interesting comparisons. In particular, the modules will be used with other string instruments such as the viola, cello and double bass. This collaborative work has provoked a high interest at IRCAM and several other artistic works are currently in progress.

9. ACKNOWLEDGMENTS

We would like to thank Anne Mercier, Alain Terrier and Jean-Loup Graton for their precious contribution and help.

The I-MAESTRO project is partially supported by the European Community under the Information Society Technologies (IST) priority of the 6th Framework Programme for R&D (IST-026883, www.i-maestro.net, www.i-maestro.org). Thanks to all I-MAESTRO project partners and participants, for their interests, contributions and collaborations.

10. REFERENCES

[1] E. Fléty, N. Leroy, J.-C. Ravarini, and F. Bevilacqua. Versatile sensor acquisition system utilizing network technology. *In Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2004.

[2] C. Goudeseune, G. Garnett, and T. Johnson. An interface for real-time classification of articulations produced by violin bowing. *In Proceedings of the Conference on Human Factors in Computing Systems*, 2001.

[3] D. Overholt. The overtone violin. *In Proceedings of the Conference on New Instruments for Musical Expression, NIME,* 2005.

[4] J. A. Paradiso and N. Gershenfeld. Musical applications of electric field sensing. *Computer Music Journal*, 21(2), 1997.

[5] C. Peiper, D. Warden, and G. Garnett. An interface for real-time classification of articulations produced by violin bowing. *In Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2003.

[6] C. Poepel. Synthesized strings for string players. *In Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2004.

[7] B. Schoner, C. Cooper, C. Douglas, and N. Gershenfeld. Cluster-weighted sampling for synthesis and cross-synthesis of violin family instrument. *In Proceedings of the International Computer Music Conference, ICMC*, 2000.

[8] S. Serafin and D. Young. Bowed string physical model validation through use of a bow controller and examination of bow strokes. *In Proceedings of the Stockholm Music Acoustics Conference, SMAC*, 2003.

[9] D. Trueman and P. R. Cook. Bossa: The deconstructed violin reconstructed. *In Proceedings of the International Computer Music Conference, ICMC,* 1999.

[10] D. Young. The hyperbow controller: Real-time dynamics measurement of violin performance. *In Proceedings of the Conference on New Instruments for Musical Expression, NIME*, Dublin, Ireland, 2002.

[11] N. Rasamimanana, E. Fléty, F. Bevilacqua, Gesture Analysis of Violin Bow Strokes, GW 2005, Lecture Notes in Artificial Intelligence 3881, S. Gibet, N. Courty, and J.-F. Kamp (Eds.): pp. 145–155, 2006.

[12] F. Bevilacqua, R. Muller, N. Schnell, MnM: a Max/MSP mapping toolbox. *In Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*, *NIME*, Vancouver, p. 85-88, 2005

Auditory Illusion and Violin: Demonstration of a Work by Jean-Claude Risset Written for Mari Kimura

Mari Kimura
The Juilliard School
60 Lincoln Center Plaza
New York, NY 10023 USA
mkimura@juilliard.edu,
mari.kimura@nyu.edu

Jean-Claude Risset
Laboratoire de Mécanique et d'Acoustique, CNRS
31 chemin Joseph Aiguier
13402 Marseille cedex 20, FRANCE
+ 33 4 91 16 40 78
jcrisset@lma.cnrs-mrs.fr

ABSTRACT

This is a description of a demonstration, regarding the use of auditory illusions and psycho-acoustic phenomenon used in the interactive work of Jean-Claude Risset, written for violinist Mari Kimura.

Keywords

Violin, psycho-acoustic phenomena, auditory illusions, signal processing, subharmonics, Risset, Kimura.

1. INTRODUCTION

In 1995, composer Jean-Claude Risset collaborated with violinist Mari Kimura to write *Variants* for violin and signal processing. He used signal processors as well as Kimura's extended technique subharmonics. This bowing technique enables one to play pitches below the open G on the violin without changing the tuning of the instrument, and Edward Rothstein of the New York Times called it 'revolutionary' [5]. From this 1995 collaboration emerges this year a new collaboration aimed at creating a work for violin, featuring more psycho-acoustic phenomena and acoustic illusions. The demonstration at NIME2006, Paris, in June 2006 will show excerpts of this work and the concepts behind the collaboration. This paper describes the demonstration and introduces the various techniques used in the piece.

2. *VARIANTS* (1995)

2.1 Collaboration History

Originally, Kimura approached Risset in 1995 with the idea of a piece featuring acoustic illusions,' which Risset wrote about in the 1970s [1, 2]. In the program notes for *Variants*, for violin and digital processing (1995), Risset writes: "[*Variants*] is the first state of a work dedicated to Mari Kimura. The title refers to the transformations of violin sounds produced by digital processing, but also to certain variation processes within the violin part. For instance, the timing intervals of melodic groups, causing so-called stream segregation, are echoed as mere rhythms. Digital echoes and reverberation build up a contrapuntal and harmonic fabric which extends the violin melodies." At the time, Kimura did not have access to the MaxMSP program and the work was realized using an Ensoniq DP/4 parallel effect digital signal processor. The first line of the score in shown in Figure 1. A few years after *Variants* was written, Kimura converted all the processes of *Variants* into a MaxMSP program, to be performed interactively with computer.

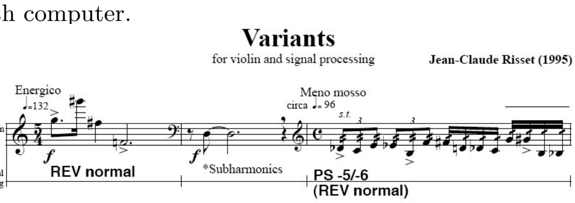


Figure 1: *Variants* by Jean-Claude Risset.

2.2 *Variants* and Subharmonics

In *Variants*, Risset also uses *subharmonics* [3, 4]. In this and in other works by Kimura, the resulting acoustic phenomenon is used as a compositional tool. Risset is the first composer to incorporate Kimura's bowing technique in works for violin and signal processing, aside from Kimura in her own works.

3. AUDITORY AND PHYSICAL ILLUSIONS

3.1 Physical Illusion and Sound

Occasionally, some effects in Kimura's repertoire (which she uses in both her compositions and her improvisations) can leave the performer somewhat disoriented, both physically and auditorily. Figure 2 shows such an example actually used by Kimura in her performances: two fingers on two strings are playing in glissando in opposite directions, up and down, switching fingers in the middle and joining in unison. This is a rather subjective example, but it illustrates especially well that towards the narrow intervals between two pitches, the performer could get a little disoriented about which finger is doing what exactly. The disorientation is more pronounced as this is played at faster speeds. As a performer, this experience is interesting as it might be akin to trying to hold your hands in your back, one arm from above and another from below, then switching hands rapidly to do the same.

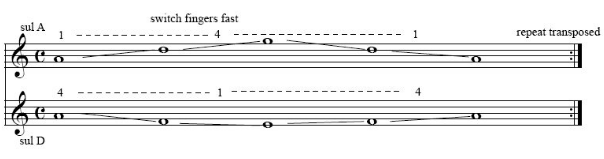


Figure 2: Fast *glissando* on two strings in opposite directions.

3.2 Auditory Illusions

Composers throughout the ages have been using auditory effects and incorporating them in their musical compositions. There are many examples, but a common one well-known to violinists is Tchaikovsky's *Violin Concerto*, which uses the pitch ranges of all instruments to highlight the soloist. Despite the thick orchestration of this romantic era concerto, the solo violin maintains dominance over the entire orchestra throughout the piece. One of the reasons is that Tchaikovsky never lets the highest-pitched instrument in the orchestra, namely the flute, to go above the violin for more than 4 measures at most. Every time the flute goes higher beyond the solo violin, never more than 4 measures later the solo violin takes over the highest note in the entire orchestra.

Figure 3: Excerpt from Tchaikovsky's *Violin Concerto*.

In a more recent example, Mario Davidovsky's *Synchronisms No.9* for Violin (1988) uses auditory illusions between violin and electronics both on performer's and listener's ears. Davidovsky, himself a violinist by training, is fully aware that the higher the note you play on the violin (above 1000 Hz B on two octaves above middle C), the more ambiguous the instrumental timbers become to the human ear. He combines seemingly innocent sine-wave like chords on the tape part, weaving the violin in the middle. This actually does affect the performer's auditory center, making the performer slightly disoriented auditorily to what he is actually playing, as the tape part is quite close to the timber of the violin itself (Figure 4).

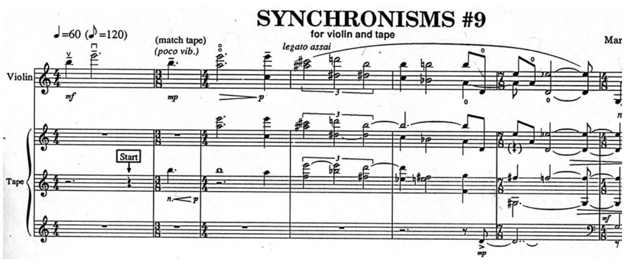


Figure 4: Excerpt from *Synchronisms No.9* for Violin and Tape by Mario Davidovksy (Courtesy of Edition Peters).

4. RISSET'S *VIOLONÉ* (2006)

4.1 Concept

The word 'Violoné' was used mostly to characterize the Louis XV style of furniture, with curved shapes similar to those of a violin. Risset would use the violin to generate structures giving rise to illusions or to certain processes - but these structures would be kin to the violin, since the violin provided their material.

4.2 Devices

In *Violoné*, Risset uses real-time MaxMSP processing as the device that creates auditory illusions with the violin. Some of the auditory illusions described in Risset's past research will be incorporated in this musical creation. Risset and Kimura develop a musical language that derives from the acoustic characteristics of the violin (and of course, not so characteristic as Kimura's *subharmonics*) combined with such electronic effects, creating auditory illusions both for the performer and the listener. It is a challenge for performers to be auditorily disoriented while performing, at the same time it is also a very intriguing prospect as one of the performance practices of electronic music for the violin. Compositionally speaking, several auditory illusions have been demonstrated thanks to computer sound synthesis, which afford a precise control of the physical structure of the synthetic sound; but some can be realized with instruments, granted a careful control. The control by the virtuoso instrumentalist can yield unexpected effects. This was already demonstrated in works from Bach to Ligeti, in Risset's orchestral works *Phases* and *Escalas*, and Kimura's *subharmonics*.

