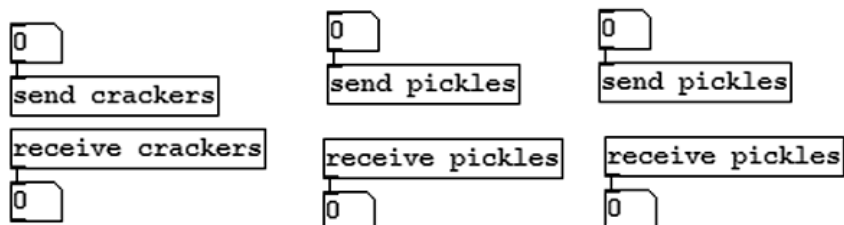


The send and receive objects allow you to make non-local connections. These work globally--you can use them to make two different patches intercommunicate if you wish. Any message a "send" gets appears at the output of every receive of the same name. There can be any number of sends and receives sharing the same name:



You can use the semicolon feature of message boxes to address receives, too. This is useful if you want to do a whole list of things:

```

;
pickles 99;
crackers 56

```

The transaction takes place in zero time---i.e., if you tried to use "timer" to measure the time delay between the two, you would get zero.

send and receive can be abbreviated:

```

s crackers
r crackers

```

updated for Pd version 0.34

Building Your Own Instrument with Pd

Hans-Christoph Steiner

1. THE PROBLEMS WITH LIVE COMPUTER MUSIC

With the power that even a cheap laptop can provide, the computer has gained widespread acceptance as musical tool. Composers have been creating music using computers for more than 40 years now, and even music created with analog instruments is generally recorded and mixed on computers. More and more musicians are using computer-based instruments for live performance, to the extent where you can see live computer music in just about any major city in the world. There is a wide range of music software specifically designed for live performance, such as GDAM⁵, Ableton Live¹, SuperCollider⁹, Max/MSP¹¹, and Pd. Although these tools can provide an engaging performance environment, the live performance leaves something to be desired and is often indistinguishable from someone reading their email. Such performance lacks physicality in the interaction and is quite limited in the range of possible gestures.

Audio synthesis has freed instrument designers from the constraints of the physical method of generating sound, thus any interface can be mapped to any synthesis algorithm. For example, a guitar's strings are both the interface and the sound generator, while a MIDI keyboard can control any MIDI synthesizer. This flexibility allows musical instrument designers to choose their interface without the constraints of the method of sound generation; the interface need not even be physical. Consequently, a multitude of means of translating gestural input from the human body are readily available. By combining such gestural input methods with Pd, a broad range of people can now make their own gestural instruments.

Yet almost all computer musicians have bound themselves to the standard keyboard/mouse/monitor interaction model. To provide an engaging performance, musicians need to move beyond what the basic laptop offers. The human body is capable of a great range of gestures, large and small, communicating emotion in a manner similar to music. There are many distinct and some universal human gestures that are well established and easily understood. Since music is about expressing and communicating, using a broader range of gesture enables the performer to have a broader range of expression. Computer musicians should not be limited to the small set of gestures that a normal computer interface can capture. In order to expand the musician's interaction with the computer, other input devices are needed. Many music software environments are already capable of using data from Human Interface Devices (HIDs) such as joysticks, drawing tablets, gamepads, and mice.



Figure 1: “The gesture is embedded in the music”[10] (Blue Vitriol [3], (c) Patina Mendez; Hazard County Girls, (c) Larry Stern)

Performers of live computer music generally stare at the screen intently while performing, rather than interacting with the audience. What the performer is staring at is obviously important, judging by the intensity of the stare. However, what the performer is looking at is out of view for the audience. This is in stark contrast to traditional musical instruments, where the instrument is generally in plain view, and the audience is familiar with the mechanisms of that instrument. The absence of these two qualities further alienates the performer from the audience. This alienation can be alleviated when the musician can perform using an expanded range of gestures. The performer can come out from behind the computer screen, bringing back a closer connection between audience and performer. Michel Waisvisz’s “The Hands” is a great example of such an interface. Built in the early eighties, it has been used to control a variety of different sound synthesis schemes. It is a novel interface that he has played for 20 years, achieving virtuosity. It allows him to stand on stage with nothing but “The Hands” and use gestures large and small to control sound and compose in realtime.