4.3 Methods

One of the methods used in realizing such auditory illusions will be the way Risset initially harmonizes the violin using MaxMSP, then gradually this harmonization turns into octaviations, so that the descents or ascents on the violin turn into illusory endless ascents or descents. The instrumental sound can also be taken as a point of departure for subsequent digital sound processing, so as to impose specific structures that are conducive to creating sonic illusions. For instance, going down a scale can be turned into an infinite descent, using either harmonization or delay and multi-play.

5. CONCLUSION AND FUTURE PLANS

Kimura's extended technique *subharmonics* for the violin has been developed not as a mere novelty but as a musical necessity, stemming from the desire to expand the musical language of the violin. One should also be aware that illusions are not mere curiosities: they reveal the inner processes of our auditory perception. As Purkinje stated, *illusions, errors of the senses, are truths of perception.*
Risset and Kimura have been considering collaborating with visual artists, in order to combine both auditory and visual illusion to create *Violoné* as interactive audio visual work. The parallel seen between Risset's auditory illusion concepts and visual illusions seen in the works of Victor Vasarely could easily be combined. By June 2006, we plan on having more concrete realization plans along these lines.

6. REFERENCES

[1] J. C. Risset, 1971. Paradoxes de hauteur. *Actes du 7e Congrès International d'Acoustique*, Budapest, S10, pp. 613-616.

[2] D. L. Wessel and J. C. Risset, 1979. Les illusions auditives. *Universalia (Encyclopedia Universalis)*, pp. 167-171.

[3] M. Kimura, 1995. Subharmonics: an extended technique for the violin. Invited author, *J. Acoustical Society of America*, Vol. 97, No. 5, Pt. 2, pp. 3270. Acoustical Society of America, Washington D.C.

[4] M. Kimura, 2001. The world below G. *Strings*, No. 96, August/September 2001, pp. 24-29.

[5] E. Rothstein, 1994. Review/Recital: A violinist tests limits in music of her time. *The New York Times*, April 21, 1994.

Augmenting the Cello

Adrian Freed
David Wessel
Michael Zbyszynski
CNMAT, UC Berkeley
1750 Arch St., Berkeley, CA94709
1(510)643 9990
{adrian,wessel,mzed}@cnmat.berkeley.edu

Frances Marie Uitti
francesmarieuitti@yahoo.com

ABSTRACT
Software and hardware enhancements to an electric 6-string cello are described with a focus on a new mechanical tuning device, a novel rotary sensor for bow interaction and control strategies to leverage a suite of polyphonic sound processing effects.

Keywords
Cello, chordophone, FSR, Rotary Absolute Position Encoder, Double Bowing, triple stops, double stops, convolution.

1. INTRODUCTION
This paper describes the fruits of the collaboration between the renowned cellist Frances-Marie Uitti and CNMAT researchers in the fall of 2005 sponsored by a UC Regents lectureship program. The augmented cello completed during the collaboration was used in performance at the end of her residence on November 4th 2005.

The starting point for the project was a 6-string cello built by Eric Jensen [4]. The main, unusual feature of this electric cello is a deep notch in front of the bridge co-designed by Ms. Uitti and Mr. Jensen. This allows Ms. Uitti to play using two bows simultaneously–one above and one below the strings–for chordal and other polyphonic textures [16] [12]. We were curious how much of our previous work on polyphonic signal processing for guitars could be leveraged for a bowed instrument in the hands of player who has already vigorously pursued the polyphonic potentiality of the instrument.

We will describe a new solution to the problem of changing tunings of the open strings, a matrix of switches and pressure sensors installed on the instrument, a novel bowed rotary encoder and the software used in the debut performance of the instrument.

1.1 Tuning Augmentation
Ms. Uitti uses a variety of non-traditional tunings to take advantage of the possibilities afforded by multiple stops and two bows.

The combinatorial elaboration of sounding strings for multiple stops described in Table 1 takes into account the limited access of the underbow to two strings. Other practical considerations make a couple of the stops difficult but the table clearly shows the advantage of 6 strings over 5 or 4 – especially for triple stops.

Table 1. Available stops for double bowing

	Strings		
Stops	4	5	6
4	1	2	3
3	4	9	12
2	6	9	13

The conflicting constraints of string displacement; stopping-hand reach, spread and strength; and the bridge arch result in a practical limit of six strings. Lap steel guitar players, freed of the reach and stopping pressure constraints, play 6-9 stringed instruments[2]. The additional constraint of the curved bridge to allow bowing of separate strings precludes adding as many strings. Chordal fingerings become more limited as the neck becomes broader, especially those chords where the little finger or ring finger needs to depress the lower strings while other fingers need a maximum curve to access the upper ones. These considerations explain why bowed chordophones such as the cello and viola d'amore have not explored the extremes of stopped string count achieved for the lute and theorbo.

The analysis so far only addresses the bowability of triple and quadruple stops. What pitches are actually available depends on additional, more complex constraints from the interaction of the stopping-hand reach and the chosen tuning. Ms. Uitti has already approached the limits of what is humanly possible with her stopping hand so the free design parameter is the tuning of the open strings, e.g. Scelsi's 4th string quartet [8].

The interesting question of which families of tunings to use will be the subject of a future paper. We choose here to focus on the ergonomics of quickly changing tunings: during a piece and even during a note, a technique used occasionally by banjo players and guitarists and developed to its extreme by Adrian Legg [3].

One approach to supporting different tunings is to use independent pitch shifting DSP algorithms on the signals captured by piezoelectric pickups under each string at the bridge. This method is used commercially for guitars and used notably by musicians who adopt many unusual tunings, Joni Mitchell, for example, who composes using scordatura tunings as a starting point [13].

During a previous project on hex guitar signal processing we identified several important challenges with electronic pitch shifting:

1) Numerous noticeable artifacts in the shifted sound.
2) Conflict between the acoustic sound and electronic sound in live performance
3) Unacceptably long latencies especially for low-pitched strings.

Our solution to these problems was to augment the cello by adding a mechanical tension-modulating device at the heel of the instrument.

Figure 1. Cello heel with string tuning device

This device was originally developed for guitars by Hipshot Inc [10]. We adapted it to the cello – primarily accommodating the larger cello string end. The device is normally floated from the heel of the instrument but we instead added it to an extension of the heel to maintain the existing short string length. This affords bowing below the bridge on the short strings.

Three possible pitches are available for each string adjusted by set-screws allowing for microtonal, 1/4 tone, 1/2 and whole tone tunings.

This arrangement works well avoiding problems with previous methods and we suggest it is a good example of the benefits of exploring non-electronic solutions to instrument augmentation challenges.

2. Gesture Sensing Augmentations

Foot control is commonly used in live performance especially with computer-based scores. We experimented with many foot pedal options and confirmed our early suspicions that these are hard to use in practice. Cellists use their legs to counteract the considerable torque generated by bowing. Their feet have to be firmly planted on the floor to comfortably do this for long periods with the necessary stability to support solid performances. Alternatives have been explored to this seated playing position including stands and harnesses[9] but these are not widely accepted on ergonomic and practical grounds. We therefore decided to focus our efforts on new interaction opportunities for the fretting and stopping hands – the core of the cellist's technique.

2.1 The stopping hand

For the stopping hand we provided a row of FSR's (Force Sensing Resistors) on the edge of the neck closest to the low-pitched strings. These were centered at the semitone positions of the string. This provides both a natural location (already thoroughly part of the cellist's technique) and no part of the hand can inadvertently touch this part of the instrument. The semitone positioning also suggests a convenient labeling of each control in a score.

FSR's have the advantage over switches of having a low profile and providing an extra control dimension (pressure). They also cost no more because the installation cost dominates the parts cost.

On the other edge of the neck we installed a continuous pressure-sensing strip accessed typically with the thumb.

FSR strips are cheap and convenient but unlike knobs and sliders they don't provide any tactile memory of a parameter setting. We addressed this by adding a slider. This most commonly was used to adjust the sound balance between processed and direct cello sound.

Figure 2. Cello Body showing neck and body FSR

We also installed a switch array directly below the bridge and an arrow of circular FSR's at the top of the body of the instrument. The switch array is used to make major "preset" changes during performance where the tactile feedback of the switches was important to confirm the change. Installing a small touch screen here would have allowed us to label the presets but we note that some performers prefer instrument interfaces where there is no dependence on visual feedback.

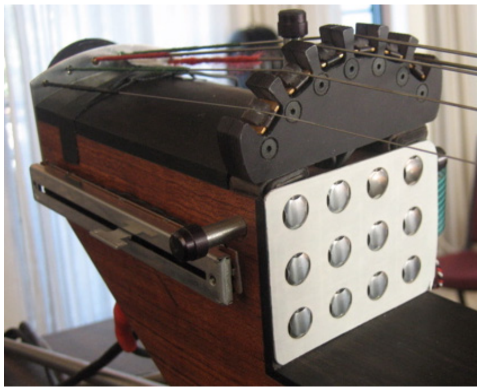


Figure 3. Cello Heel with switch array, hex pickups and slider

We attempted to sense string stop position using a resistive strip designed as a "ribbon" controller but found it too wide and short for this application. We also to measure the electrical resistance of the string from a conductive fingerboard to the nut but found that the distance/resistance function was highly non-linear and varied from string to string, presumably because of the exotic alloys and solid wound and stranded construction techniques used in cello strings. These difficulties were a turning point for the project: where we decided not to try to measure and track traditional cello-playing gestures but instead augment the instrument with new possibilities.

2.2 The Bowing Hand

For the bowing hand we introduced a novel application of a rotary absolute position encoder, a device that outputs a voltage corresponding to the angle of rotation of a shift from a reference position. We attached a wheel to the shaft of a commercially available encoder with a surface preparation that the bow could easily grip. We installed the wheel behind the heel of the instrument where it can be thought of as an extension of the "short string" bowing technique.

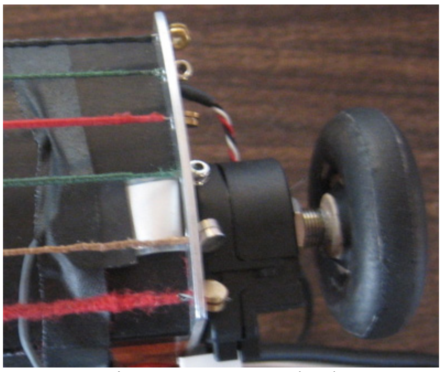


Figure 4. Sensor Wheel

3. Sensor and Sound Data Capture

All the resistive and switched inputs for gesture sensors were translated into voltages between 0 and 5v using simple resistor divider networks. These signals were carried on a multiwire cable to a DB25 connector plugged into one of the two Sensor ports of CNMAT's Connectivity processor [1].

Figure 5 CNMAT Connectiviy Processor

The piezo sensors for each string and two additional piezo pickups near the tail of the short strings were converted by custom-built charge amplifiers built into a special daughter card for the Connectivity processor.

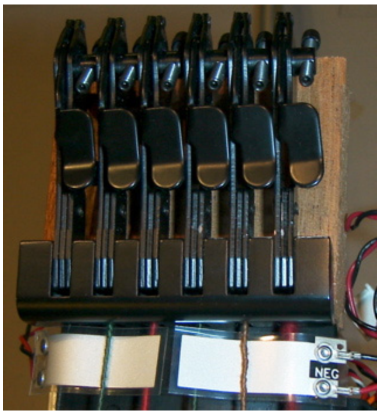


Figure 6. Short String Piezo Pickups

These analog signals are conditioned, converted into digital signals, serialized and aggregated into an Ethernet stream that was processed by custom software in Max/MSP. Sound output was also routed through Ethernet packets to the connectivity processor and demultiplexed into 8 balanced analog audio outputs.

4. Performance Software

We elaborated and augmented ideas originally developed for an earlier polyphonic guitar project [15] to reflect Ms. Uitti's aesthetic needs.

Each idea was implemented as a separate Max/MSP patch and each patch was controlled by a main supervisory patch that managed all the signal and gesture routing and also switched active patches according to selections by the performer.

One programming challenge is to give the performer as much meaningful control as possible without overwhelming them with parameters that they will find useless or, worse yet, distracting. It is important to work in a style that allows the programmer to quickly remap controllers and values to any location in the patch, and empowers the performer to feel that the software is actually responding to her actions.

To that end, overall control of the performance subpatches was managed using a combination of OSC (Open Sound Control)[14] and the *pattr* family of objects. Each of the hardware sensors was given a unique address in an OSC namespace, allowing individual subpatches to tap into the appropriate control data. Configurations that activated one or more subpatches were stored as presets in the *pattrstorage* object and triggered via the switch array (below the bridge). Smooth crossfades between successive configurations were achieved with *pattr*'s built-in interpolation features.

These features allowed the cellist to dynamically remap the meaning of her performance gestures according to the needs of the musical situation, quickly and smoothly moving between one set of patches and the next.

No matter what patches are in effect, the cellist always has control of her throughput gain, and the overall gain of the effects. Single controllers are mapped to each these gains, and remain fixed throughout the performance. This was important to allow the performer to react instantly to the musical situation, especially if the processing does not fit the character of the musical moment.

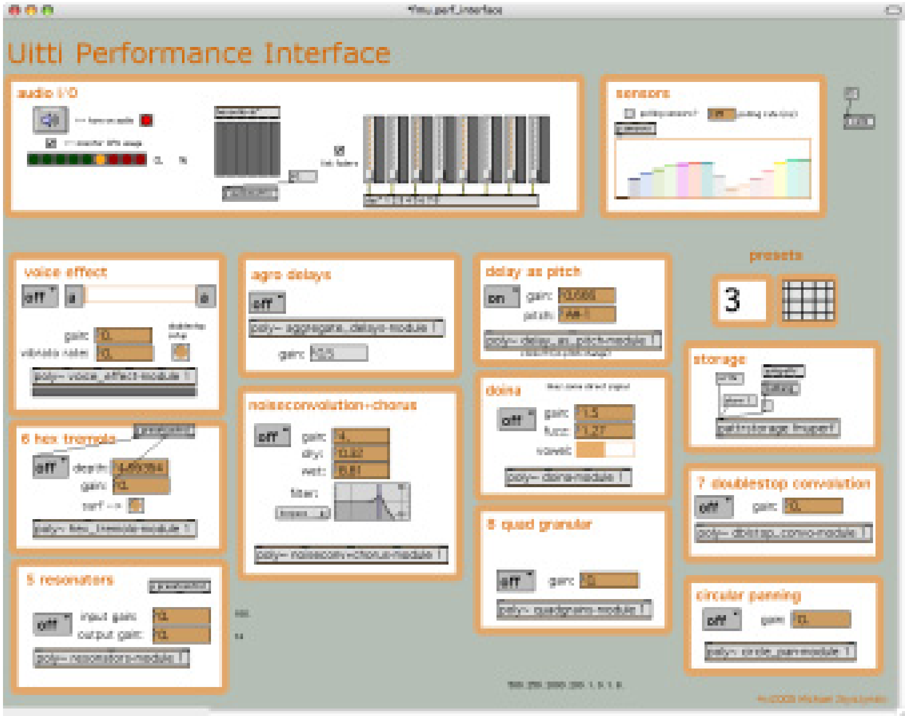


Figure 7. Performance patch

We briefly describe in the next sections some of the more compelling subpatches available.

4.1 Vocal Effect

For this effect we used a separate bank of five resonant formant filters for each string. These were tuned dynamically by interpolating between vowel pairs stored from a data set that included a,e,i,o,u for soprano, alto, bass, contrabass, and tenor voices. The appropriate vocal data set was matched to the tessitura of each string. Vibrato was created artificially by interpolated delay line modulation and modulated by pressure of the fingerboard FSR strip. This was used as a micro-rhythmic contrast against Uitti's normally fluctuating vibrato, creating changing beating patterns and synchronizations. Vowel pairs were chosen using the fingerboard FSR's and interpolations were driven by the patch.

4.2 Double-stop Convolution

The key idea of this patch is to use a separate convolution for all the double stop combinations and to process and spatialize the output of the convolved pairs independently. Since the convolution was performed by FFT's we were able to save computation by sharing the forward transform of each string signal.

Convolution works well in this situation because sound is only output if there is a signal in both inputs of the convolution. This is a fruitful area of exploration because double stops are a reliable musical gesture and the performer has immediate access to many independent streams of processing without having to choose them ahead of time with other gestures.

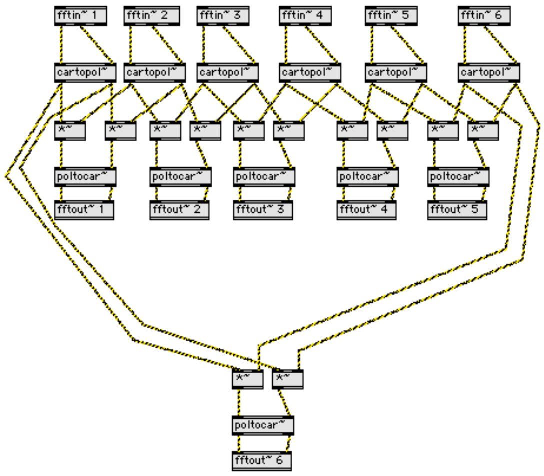


Figure 8. Double Stop Convolution

4.3 Quad Granular and Circular Panning

Two patches were combined in this effect with the intent of surrounding the direct sound of the cello with a diffused aura of related fragments. The fragmentation was achieved with a pair of stereo granulators, specifically *munger~* (from the PerColate [11] collection). These were set to create relatively long (2000ms ±200ms), widely spaced (500 ms ±250ms), irregular grains. Grains were generated from a 3000ms buffer, and could play back either forwards or backwards at the speed of the original performance. Each granulator is independent, and their outputs were interlaced and sent to the circular panner.

The panning patch diffused the sound in a circular array, maintaining a 180 degree separation between each channel of each granulator. That is, if left and right for the first granulator appeared at 45 and 225 degrees from the listener, the second granulator would appear at 135 and 315 degrees. Each granulator generated grains at random locations in their stereo field, so the result was a complex constellation of sounds. The entire sound field was rotated by the performer using the rotary encoder behind the heel of the cello. This gave the performer sensitive and expressive control of the direction and rate of the perceived motion. The angular displacement of the sounds was generated by Ville Pulkki's *VBAP* objects[5], allowing the angle to be specified independently of the specific number and location of loudspeakers.

5. Future Work and Conclusion

We will explore the use of touch panel displays for labeled buttons and the use of two-dimensional pressure sensing panels on the side of the body.

The position encoding wheel/bow sensor interaction shows a lot of promise in the augmented instrument context. We are exploring use of detents and weights to see how much tactile feedback can be exploited by the musician. We are also exploring new instrument interfaces built around this sensor. We will explore the addition of a servo motor to the drive of the encoder, a strategy that has been explored to research violin bowing [7].

We used surface wiring and temporary adhesives to provide the most flexibility in the development of the augmented instrument. Now that the design issues are settled we will mechanically integrate the sensors and bury the wiring within the instrument. We note that current construction techniques in solid-bodied musical instruments do not provide the channels and cavities in the neck of the instrument to facilitate this and suggest that simply routing cavities in the body of instruments for transducer electronics is insufficient to embrace the potential of modern sensing technology and the ambitions of future musicians.

The solutions developed in this collaboration can be further enhanced with a newly designed instrument and we can accommodate some of the ideas we were forced to discard. In particular we will be able to integrate stop position sensing and we will significantly augment the control possibilities of the new instrument by marrying it with a sensor-laden bow [6], a project already in the initial phases of design and construction by F.M.Uitti in her Sonic Lens Project. This sound/vision project supported by Stichting Steim and the Biennale of the Amsterdam Film Museum involved the triggering and manipulation of film using bowing gestures.