Haptic feedback is another element that is missing from the keyboard/mouse/monitor interaction in contrast to traditional musical instruments. “Haptic” means “relating to the sense of

touch to the skin and the sense of forces to the muscles and joints”. Traditional instruments provide haptic feedback because the interface is producing the sound itself, so the vibrations can be directly felt. Practiced musicians rely heavily on this feedback, often correcting mistakes by feel before hearing them. During performance, computer musicians are obviously using feedback beyond just listening to the music, the intensity of their stare at the computer screen is a measure of this. They are relying on visual feedback that the software provides via the screen. Providing haptic feedback means the performer can rely less on the visual feedback and instead engage the audience.

Pd is a fertile platform for creating live musical performances in an environment that is accessible to a wide range of people with varying skill levels. It is a unified platform for a broad range of activities, combining realtime synthesis and manipulation of both audio and video, physical modeling, and more. Pd provides many options for data input/output including MIDI, networking, USB HID, and general serial communications. Since Pd is free software that runs on most operating systems, even musicians with very limited budgets can build their own computer music instruments. Up until recently, computer music has been out of reach to all but a select few. It is now possible to create an instrument using Pd that costs less than most traditional musical instruments, including the cost of the computer.

2. THE FUNDAMENTAL BREAKDOWN

When approaching instrument design, the overall problem can be broken down into input, output, mapping, and feedback. The idea of input is straightforward: the data used to control the instrument. Output is also a simple concept: the desired result of playing the instrument. Mapping is a more complex idea: the processing and connecting of input data to parameters which control output. And last but not least, feedback is communication generated from the input, output, and/or mapping data.

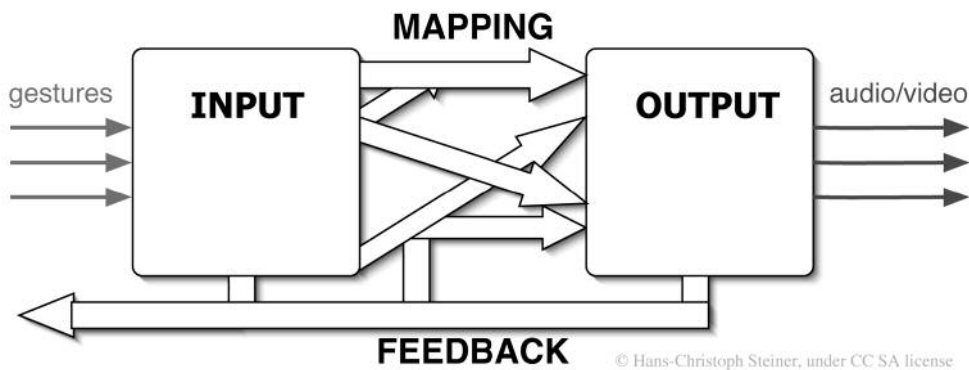


Figure 2: The fundamental breakdown of instrument design.

3. GETTING INPUT DATA

3.1. *Human Interface Devices*

When talking about interacting with computers, “HID” has become the standard term for devices designed to control some aspect of a computer. A wide range of devices are classified as HID, including standard computer devices like mice and keyboards, as well as gaming devices like joysticks and gamepads, to devices for more specific needs like drawing tablets. Pd now has a unified method for getting data from HID: the [hid] toolkit. Consumer input devices like mice and graphics tablets are affordable and readily available and have a lot of potential as musical controllers. They are familiar to most contemporary concert audiences. Using HID in performance therefore has the potential for making the experience much more understandable to the audience. There are a number of examples of contemporary musicians who have mastered using a standard HID as a musical controller: Gerard Van Dongen tours with his Saitek force feedback joystick¹² controlling Pd; Hans-Christoph Steiner has performed live with StickMusic¹⁷, built using a force-feedback joystick and mouse.

3.2. *Sensors and Microcontrollers*

There is a huge variety of sensors, switches, buttons, displays, and electronic devices readily available, from force-sensitive resistors to accelerometers to infrared proximity sensors. Building from individual parts allows the designer to tailor the controller closely to his desires. Recently, there has been a surge in the development of various sensor boxes which allow users to easily get data from various sensors into their computers. The Multi/O-Box⁸ is the easiest way to use arbitrary devices for input, converting sensor data to USB HID and MIDI. Such sensor boxes convert analog signals to digital signals, making them very easy to use within Pd. Microcontrollers such as the Microchip PIC⁷ or the Atmel AVR² have become a popular method of getting sensor data into computers. They are cheap and run fast enough to track the output of an array of sensors. The downside is that a solid knowledge of electronics is needed to create reliable instruments. Also, many microcontrollers are too slow to provide good resolution.