6. Acknowledgements

We gratefully acknowledge the prior work of Matt Wright and John Schott and the support the Chambers fund, UC Regents, and Waves Audio Ltd.

7. References

1. Avizienis, R., Freed, A., Suzuki, T. and Wessel, D., Scalable Connectivity Processor for Computer Music Performance Systems. in International Computer Music Conference, (Berlin, Germany, 2000), ICMA, 523-526, http://cnmat.CNMAT.Berkeley.EDU/ICMC2000/pdf/connectivity-processor.pdf.

2. Bechtel, B. Lap Steel Tunings, 2006, http://www.well.com/user/wellvis/tuning.html#8string.

3. Ellison, T. Against the Grain: A Chat with Adrian Legg, 2006, http://www.solidairrecords.com/AMR_interviews/legg.html.

4. Jensen, E. Jensen Electric Cellos, 2006, http://www.halcyon.com/jensmus/cello.htm.

5. Pulkki, V. Virtual sound source positioning using vector base amplitude panning. Journal of the Audio Engineering Society, 45 (6). 456-466.

6. Rasamimanana, N.H. Gesture Analysis of Bow Strokes Using an Augmented Violin IRCAM, Université Pierre et Marie Curie, Paris VI, 2004.

7. S. Serafin, M.B., S. O'Modhrain, C. Nichols. , Expressive controllers for bowed string physical models. in DAFX, (Limerick, Ireland, 2001).

8. Sciannameo, F. A personal memoir: Remembering Scelsi. The Musical Times, 142:1875. 22-26.

9. Steinberger, n. Cello Stand Instructions, 2006, http://www.nedsteinberger.com/instruments/cello/cr/celloinstr.html.

10. Trilogy. Hipshot Trilogy Multiple Tuning Bridge, 2006, http://www.hipshotproducts.com/trilogy.htm.

11. Trueman, D. and DuBois, R.L. PeRColate, 2001, http://music.columbia.edu/PeRColate/

12. UItti, F.-M. Two Bows, 2006, http://uitti.org/twobows.html.

13. Whitesell, l. Harmonic palette in early Joni Mitchell. Popular Music, 21 (2). 179-193, http://www.journals.cambridge.org/action/displayAbstract?fromPage=online&aid=107747.

14. Wright, M., Freed, A. and Momeni, A., Open Sound Control: State of the Art 2003. in International Conference on New Interfaces for Musical Expression, (Montreal, 2003), 153-159, http://www.music.mcgill.ca/musictech/nime/onlineproceedings/Papers/NIME03_Wright.pdf.

15. Yeung, B. Guitar Dreams SF Weekly, San Francisco, 2004, http://www.sfweekly.com/Issues/2004-01-07/news/feature.html.

16. Zorn, J.E. Arcana. Granary Books, 2002.

Keynotes and Panel Discussion

Keynote by George Lewis

Living with Creative Machines: A Composer Reflects

George Lewis, improviser-trombonist, composer, and computer/installation artist, studied composition with Muhal Richard Abrams at the AACM (Association for the Advancement of Creative Musicians) School of Music, and trombone with Dean Hey. The recipient of a MacArthur Fellowship in 2002, a Cal Arts/Alpert Award in the Arts in 1999, and numerous fellowships from the National Endowment for the Arts, Lewis has explored electronic and computer music, computer-based multimedia installations, text-sound works, and notated forms. A member of the AACM since 1971, Lewis's work as composer, improviser, performer, and interpreter is documented on more than 120 recordings. His published articles on music, experimental video, visual art, and cultural studies have appeared in numerous scholarly journals and edited volumes, and a book, *Power Stronger Than Itself: The Association for the Advancement of Creative Musicians* will be published by the University of Chicago Press in 2007. Lewis is the Edwin H. Case Professor of American Music at Columbia University. Lewis's pioneering multi-computer interactive work, *Rainbow Family*, was commissioned by Ircam and performed in 1984 in its Espace de Projection. Lewis also performed his virtual orchestra work, *Voyager*, at the Ircam Summer Academy in 1994.

Keynote by William Gaver

Listening to the World: Information and Aesthetics

Auditory interfaces for computers can be based on auditory icons, in which computational events are signaled by analogous sound-producing events. For example, selecting a file might make the sound of tapping on an object, with the type of file indicated by the material of the object and the size of the file by the object's size. This strategy is based on a distinction between musical listening, in which we attend to features of sound such as pitch, loudness and timbre, and everyday listening, in which we attend to features of events such as size, force and texture. Everyday listening is relatively neglected, but it is clear we can perceive a huge amount of information about the world from the sounds it makes. I will illustrate a variety of auditory interfaces that make use of this potential.

If the informative nature of sound is important for auditory interfaces, so are the aesthetics of auditory cues. Positioning designs on the cusp of information and aesthetics can be used to create opportunities for exploration that are simultaneously pleasurable and meaningful. I will describe several prototype systems that illustrate design for this sort of playful engagement, and discuss the design of a new system that brings environmental sounds into the home to support local awareness and evoke curiosity in aesthetically pleasing ways.

William Gaver is a professor of design at the Goldsmiths College, University of London, and a Principle Investigator on the Equator IRC. He has pursued research on innovative technologies for over 15 years, working with and for companies such as Intel, France Telecom, Hewlett Packard, IBM, and Xerox. He has gained an international reputation for a range of work that spans auditory interfaces, theories of perception and action, and interaction design. Currently he focuses on design-led methodologies and ludic technologies for everyday life.

Panel Discussion moderated by Michel Waisvisz

Manager or Musician?

About virtuosity in live electronic music.
Do we operate our electronic systems or do we play them?

enchantment versus interaction

if our goal is musical expression we have to move beyond designing technical systems.
we have to move beyond symbolic interaction.
we have to transcend responsive logic;
engage with the system:
power it and touch it with our bodies,
with our brains.
invent it and discover it's life;
embrace it as instrument.
an instrument that sounds between our minds.

we will have to operate beyond pushing buttons and activating sensors
beyond isolating gestures and mapping data and parameters
beyond calculating response
beyond assuming that the concept will create music

we should abolish the illusion of 'control'
merge our intentions into those of the instrument and the audience
get inspired by change, miscalculation, invested instinct, insightful anticipation, surprise and failure

the sensors, the logic, the artistic debate, the technical debate, the circuits, the theories about perception, the new war driven technologies, the ability or dis-ability to communicate, the conferences, the endless experimentation with system tweaks, the touch and sound, the reoccurring state of disbelief, the craving for the stage, the difficult and great collaborations, composing the now, the survival of the electronic music scenes, the nime, the industry, the independents, the musical fun, the appreciation of difference, the body as source of electrical and musical energy, the bonding of thinkers, the rapid improvisers,
... just to extrapolate some ingredients and vehicles of our quest.

it might work if we manage to express ourselves musically by moving beyond interaction,
beyond mere technical beliefs and disbelief.
by engaging, by trusting ourselves into the potential of our new instruments,
enchanting our sounds, our audience.

enchantment is not only a state of mind,
it is a technology
designing for new musical expression is casting a spell on instrumental practice.

Michel Waisvisz, Limerlé, May 2006

Workshops

Sunday, June 4, Workshop, IRCAM Salle Igor-Stravinsky

Improvisation and Computers

After a first workshop at the Sound and Music Computing conference in 2004, IRCAM is organizing the second workshop for NIME 06, this time addressing the LAM Research Network (Live Algorithms for Music). The workshop brings together internationally recognized musicians and researchers who will present their work from a musical as well as technological point of view. In particular, artistic aspects of interaction and the implication of "creative machines" in the process of music improvisation and composition will be discussed.

Presenters: George Lewis (Columbia University, New York City), Bernard Lubat (Pianist, France), Michel Waisvisz (STEIM, Netherlands), Joel Ryan (Institute of Sonology, Netherlands), David Wessel (CNMAT, California), Mari Kimura (Julliard, New York City), Michael Young and Tim Blackwell (Goldsmiths College, Great Brittan) with the LAM Research Network (Great Britain), Francois Pachet (Sony CSL, France), Gerard Assayag (IRCAM, France), Marc Chemiller (IRCAM/CNRS, France), George Bloch (Composer, France) with Philippe Leclerc (Saxophonist, France), Émilie Rossez (Video artist, France)

Sunday, June 4, Workshop, IRCAM Studio 5

Choreographic Computations:

Motion Capture and Analysis for Dance

This workshop will focus on new innovations combining motion capture and computer-based techniques with choreography and performance, an area in which an international group of artists and researchers has been breaking new ground. The software artists and programmers involved are exploring a range of heterogeneous computer concepts and approaches from agent-based aesthetics to the development of new tools and pathways to support collaborative composition (using software platforms and environments such as EyesWeb, Isadora, MnM, and Fluid). Through their close collaboration with the choreographers (e.g. Trisha Brown, Myriam Gourfink, and Dawn Stoppiello), a shared understanding of movement and gesture is evolving to support the application of complex algorithmic procedures to equally complex choreographic creation. Together these practitioners are carving out fresh territory for correspondences between choreography and computation. We have invited some of these artist-researchers here to the present their recent collaborative work in the context of this NIME06/IRCAM workshop.

Organizers/Presenters: Scott deLahunta (Writing Research Associates, Netherlands), Frédéric Bevilacqua (IRCAM, France)

Presenters: Antonio Camurri (Director, InfoMUS lab, DIST-University of Genoa, Italy), Mark Coniglio (Composer/Media Artist, Troika Ranch Company, New York City), Marc Downie (Artist and Researcher, OpenEndedGroup), Myriam Gourfink (Choreographer), Rémy Müller (Researcher, IRCAM, Paris)

Thursday, June 8, Workshop, IRCAM Studio 5

Network Performance

This workshop brings together artists and researchers to present different artistic approaches, paradigms and technologies for music performance over the network. While the first part of the workshop is focusing on electro-acoustic music performance. distributed over the network, the second part invites various approaches to collaborative performance inherently based on digital interfaces and network technology.