3.3. *MIDI Equipment*

A wide variety of controllers use MIDI to communicate. MIDI guitars and breath controllers emulate traditional instruments but are usually a poor facsimile. The Kaos Pad⁶ is a more esoteric controller, which can be used within Pd with Derek Holzer’s Kaos Tools¹⁴. There are many variations of the mixing board, known as MIDI “control surfaces”, which provide anything from rows of basic sliders to large consoles with sliders, knobs, buttons, etc. They generally are reliable and designed for musical applications, making them a natural choice for a musical controller. Nick Fells uses MIDI control surfaces in a number of different instruments.

His pieces “Words on the streets are these” and “Still Life”¹³ are two examples. He uses the Peavey PC1600x control surface, mapping each slider to various parameters to be directly controlled in realtime. Since the roots of Pd lie in MIDI, it is very well supported. MIDI devices are generally low latency, but the MIDI protocol itself is designed around 7-bit resolution with some 14-bit resolution devices. 7-bit is a quite limited range for a musical controller, especially compared to other devices like USB tablets and mice.

3.4. *Video*

Computers that can do heavy video and graphics processing are now quite common. This opens up the visual dimension to the musician in a whole new way. Motion, color, and blob tracking using video processing allows all sorts of interactions that previously would have been quite expensive and difficult to implement. By extracting data from live video streams, a wide range of gestures can be captured and mapped as the instrument designer sees fit. There are three key methods of tracking gestures with video: color, motion, and shape. The most common one is using motion detection. With Gem, you can use [pix_movement] in combination with [pix_blob]; PDP provides [pdp_mgrid], which is grid-based motion tracking; GridFlow provides motion detection by subtracting the previous frame from current frame using [@-]. For color and shape tracking, PDP provides [pdp_ctrack] and [pdp_shape] respectively. Another option is to process the visual data using outside software and feed that data into Pd. reacTable¹⁶ takes that approach, using OSC to communicate between the two pieces of software.

4. *Mapping*

In the same way digital synthesis has freed instrument design from the constraints of the interface generating the sound, instrument designers are also free to design the mapping between the interface and the synthesis separately from the design of the input and the output. Thus any arbitrary interface can be mapped to any given synthesis algorithm; indeed the mapping can also be designed to suit the goals of the designer¹⁵. Most input devices produce linear data, but mappings in expressive instruments are rarely linear. Sensors often have arbitrary curves which don't make sense in the context of a given instrument. More complex mappings usually create more engaging instruments. There are many common ideas that are frequently used when designing a mapping. For example, since humans perceive loudness and pitch on a logarithmic scale, the amplitude and frequency control data are generally mapped to a logarithmic scale as well. In most compelling instruments, the mappings are not one-to-one between input and output. Certain input parameters usually affect more than one output parameter. For example, how a guitarist plucks a string affects both loudness and brightness.

5. FEEDBACK

In standard computer music performance environments, the screen is the sole source of feedback besides the audio itself. The software interface provides visual feedback, usually in a very concrete manner: by displaying the status of various parameters with virtual knobs, sliders, or even just numeric values. Since non-auditory feedback can greatly enhance the interaction of human and computer, such feedback should become a standard part of instrument design. Adding feedback also allows the musician greater control over the output. The feedback can be in the form of haptics, voice alerts, visuals, or even smell or taste, generated from data coming from any part of the system: input, output, or mapping. Since video synthesis and control is part of Pd, generating novel visual feedback could provide more real-time information to the performer while adding another dimension to the performance. By providing haptic feedback through the physical interface, the need to stare at the screen can be alleviated. Haptic devices have become readily available and affordable. There are numerous gaming HIDs, such as joysticks, gamepads, and mice, which can provide a range of haptic ‘effects’ from vibrations to forces to friction. Since the motor control in these haptic controllers has been encapsulated into haptic ‘effects’, they are generally quite easy to control. The [hid] toolkit provides a number of objects for generating haptic ‘effects’ such as [hid_ff_periodic] or [hid_ff_spring]. They follow the same conventions as the rest of the [hid] toolkit, so they should easily interoperate with the whole set of mapping objects.