Presenters: Don Foresta (MARCEL, LSE London), Atau Tanaka (Sony CSL, France), Chris Brown and John Bischoff (Mills College, California), Phil Stone (University of California Davis), Scot Gresham-Lancaster (Cogswell Polytechnical College, California), Tim Perkis (Composer, Performer), Mark Trayle (CalArts School of Music, California), Alain Renaud and Pedro Rebelo (Sonic Arts Center, Northern Ireland), Alexander Carôt (International School of New Media, Germany), Bruno Verbrugghe (IRCAM, France), Elaine Chew (Integrated Media Systems Center, California), Lodewijk Loos and Fokke de Jong (Waag Society, Netherlands), Sergi Jorda, Martin Kaltenbrunner, and Günter Geiger (MTG – University Pompeu Fabra, Spain), Frédéric Voisin (CIRM, France), Robin Meier (Composer, France)

performances and installations

NIME Performances

Voice and Infrared Sensor Shirt

Tomomi Adachi

This project aims to connect sound processing techniques with physical gesture accompanied by vocal utterance. I am interested in the physical movement of vocal performers, in the use of gesture in everyday conversation and, as an extreme example, in sign language. Almost all vocal performers use conspicuous hand movements emphasizing the character of their music. I work with electronics for real-time voice processing, because I believe that the interaction between subjectivity and objectivity is an important aspect of art. The sound of the human voice is rather subjective and sound-processing acts to objectify the raw voice. In this performance the audience can trace the process of objectification. It integrates these two themes: reconstructing the relationship between utterance and gesture, and clarifying the musical process.

The shirt I use in this performance is equipped with ten infrared distance sensors, as well as four buttons on a band in my left hand. This way the space in front of the performer becomes a three-dimensional sensing area connected to a Max/MSP patch running on a laptop computer through a Voltage-to-MIDI converter. The patch modulates the vocal input in real-time. The sensor system and its wiring are visible to the audience. Very simple processing techniques are used such as delay, sample playback, and pitch-shifting. Complex effects are generated through compound gestures. This allows the audience to visually perceive the entire process like with traditional instruments. All sounds come directly from the voice. No prerecorded samples are used except for a cluster of sine waves in the final part of the piece expressing a counterpoint between the voice and the movement.This is a roughly structured improvisation. The ultimate goal is to create a humorous and exciting musical performance with contemporary technology.

http://www.adachitomomi.com/

Solo Performance for Indeterminate 'Dirty Electronics', 'Flickertron' and 'Pseudophone+'

John Richards (aka kREEPA)

A ten-minute foray into the irrational and indeterminate world of "*Dirty Electronics*". The electronic nebulous of the "*Kreepback*" instrument - a modular assemblage of digital, analog, and acoustic sound sources - is left to run riot, generating a dissonant sound world of hisses, squelches and pulses. The performer's control of the system is carried out through the very essence of the instrument: electricity. Hand-driven dynamos "feed" the tin, copper-wired, binary organism. The performance will also feature interfaces specially constructed for NIME 2006: the "*Flickertron*", a strobe light driven photocell instrument, and the *Pseudophone+* where crushed cans and tins are brought to life musically through the use of natural resonance and conductivity.

http://www.kreepa.com/

Pieces for Plants #7

Miya Masaoka

Pieces for Plants is an interactive performance for houseplants, electrodes and computer, and performer. The electrodes and computer provide a sonic indicator of the plant's electrical activity and its physiological response to its surroundings in real-time.

In this piece, highly sensitive electrodes connect the leaves of a philodendron to a laptop computer. The electrodes give a "voice" to the plant, providing a sonic indicator of the plant's electrical activity and its physiological response to its surroundings. When the performer approaches the plant, one can hear the plant's response to her presence; when her hands are moved around the plant at varying speeds, the plant's responses are audible. The responses are extremely varied, and are sonified employing an array of parameters including partials selection, granular synthesis, and density. With this "sonification" of data, the performer can excite the plant and shape the performance. Audience members are encouraged to consider the possibility and potential of a plant that has a kind of awareness, that can communicate, taking on qualities of consciousness. Previous versions have been performed at the Lincoln Center Out of Doors Festival's Homemade Instrument Day and the Electronic Music Foundation's Ear to the Earth Festival in 2005.

http://www.miyamasaoka.com/

Sight Sound -Phenakistoscope-

Saburo Hirano

Sight Sound comes from the concept of Sight Singing. Visual information is transformed into sound in real-time; sound is controlled by image. A *Phenakistoscope* creates the illusion of moving image by presenting a series of images through a slit. Meanwhile, in the DJ scene, turntables are used not just for music reproduction, but as a musical instrument. In this piece, the turntable has taken on these two functions. It rotates the Phenakistoscope's “sheet" instead of a vinyl record. Here rather than slits, the computer program captures a continuous “roll” of slits at an image sampling rate. I combine these two media (aural, visual) in a new type of musical instrument interface that utilizes moving images. The Phenakistoscope's sheets are captured by a camera to create a moving image, the camera detects specific colors and converts them to electronic percussion sounds. To display the color detection, simple geometric shapes are overlapped with the moving image. We can perform with pre-recorded sounds, percussion sounds, and moving images by playing the turntable.

Mouseketier Praxis

Mark Applebaum

Mouseketier Praxis (2003/2006) is an improvisation for the *Mouseketier*, an original instrument built in the summer of 2001. The Mouseketier is the most recent electroacoustic sound sculpture in a lineage that begins in 1990 with the Mousetrap, and subsequently includes the Mini-Mouse, the Duplex Mausphon, the Midi-Mouse, and 6 Micro Mice (constructed for the Paul Dresher Ensemble).

The Mouseketier consists of three amplified soundboards—pink, blue, and yellow triangles with piezo contact pickups—arranged as tiers. In addition to its three principal pickups are five that work as switches to trigger external processes or computer functions. Mounted on the soundboards (the three tiers) are junk, hardware, and found objects (combs, squeaky wheels, threaded rods, doorstops, nails, springs, Astroturf, ratchets, strings stretched through pulleys, twisted bronze braising rods, and, of course, mousetraps) that are played with chopsticks, plectrums, knitting needles, a violin bow, and wind-up toys. The resulting sounds are modified with a tangle of external digital and analog signal processors. The instrument sounds great, but it is intended equally for its visual allure.

Annoyed by the transportation and set-up challenges associated with the behemoth Mousetrap, I built the Mouseketier as a kind of travel model. Not only does it set up in minutes (instead of hours), its flight case—meeting the airline specifications—was designed first. Thanks go to my wife Joan for contributing the Mouseketier's basic architecture and name.

http://www.markapplebaum.com/

Children of Grainger

Bent Leather Band

Joanne Cannon (*Serpentine-Bassoon* and *Contra-Monster*) and Stuart Favilla (*LightHarp*).

… Music is an art not yet grown up; its condition is comparable to that stage of Egyptian bas-reliefs when the head and legs were shown in profile while the torso appeared "front face" - the stage of development in which the myriad irregular suggestions of nature can only be taken up in regularised or conventionalised forms. With Free Music we enter the phase of technical maturity such as that enjoyed by the Greek sculptors when all aspects and attitudes of the human body could be shown in arrested movement.

… It seems to me absurd to live in an age of flying and yet not to be able to execute tonal glides and curves - just as absurd as it would be to have to paint a portrait in little squares (as in the case of mosaic) and not to be able to use every type of curved lines. If, in the theatre, several actors (on the stage together) had to continually move in a set theatrical relation to each other (to be incapable of individualistic, independent movement) we would think it ridiculous, yet this absurd goose-stepping still persists in music. Out in nature we hear all kinds of lovely and touching "free"; (non-harmonic) combinations of tones, yet we are unable to take up these beauties and expressivenesses into the art of music because of our archaic notions of harmony. - Percy Aldridge Grainger 1938

http://home.mira.net/~favilla/

Club NIME I

Revelations

Juraj Kojs

Revelations is scored for elastic rubber balls, Bocci metal balls, glass marbles, resonant plates made of Plexiglas, plastic (glossy and matte), aluminum, brass, and computer. The three performers use circular toys to excite the plates in a variety of performance modes such as bouncing, rolling, and scraping. At the end, the players perform with a shaker filled with glass marbles.

Stochastic physical models and bowed percussion bar and violin models were utilized in the computer part of *Revelations*. The computer component consists of preprocessed (tape) and real-time materials. Sampled materials of percussive surfaces such as timpani, tam-tam, Glockenspiel, Tenor Drum, and glass pane scraped with a rubber ball are mixed with the models in the pre-recorded part.

Hymn to Ampere

The Breadboard Band

Katsuhiko Harada, Masayuki Akamatsu, Shosei Oishi, Kazuki Saita

The Breadboard Band uses self-made electronic circuits to perform music. A breadboard is a perforated board with connector holes in a grid, in which electronic components have been inserted in order to build the prototype. The electronic components can be inserted or removed with ease, making it simple to change the wiring with jumper cables. Utilizing the features of the breadboard, The Breadboard Band creates audio and visual circuits on the board, modifying them during performance.

One hundred years after the first public performanceof the Telharmonium, the first electronic musical instrument, The Breadboard Band reexamines black-box electronic musical instruments and computers. We touch and form electric circuits manually, producing audio and visual expression through the most minimal, fundamental elements. It can be considered the hardware version of software programming, a performance of "on-the-fly wiring".

http://www.breadboardband.org/

Modal Kombat

David Hindman and Evan Drummond

Modal Kombat is a live instrument battle channeled through the video game Mortal Kombat. Audio from two classical guitars controls the characters' movements in Mortal Kombat Trilogy. Acoustic-electric instruments have replaced the typical video game controller. We have developed means of mapping pitch, pitch sequences, and volume to corresponding actions of a video game character. In the case of *Modal Kombat*, pitch information corresponds to the fighter's actions, creating a mix between creative choreography and intense competition. The duo delivers at once modern classical guitar performance and public video game competition. The result, *Modal Kombat*, is a modern-day "dueling banjos." *Modal Kombat* creates an engaging combination of music, choreography, and good old-fashioned fantasy street fighting.

http://www.modalkombat.com/

sCrAmBlEd?HaCkZ!

Sven König

Gramophone records, magnetic tapes, vinyl records, digital samplers, and computers liberated samples long ago. But still - to avoid infringing on copyrights - one has to decide which sample one actually wants to steal. One has to arduously load audio files into sample editors or sequencers. One has to cut, copy, paste, and arrange.

sCrAmBlEd?HaCkZ! is a Realtime-Mind-Music-Video-Re-De-Construction-Machine. It is conceptual software that makes it possible to work with samples by making them available in a manner that does justice to their nature as concrete musical memories.

Through my interest in artistic strategies and social practices of appropriation – collage, montage, sampling and remix in general and plunderphonics, bastardpop and mashups in particular – the idea of a hypothetical mind music machine has evolved which, as a metaphor, helped the concept and the design of sCrAmBlEd?HaCkZ! to take shape.

http://popmodernism.org/scrambledhackz/

Audile

Yutaka Makino and dj sniff

Audile is an improvisational duo where the musicians utilize custom-built software and hardware to explore new expression in contemporary electronic music. Yutaka Makino's primary instrument is a unique real-time granular synthesis software. dj sniff uses the turntable to generate a variety of sound textures that are processed by a set of custom dsp effects. The use of custom tools allows them to break away from the subtle gestures of conventional computer interfaces. We seek a virtuosity that a modern-day electronic musician can attain in a live improvisational context. We realize that this cannot be achieved by the software they write or the speed with which they cut the fader, but only through the refinement of their audile senses.

http://audile.org/

Invited Headline Club Performance

eRikm

eRikm has instinctively followed an unusual, risky, musical path. Stemming from his interest in visual arts and his first musical experience as a rock guitarist, he has for some years now become steadily more and more renowned for his virtuoso turntabling and his use of electronic instruments and tools in an integrated scenic set-up. He demonstrates the relationship between rock music (in its broadest sense) and contemporary music, not hiding behind specific cultural camouflage.

His style stems from the duality between his experimentations in sound material, its envelope and its tessiture, and his work and taste for the anecdote, the reference. His primary aim is not to quote his system of references but to create for himself a bank of singular material to compose without referring to other works. He seeks a subtle abstraction, a new electronic position: like a pause in the contemporary sonic agitation, and, in opposition to this, a desire to weave a musical oeuvre whose intricacies are easy to follow, easing the listener into the vistas of his composition, without exaggerated technical or intellectual posturings.

http://www.erikm.com/

NIME 06 Special Guests

Vista, Tricot

Mari Kimura

For violin and interactive system

As a solo performer, I find it mysteriously captivating to wear two musical identities at once. My interest in these kinds of multi-musical identities has led me to develop interactive works that aim not at extending the violin but rather at splitting my musical identity. I try to slightly disorient myself aurally and physically.

In *Vista*, I challenge my own improvisational performance technique by using certain pitch and rhythm schemes, to deliberately play tricks on my ears and confuse my musical psyche. *Vista* was originally recorded as a "duo" in which I over-dubbed my own improvisation. The creative process of the recording, during which I got slightly disoriented, inspired me to create this interactive version. The solo violin is processed by and interacts with the computer in real-time, and no recorded or sampled materials are used. Nevertheless, the virtual "other" violin takes on a life of its own and, through feedback, becomes quite a disorienting and challenging partner.

I find it challenging to develop this kind of mental agility along with physical virtuosity, and find that it ultimately expands my vocabulary as a composer with real-time interactive computer, as a violinist, and as an improviser.

Tricot (lit: "Knitting") is an interactive improvisation work for violin and computer. For several years, I have been mainly focusing on writing interactive works with quite a specific and predetermined compositional structure. *Tricot* is a departure from my usual schemes. I am giving the computer freedom of choice on how to respond musically, as well as the right to participate in creating the overall musical structure with me. Although these "choices" are still created (programmed) in the first place by myself, they are quite unpredictable and depend a lot on the particular interaction during the performance. In this way, I am forced to encounter and interact with the computer's musical and structural choices.

While my previous work *Vista* is more acoustic interaction, playing on one's own auditory illusion, in *Tricot* I see myself weaving in and out of musical threads and patterns created by the computer, and together we converse and come up with a new creative environment.

Trio Improvisation

George Lewis and Alexander von Schlippenbach

For piano, trombone, and two interactive computer pianists *(Yamaha Disklaviers)*

This is an open improvisation for two people and two computer pianists, both of which are running a version of an interactive music system designed by George Lewis and programmed in Max/MSP by Damon Holzborn. In this work, the improvisers are engaged in dialogue with a computer-driven, interactive "virtual improviser." As with Lewis's earlier work, *Rainbow Family* (commissioned by Ircam and premiered in 1984), a computer program analyzes aspects of a performance in real-time and uses that analysis to guide the generation of complex responses to the musician's playing, while also establishing its own independent generative and analytic behavior. The improvised musical encounter is constructed as a negotiation between players, some of whom are people, others not—a subject-subject model of discourse, rather than a stimulus/response setup. The pianos and the people improvise the music in real-time, accompanying, dialoguing, and negotiating among each other. No pre-stored motives or themes are used by any of the players at any point in the performance.

Le Loup, Lifting, Crackle, Noise, Nevrose

Sensors_Sonics_Sights

Cécile Babiole (*ultrasound sensors)*, Laurent Dailleau (*Theremin)*, Atau Tanaka (*BioMuse)*

S.S.S creates a dynamic sound/image environment performing visual music through gesture, a trio of sensor instruments where two are playing sound and one plays image. Advanced interfaces where electromyogram biosignals allow direct mastery of sound synthesis are found alongside historical analog instruments that are digitally augmented. Computer graphics become not the result of algorithmic or synesthetic processes, but an instrument of gestural, musical expression.

Le Loup – manipulation of recorded animal sounds create the mysterious ambience where nature is re-sculpted.

Lifting – Melodic identity and pure tones native to the Theremin are taken over by resonant noise, leading way to a quotation and zooming into samples of music from the art rock movement of the 1970's.

Crackles – Particles in a three-dimensional pixel space set the stage for a playful throwing and catching of elements where absence creates as much space as presence.

Noise – Sound and image take on a physicality as wireframes explode beyond the edges of the screen, and frequencies materialize creating visceral sensations.

Nervrose – A grid structure contains an organic beast, seeking to escape from its cage. We enter this room of padded walls, a straitjacket muzzling our utterances.

http://www.xmira.com/sss/

On composing the now

Michel Waisvisz

On composing the now is a live electronic performance of musical thoughts about immediacy, involvement, touch, energy, engagement, grooves, narrative sound, reckless jumps, modest anticipation, desired change, intention space, absent-minded play, gamble, instinct, and disaster executed by leaving these words behind and diving into sound.

This performance is dedicated to the NIME community.

http://www.crackle.org/

Club NIME II

Dual Airbags

Ben Neill and Nicolas Collins

Ben Neill and Nicolas Collins began their collaboration inspired by their unusual, if highly compatible, home-made musical instruments. In addition to the valves normally used in playing a trumpet, Neill's *Mutantrumpet* has three extra valves for channeling the air column to three different bells, one of which is equipped with a slide mechanism (like that of a piccolo trombone). In an age of electronically extended instruments, the "acoustic extensions" of the *Mutantrumpet* are refreshingly different.

Collins performs on his trombone-propelled electronics. The slide of an old trombone was coupled to an optical shaft encoder (essentially 1⁄2 of a mouse) via a retractable dog leash; by pressing switches on a small keypad mounted on the slide the player "clicks and drags" numerous parameters of a custom digital signal processor. The processor output is connected to a speaker attached to the mouthpiece of the trombone, so that the electronic sounds can be further processed acoustically by moving the slide and using mutes to affect formants, filtering, and articulation.

http://www.benneill.com
http://www.nicolascollins.com

Circumference Cycles

Chris Strollo, Tina Blaine, Robin Stanaway

Circumference Cycles is a sound and light performance piece incorporating the sculpture/instrument of artist Robin Stanaway. This sculpture called "Circumference" is made of hand blown glass circular disks suspended by thin steel cables. Vibrations made from striking the glass and strings are amplified and processed while light is projected through the 30 inch diameter rondels. The music created with the Circumference is both rhythmic and harmonic, resulting in a sound that is a blend of the tones of the glass and the guitar strings, combined and resonating sympathetically. This performance includes two rondels and two players, each processing their signal independently. Once in the digital domain, multi-timbral shaping and long repeated delays help to create a fabric of rhythms and tones. Light projected through the pieces generates reflections on the backdrop.

This performance is sponsored with the support of the Entertainment Technology Center at Carnegie Mellon University in Pittsburgh, PA.

http://www.jamodrum.net/

---

*reacTable**

Sergi Jordà and Chris Brown

The reacTable* is an electronic musical instrument built upon a tabletop tangible user interface. Several simultaneous performers share complete control over the instrument by moving physical artifacts on the table surface and constructing different audio topologies in a kind of tangible modular synthesizer or graspable flow-controlled programming language.

The reacTable* is a translucent table. A video camera continuously analyzes the table surface, tracking the nature, position and orientation of the objects that are distributed on its surface. Users interact by moving the objects, changing their position, their orientation, or their faces. These actions directly control the topological structure and parameters of the sound synthesizer. A projector draws dynamic animations on its surface, providing a visual feedback of the state, the activity, and the characteristics of the synthesized sounds.

http://www.iua.upf.es/mtg/reacTable/

Connected Spaces

Satoru Higa and Akihiro Kubota

Connected Spaces is a duo live coding performance using originally developed three-dimensional visual audio programming environment, called "VP3L." VP3L is a kind of patch-based visual music programming language that is a 3D space, allowing the musician to rotate, zoom in/out, and walk around inside the program during coding and performances.