6. A NEW MODEL OF INSTRUMENT DESIGN

A new model of instrument design is emerging, shifting away from instruments designed for a broad user base, such as the Theremin, the MIDI keyboard, and the vast majority of traditional instruments. Instead many instrument builders are using systems of building blocks that allow them to create their own instrument relatively easily and quickly. This has contributed to a shift in the idea that a musical instrument should be a device for playing a wide range of pieces. Individual musicians can create their own instrument tailored to their performance goals, or even tailor an instrument to a specific piece or performance. One great advantage of the old model of instrument design is that musicians can develop and share a body of knowledge about how to play that instrument. This is something that has been severely lacking in the world of new interfaces for musical expression: it is rare for anyone to achieve virtuosity on these new instruments, even among the designers themselves. Using standard interfaces such as joysticks and tablets allows people to build shared technique without sacrificing the ability to specifically tailor the instrument via the design of the mapping and the output.

7. GESTURAL VIDEO

This text has been about musicians, since traditionally, gestural instruments have been used to create music. But this is no longer the case: the newfound power of the computer has opened up visual synthesis to gestural control. Some artists are starting to control visuals with gestures, breaking away from the standard on-screen mixing interfaces. A performer can control tightly synchronized audio and visuals generated from a real-time gestural interface. *chdh*⁴ is an example of this kind of work, combining three-dimensional visual elements linked with matching timbres and themes. All of this is controlled in real time using MIDI control surfaces by the two brothers that make up the group. In conclusion, the power that the computer provides makes new forms of musical expression possible and accessible to almost every person. With environments like Pd, the instrument designer no longer needs to spend decades learning the trade, but instead can start experimenting quite rapidly, yet still spend decades perfecting their skills.

REFERENCES

- ¹ Ableton Live. 11 Dec. 2005 <<http://www.ableton.com/index.php?main=live>>.
- ² Atmel AVR. 11 Dec. 2005 <<http://atmel.com/products/avr/>>.
- ³ Blue Vitriol. 11 Dec. 2005 <<http://bluevitriol.com>>.
- ⁴ *chdh*. 11 Dec. 2005 <<http://chdh.net>>.
- ⁵ GDAM. 11 Dec. 2005 <<http://gdam.ffem.org>>.
- ⁶ Korg Kaoss Pad. 11 Dec. 2005 <http://www.korg.com/gear/info.asp?A_PROD_NO=KP2>.
- ⁷ Microchip PIC. 11 Dec. 2005
<http://microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=74>.
- ⁸ *multI/O-Box*. 11 Dec. 2005 <<http://multio.mamalala.de/>>.
- ⁹ *SuperCollider*. 11 Dec. 2005 <<http://audiosynth.com>>.
- ¹⁰ B. Buxton. Nime05 keynote speech. 2005.
- ¹¹ Cycling '74. Max/MSP. 11 Dec. 2005 <<http://cycling74.com>>.
- ¹² Dongen, G.V. 11 Dec. 2005 <<http://www.xs4all.nl/~gml/>>.
- ¹³ Fells, N. *On space, listening and interaction: Words on the streets are these and still life*. Organised Sound 7, no.3 (2002): 287-294.
- ¹⁴ Holzer, D. *Kaos tools*. <<http://puredata.org/Members/derek/kaos.tools.zip>>.
- ¹⁵ Hunt, A. / Wanderley, M. / Paradis, M. *The importance of parameter mapping in electronic instrument design*. Proceedings of the Conference on New Interfaces for Musical Expression (NIME02). Dublin 2002.
- ¹⁶ Kaltenbrunner, M. / Geiger, G. / Jordà, S. *Dynamic patches for live musical performance*. Proceedings of the Conference on New Interfaces for Musical Expression (NIME04). Hamamatsu, Japan 2004.
- ¹⁷ Steiner, H.-C. *Stickmusic: Using haptic feedback with a phase vocoder*. In Proceedings of the Conference on New Interfaces for Musical Expression (NIME04). Hamamatsu, Japan 2004.