VP3L's 3D programming space is also a sound space. The position of the output objects in the 3D programming space corresponds to that of the sound sources in the 3D audio environment. Both performers and audience members can walk through the sound space during the performance.

http://lalalila.org/vp3l/ConnectedSpaces/

---

backscatter

Robin Fox

Robin Fox's recent research into audiovisual equivalence employs an audio-controlled laser system that translates voltage from a sound card directly into visible light shapes and abstract forms. This recent work is a progression from the two dimensional explorations of signal equivalence that he has undertaken previously, using the Cathode Ray Tube to convert audio to visual data. Combining preset sonic objects with live laptop based improvisation; the result is a quasi-synesthetic amalgam of sense modalities. As well as the new interface for musical expression used to perform the audio, the translation from sound to light and movement extends the possibilities for the expression of sonic forms.

The backscatter project represents the audiovisual aspect of performer, composer, and researcher Robin Fox's diverse output. The performance presents a series of real-time synesthetic experiences which meld sound and light at the point of signal path and electrical current. The same electrical signal that moves the speaker cones also affects the electromagnets inside a Cathode Ray Tube creating a synesthetic audio-visual immediacy for both the performer and the audience.

The performance system designed for the live realization of these works involves the design of visibly interesting sound sources in Max/MSP, the adaptation of certain haptic controllers (shuttle pro, track ball mouse etc) that allows for interactive sound/visual performance at the level of a performance instrument and, finally, the incorporation of pressure and light based sensors to enhance the playability of the system.

http://www.synrecords.com/display.php?CatID=27

NIME Gallery

Mandala 3 and Mandala 4

Greg Schiemer

Mandala 3 is an interactive work for four mobile phones. The work was created using java software developed in the Pocket Gamelan project, which explores microtonal musical applications of mobile technology. The work is based on a Dorian mode of Al-Farabi's Diatonic, a seven-note scale with the following pitches: 1/1, 49/48, 7/6, 4/3, 3/2, 49/32, 7/4, 2/1. *Mandala* 4 explores the Euler-Fokker Genus, another of Wilson's scales based on Product Sets, in which the 7th harmonic is predominant among the 6 generating harmonics (3.7.7.7.7) used to form the scale: 1/1, 1029/1024, 8/7, 147/128, 21/16, 343/256, 3/2, 49/32, 12/7, 7/4, and 2/1. The work is dedicated to Erv Wilson. The work uses purpose-built algorithms realized on java phones; each phone is used both as a sound source and a controller; each performer in turn controls harmonic modulation via Bluetooth using a form of pitch multiplication which is effectively a form of transposition. Because the piece also exploits various beat patterns created by unequally spaced just intervals, its musical focus is as much on variation in meter, rhythm, and timbre as it is about harmonic movement. The system was developed with funding by the Australian Research Council.

http://www.uow.edu.au/crearts/staff/schiemer.html

Bowl Chant

Garth Paine and Michael Atherton

Bowl Chant is a composition for four Tibetan Singing Bowls, two small Thai mediation bells, and two ancient Balinese gongs. The work is controlled using four sensors allowing the performers to add or refresh material being explored by the Capybara system; a dialogue between performers and computer is established. The composition explores the interstitial spaces between the ancient and modern, seeking a meeting point where the two worlds support each other in a rich new timbral space.

http://www.activatedspace.com/
http://www.syncsonics.com/music/BowlChant.html

Live Fire

Ximena Franco and Enrique Franco

Live Fire is an interactive video, sound, and animation piece. The interactivity of the piece is controlled by two performers with a keyboard and a mouse in real-time through PD (Pure Data) and GEM (Graphics Environment for Multimedia). In this way it is possible to change the order and duration of the video footage, animations, and audio samples at will during the performance. There is also a typist, the third performer, who uses an old typewriter that triggers sounds and visual effects through a microphone and a video camera. In this way we construct a narration in random order but that ends the same way as a result of the completion of an interactive animation that is constructed throughout the performance. The narration is about the violence and the war in Colombia in which the government, national army, paramilitaries, guerrillas, and people have their own part as culprits and victims. Our point of view on the conflict is that there is always hope and desire within the Colombian population to keep trying to fulfill their dreams and to stay alive.

http://homepage.mac.com/rugitus/

Jam'aa

Gil Weinberg

J*am'aa* is an interactive composition for a group of human percussionists and a robotic drummer. The robot, named Haile, is designed to listen to live players, analyze their drumming in real-time, and use the product of this analysis to play back in an improvisational manner. It is designed to combine the benefits of computational power and algorithmic music with the richness, visual interactivity, and expression of acoustic playing. In *Jam'aa*, Haile listens to and interacts with two humans playing Darbukas, Middle Eastern goblet-shaped hand drums. Haile listens to audio input via a microphone installed on each drum, detecting aspects such as note onset, pitch, amplitude, tempo, and rhythmic density. Based on these detected features, it utilizes six interaction modes that are designed to address the unique improvisatory aesthetics of the Middle Eastern percussion ensemble. Haile responds physically by operating its mechanical arms, adjusting the sound of its hits in two manners: pitch and timbre variety are achieved by striking the drum head in different locations while volume variety is achieved by hitting harder or softer. We believe that when collaborating with live players in *Jam'aa*, Haile can facilitate a musical experience that is not possible by any other means, inspiring players to interact with it in novel expressive manners.

http://www-static.cc.gatech.edu/~gilwein/Jamaa.htm

ATT

Georg Holzmann

ATT, acoustical table tennis, is a mixture of a performance and an audiovisual installation for two, 8 or 12 speakers spread over the room, one table tennis ball, a video projector and a computer. The sound materials are live sampled table tennis balls and the composition rules are derived from table tennis rules. The speakers are divided into two players (one player at the right, one at the left side, see picture 2) and they are controlled via specific probability functions (markov chains). The game is over if one player reaches 21 points (like in the original game rules). The current points of the players are displayed very large with a video projector. As an additional dimension the room/place comes into the game: With the 12 speakers the table tennis field is enlarged, and the listeners can walk through this field hearing the "acoustical balls" flying from one player to the other. In contrast to the "real" table tennis, more and more balls are sampled and thrown into the game. So a very complex sound network becomes spread over the whole room/place. The exciting thing: You never know in advance which player wins and how much time he will need to eliminate the adversary!

http://grh.mur.at/projects/att.html

Club NIME III

piano works

Antoine Schmitt

piano works is an improvisational piece composed with and for the nanomachine instrument. It is an homage to the king of the instruments, the piano, a very complex machine with a very straightforward approach, which can be explored by children and expressed by experienced improvisers at the same time through the material, the harmonics, and the rhythm: the one place in music where language (the note) rubs itself against reality (the touch). In this piece, small fragments of very fragile piano phrases and touches are slowly looped and layered, yielding a very harmonically and rhythmically complex audio and visual composition that barely remains on the verge of readability.

The nanomachine is an audiovisual performance, in which I build nanomachines in public, using self-designed software called the nanomachine, my instrument. A nanomachine is composed of an ensemble of objects, each with its own shape, sound and autonomous behavior, which influences its neighbors, thus yielding a complex semi-autonomous rhythmic machine sensitive to control. My computer screen is projected and its audio is sent to the sound system, so that the whole building and exploration process is accessible to the public, making it a spectator experience. The audio signature is one of an ambiguous rhythmic sample loop; the visual aspect is minimal and abstract.

http://www.gratin.org/as/nanos/

---

chdh

Cyrille Henry and Nicolas Montgermont

chdh developped a body of work from theoretical principles which makes the creation of a live, real-time, audio-visual performance based on about thirty instruments possible. These instruments are made of generative, stochastic, or physical modeling algorithms and each of them control a sound and a visual. With mathematical algorithms as well as physical modeling for the real-time generation of control data for an audio/video synthesis, chdh brings a new vision of the use of data-processing tools in musical creation. The use of instruments having an audio component and a video component, controlled by the same parameters, allows for an effective management of the audiovisual relations.

This project required the development of a virtual world of more or less autonomous abstract creatures. During a performance, chdh plays with these “instruments”, in order to make them react both visually and aurally. Two musicians, connected through a network, interact on the same interface with motorized MIDI faders. Each instrument, or “patch”, can then be played by one of the two musicians or by both simultaneously. They handle the instruments by using an abstract layer, which modifies the parameters of the algorithm. Each algorithm then creates data used for the synthesis of video and sound, creating a strong cohesion between the two media. The video and sound aesthetic is minimalist: sines, diracs and noise interact with cubes, spheres, and other primitive 3D forms in a black and white environment. The different instruments make the creation of a solo-accompaniment musical structure possible; letting the audience discover the intrinsic bonds between image and sound as well as creating polyrhythms by playing on the visual and sound space granted to each instrument.

http://www.chdh.net/

I.D.L. (Icon of Desire Lost)

Alice Daquet

The media, consumer society, technological advancement, medical progress have built an image of perfection that is now unattainable for the common people. On stage, the perfect woman, represented by a plastic figurine, is smiling, obliging. Each gesture she makes triggers sound, music that is perfectly digital, insipid. This music cannot evolve because it is associated with the morphological constraints of a plastic figurine. Then, suddenly, the instability of her body can be heard; humanity arises. Her gestures are less and less anticipated, she leaves her plastic figurine yoke. The sound related to her gestures grows richer, becomes alive and betrays the impossibility of a human being a machine. When she remains motionless, the tremor - unperceivable with the eyes - deforms the sound and thus becomes perceptible through the sound. Exhaustion slows down the sonic rate of a repetitive gesture. The step becomes more feminine; the legs fold, the head falls, and this woman appears like a puppet, held by her own fantasies of perfection, dependent on the glance of the others, on the phantasms of the others. A female victim of a perfect world's desire, a stereotype world, perfectly unattainable. A contrast between fantasy and reality, translated by a work on sound and visual opposition. A dialogue between the man and the machine he is, bringing the sound to various poles of infinity of possibilities.

1ProVisTruments

Cyrille Brissot, Emilie Simon, Olivier Pasquet, Cyril Hernandez

The hand of man, of a potter, a glass blower, or a gardener has been able to make his creations tangible using different objects, tools, and instruments. The process of writing a musical piece for a specific instrument contrasts greatly with digital music, which was born missing one facet, a control tool: the instrument. This missing element can make the user feel frustrated; but it also leads to a new breed of musician. A new breed in that they are no longer tied to a specific instrument, but to their own specific needs, both in terms of the control of the musical process and in terms of their needs on stage.

Everything that communicates with the computer has the potential to become a controller and, therefore, a part of the instrumental practices of the user. Any object can be used as an interface. Any user can become, to a certain extent, an artist. The unintended use of an interface causes a certain number of “happy accidents” to come into play during the creation of new work. The ability to assimilate new technology, sometimes in an unexpected way, maybe even in an illicit way, lets us invent new forms.

Installations

Acousmeaucorps

Tom Mays

IRCAM, Entrance Bridge

Acousmeaucorps (pronounced "acous-mo-cor") is an interactive sound installation that creates an acousmatic body space using a video camera, a computer, and four speakers. A video camera (situated above the space and facing downward) is connected to a computer running Max/MSP/SoftVNS which uses movement and position data to generate spatialized sound. The human body thus becomes a performance instrument, generating and triggering sounds which build musical sequences through walking, running, making arm movements, or even just flexing one's fingers.

For the current version of *Acousmeaucorps*, the sounds are played on two levels. One is a resonant "mass" that seems to move like water in a wading pool, favoring different pitches depending on the area of movement. The other is the triggering of different "found objects" that seem to jump out of very specific locations within the space.

Together, the sound types encourage people to shed their inhibitions and enjoy searching with movements that become fluid and questioning, the idea of "body" and "space" taking on new significance.

Co-production and computer/audio equipment by La Grande Fabrique, Dieppe.

http://www.tommays.net/

---

anemo

Ryoho Kobayashi

IRCAM, Level -1

Anemo is a musical interface using floating balloons.

The source sound for this work is taken from ping-pong balls rolling on an electric fan. The sound is then processed by the movement of floating balloons. The colors and the movements of the balloons are tracked by a video camera and a computer.Each color creates its own effect, and the performer can control the movement by moving their hands and changing the wind from the electric fan.

Métier à Tisser Musical – The Musical Loom

Kingsley Ng

IRCAM, Level -1, Espace CE

Transformation of a 250 year-old loom into a sound and image instrument.

The work was created in the context of northern France, where the loom has played a very important role in the region's economic development and recession during the last century. Its mechanical motion, its sounds, and the flow of the threads do not only evoke an industrial past but also a whole set of collective emotions that range from poetic to distressing, depending on the person. It is in a way comparable to Japanese Haikus, where minimal words and syllables can generate a magnificent array of images.

At the same time, the loom is also of remarkable global significance to the history of new media art. Art historian Lev Manovich cites a remark made by Ada Augusta, the first computer programmer, who said: "the Analytical Engine weaves algebraical patterns just as the Jacquard loom weaves flowers and leaves... The connection between the Jacquard loom and the Analytical Engine is not something historians of computers make much of, since, for them, computer image synthesis represents just one application of the modern digital computer out of thousands. But for a historian of new media it is full of significance." Here, this readymade from 250 years ago speaks individually to each spectator, each one having the freedom to make his own connections with local or global history and to weave a unique soundscape based on his experience with the work.

Production: Le Fresnoy, Studio national des arts contemporains, Tourcoing France. Francis Bras, Jean-Pierre Courcelle, Jean-Baptiste Droulers, Isabelle Bohnke. Special Thanks to: Richard Campagne, Matthieu Chéreau, Alain Fleischer, José Honoré (Musée du Jacquard), Joelle Pijaudier, Pascal Pronnier, Blandine Tourneux.

16:9 intercreative sound installation

Daniel Teige and Martin Rumori

IRCAM, Level -2, Wall

16:9 is an intercreative sound installation. Using a wireless, portable interface, sounds can be mixed and freely spatialized on a speaker canvas of several square meters by painting with colors on a handheld touchscreen. The virtual painter is able to move freely in front of the canvas to interact with the acoustic painting in a tactile and playful way. 16:9 offers a projection screen for a user-generated audio painting.

Sixty-four independent speakers installed as a pattern in a large white field comprise the projection screen. Adapted to the architecture of the space, the speaker matrix is designed to be fixed on a wall like a painting. The array of speakers offers a precise placement and radiation of sound. Realistic depth effects as well as great freedom in sound positioning are easily attainable and the basis for the intercreative audio painting.

Sound recordings made in the exhibition city become source materials to produce the different sound/color textures. By exploring the city in which the installation is shown the composer seeks the character unique to the venue surroundings. By changing the character of these sounds or colors in several degrees of abstraction it is possible to experience a synaesthetic urban situation.

Perceiving the « 16:9 » means to reflect and discover an artificial situation by creating an audible image. Virtuality gets interspersed by reality. Acoustical perspectives, movements and situational moments form the base for this audio image. The visitor gets integrated as the source of action in the installation. Without action no sound will appear on the canvas. By deciding how to mix and place a color the given result will always be a unique interpretation.

16:9 is sponsered by: Apogee using Apple Computers, der Senatsverwaltung für Wissenschaft, Forschung und Kultur, Wolf and Elisabeth Teige, Dipl.Ing. Folkmar Hein, Thomas Seelig, Michael Hoeldke and Dipl.Ing. Manfred Fox.
Very special thanks to our producer Catherine Mahoney in New York, Dipl. Ing. Osswald Krienke from Digital Audio Service in Berlin and Hanjo Scharfenberg / Galerie Rachel Haferkamp in Cologne.

codespace_<tag>

jasch

IRCAM, Level -2, Assistant's Studio

The focal point of this installation lies in the relationship between the visitors and an abstract audiovisual world, the intersection between the space of a person in a room and an imaginary geometrical and acoustical space.

Processes: An abstract world is created in real-time by the means of generative drawing and electronic sounds. The evolution of the visual and acoustical processes depends on interaction with the visitors but is also partly autonomous, using information gathered from various sources on the net. The images represent an imaginary space in which relationships between entities are visible. It's a shifting scenery with a strong graphical look; like a painted film.

Interaction: The interaction gives the visitor the opportunity to interact directly with the process by means moving in the installation. The interactive system observes the exhibition space with a camera and then extracts information from the visitor's movements and behavior to control sonic and graphical processes.

Media: Two layers of media appear in the installation: one consists of prepared footage and audio-recordings from urban and architectural spaces. The other is images and sounds from the actual gallery space. These are mixed, modified, and projected as fragments and textures into the imaginary space.

Space: The installation space is modified in such a way as to guide the visitor's attention from the actual to the imaginary space. Two images -- one large, one small -- are present, as well as an immersive surround sound system. A semitransparent screen divides the space, creating a path, which the visitor can take, from an "outside" with a single view to an "inside" where all elements of the installation are present.

http://www.jasch.ch/

LINE

Daisuke Kobori, Kojiro Kagawa, Makoto Iida, Chuichi Arakawa

IRCAM, Level -2, Entrance Studio 6

LINE is an interactive installation that enables a synchronous expression of sound and light. Audiences can simultaneously control sound and a three-dimensional light object, which appears in a cylindrical display. By moving a handheld control device in the air, audience members can experience a harmonious expression of sound and light. Once the audience member swings his arm up, the light object and sound emerge in space. When he swings his arm up again, the light object vanishes without a trace.

Audio Shaker

Mark Hauenstein, Tom Jenkins

IRCAM, Level -2, Corridor

The Audio Shaker explores our perceptual understanding of sound. Anything sung, spoken, clapped, whistled, or played near it is trapped inside, where it takes on an imagined yet tangible physicality. Sounds caught in this void are transformed, given weight and permanence, reacting directly to the shaker's movements, subtle or violent. Shaken sounds have to settle down before becoming still and silent, behaving more like fluid than transient energy.

The linear timescale of sound is broken, a conversation is split into words and mixed up in the shaker, and can be poured out separately, tipped out in a simultaneous splash or added to and shaken up further.

Put simply, it is a tactile container to capture, shake up and pour out sounds. Creating a rich, intuitive experience that is purposefully open to interpretation and imagination.

http://www.nurons.net/

Sonobotanic plants

Marije Baalman and Alberto de Campo

IRCAM, Level -3, Stairs

IPSO-FACTO
Institute for Predictive Sonobotanics – Foundation for the Auralisation and Computation of Transient Objects

Sonobotanics is still a widely unknown science; it studies plants whose life experience is predominantly in the auditory domain. Since the 1970's Dr. Hortensia Audactor has carried out the core research in this area. Despite difficulties encountered in the publication of her results, she has collected a substantial body of research about the growth patterns, communication behavior, and other characteristics of these plants.

Recently, the field of Predictive Sonobotanics has been founded, attempting to create models of the plants with the aim of predicting the behavior of sonobotanic plants and to gain a deeper understanding of the subtleties in sonobotanic plant behavior.

In the exhibition models of the Periperceptoida Dendriformis Sensibilis and the Periperceptoida Dendriformis Imaginaris are presented.

http://www.sonobotanics.org/

Transduction.2

Marc Fournel

Level -4, Foyer Espace de Projection

Transduction.2 is an immersive environment that makes use of spherical electronic interfaces. Five balls lie spread about here and there on the floor. Visitors pick them up to trigger spatial sound diffusion and to interact the environment.

Sounds are only produced when an interface is being moved. The distribution of the sounds is determined by the position of the interfaces in the space. The sounds "follow" the interactors as they move in the space. The system also allows users to "throw" sound from one speaker to another by simply making a movement to throw the spheres toward the chosen speaker. Interactors can control the volume of the sound according to their position in the space. Transduction.2 use three types of sound: sound samples, sound produced by algorithms and, live sound introduced by the interactors in the system.

Interactors have the possibility of introducing sounds into the system via a miniaturized microphone inserted into one of the interfaces. This allows interactors to input spoken words, breathing, music, rhythms or any other sound proposition. By moving the interface, they modify their input sound. The input of live sounds allows users to influence Transduction.2's sound environment in a personal way.

The Local Positioning System (LPS) used in Tranduction.2 was developed by Ubisense..

Index