EYE.BREATHE.MUSIC: A MULTIMEDIA CONTROLLER THAT USES EYE MOVEMENT AND BREATHING

Sam Bailey, Adam Scott, Harry Wright, Ian Symonds and Kia Ng

ICSRiM - University of Leeds
School of Computing, School of Electronic and Electrical Engineering, and School of Music,
Leeds LS2 9JT, UK
ebm@icsrim.org.uk, www.icsrim.org.uk

ABSTRACT

The Lumiselo is an instrument that is operated through moving the eyes and breathing through a tube. A camera tracks the coordinates of the user's pupil; a pressure sensor outputs a respective voltage according to how hard the user breathes into a closed pipe and a display communicates this back to the user. This paper performs background research, discussing the creation of the instrument and critically evaluates it on completion.

1. INTRODUCTION

The aim of *Eye.Breathe.Music* was to create an electronic musical instrument that uses small, unobservable movements. The interface, called the Lumiselo, is played through moving the eyes and breathing through a PVC tube.

The eye operates what note is played. This is done through a camera tracking the coordinates of the user's pupil. Pre-set coordinates represent specific pitches, which correlate with the user's display. This allows the user to choose what note they wish to play through looking at a known position. How hard the user breathes controls the volume of the note. This is created using a pressure sensor detecting the intensity of the user's breathing, which outputs a corresponding voltage. This voltage level is then scaled accordingly, to produce an appropriate dynamic range.

The Lumiselo is effectively an electronic wind instrument, replacing the hands with the eyes. Consequently, users will have to learn to play the instrument just like any other. However, the fact that it is electronic provides certain advantages; users will be able to alter the dynamic range so it is appropriate for them and they will be able to change what sound they are producing. The instrument is effectively a controller, therefore users will be able to alter the instrument to play in the diatonic scale instead of chromatic, making it less demanding upon musical ability.

It is hoped that the interface will give users as much control over musical elements as existing instruments, but through more confined movements. Part of fulfilling this aim involves adapting the instrument, so that the user can control musical elements such as phrasing, in an emotional manner and perform techniques such as glissandos and vibrato.

2. BACKGROUND RESEARCH

Oculog: Playing With Eye Movements [8] describes a system used for performing electronic music that uses eye movement to control sounds. In this paper four types of eye tracking are discussed. Electrooculography (or EOG) measures the small potential difference between two electrodes placed on both eyes. Assuming that the rested potential is constant, the recorded potential is a measure of the eye position.

Another technique mentioned uses magnetic search coils. By embedding magnetic search coils in a contact lens, voltages are given off by the surrounding magnetic field. The disadvantage of the first two methods, even though they are extremely accurate, is that they involve placing components on the user's eye.

Video based eye tracking involves a high-speed camera (up to 250 frames per second), which can record the eye position in real time. The disadvantage of this method is that a camera of this speed is expensive.

Infrared tracking uses a camera to measure the reflection of an infrared source on the cornea. Although it is not as efficient as the first two methods, this form of tracking allows the camera to be mounted onto goggles or glasses, which the user is free to attach and remove. Another system that uses this method is the *EyeWriter* [5]. A remote CCD camera lens is placed on a wire, which is attached to a pair of glasses. As people's facial features vary, this design allows the position of the camera to be altered so that the cornea is located in the centre of the frame.

The Pipe [9] is a musical instrument modeled around the design of a wind instrument. The user blows into a pressure sensor within the pipe and uses their fingers to control force sensing resistors, two switches, two potentiometers and an accelerometer. *The Pipe* uses static airflow as opposed to a dynamic free-flowing system that traditional wind instruments and current MIDI wind controllers use. The static pressure system is more hygienic as it does not require a humid airflow, meaning that there is a decreased chance of electrical components getting wet. As no air is escaping between the contact of the pipe and the user's mouth, it is possible to maintain normal breathing through the nose whilst playing, eliminating the need to pause for breath. However, for a wind player, the change from dynamic to static breath pressure could seem unnatural. The pressure sensor is attached as an input to an analogue to digital converter, then through a Parallax Basic stamp IIsx microprocessor that converts the signal into MIDI.

3. DESIGN AND IMPLEMENTATION

All the elements of the full system are mounted onto a single headset. The prototype is built up of a camera, goggles, an aluminum frame, (which holds) a visor, (which holds) various LEDs and a PVC pipe.

The camera is attached to the goggles with a metal bracket to prevent any unwanted movement. The goggles provide protection for the eyes and make the system more structurally sound. The display has forty-eight regular LEDs and one infrared LED.

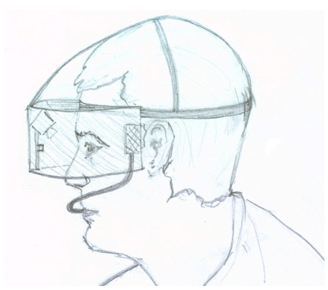

Figure 1. Drawing of a conceptual diagram.

3.1 Testing the Eyes

The Lumiselo was constructed around the features of the average user. Recordings were taken of the eyes moving from left to right at different speeds. As both eyes move in sync, a decision was made to track only one of the user's eyes.

Recordings were taken of numerous subjects reading out the numbers 1-12 from a piece of paper. This test showed that the eyes move a sufficient amount between each point allowing the difference between each point to be detected. It also showed that beyond a certain point, the nose gets in the way and causes one eye to stop moving. This further reinforced the idea of only tracking one of the user's eyes. Furthermore, the results showed that if the eye moved quickly from far left to far right, the USB camera would not be fast enough to follow it in a continuous stream. Consequently, a faster connection from the camera to the computer would be needed. Research showed that a rate of 20 frames per second would be sufficient; however a firewire connection would provide the best results.

Before designing the display, the average user's field of view had to be discovered. A user's field of view is determined by how far their peripheral vision allows them to see to the side. Ideally the display will comfortably be within everyone's field of view. In order to test this, several subjects were asked to close one eye and indicate how far to each side they could focus on comfortably. This was recorded at a distance of 10cm, 20cm and 30cm away from a wall. The test was then repeated for the user's other eye. The results were plotted on a graph and an allowance range was determined. This allowance range was formulated to allow for the possibility of a user having a smaller field of view than any of the test subjects used.

To determine what distance the display should be from the user's eyes, a test was performed to find the average person's near focal length (or near point). The test involved placing a piece of paper with the numbers one to twelve written in a line. The user was asked to read the numbers at various distances. It was found that the average subject's near point was 17cm.

3.2 Designing the Breath Operated Controller

Out of the numerous pressure sensors tested the Freescale MPXV5010 gave the best results. Atmospheric pressure on the sensor created a voltage output of 0.265V; however this will clearly vary with location. The final program must therefore detect this atmospheric pressure level and treat it as the minimum.

Using an empty plastic bottle with the pressure sensor inside, measurements were taken of various users trying to control their airflow. The maximum pressure reached by all subjects was 4.97V, though this was with considerable effort. The comfortable maximum was generally 3V. Users found it easy to keep within a range of 0.05V of a designated pressure level, this allowed for a large range of dynamic values. These tests were then repeated at a later stage with the PVC tubing and provided almost identical results.

Sucking on the pressure sensor was tested as a possible additional control input. Voltages down to 0.035V could be achieved while sucking, but this is only 0.23V below the atmospheric pressure, and with steps of 0.05V there would only be 4.6 velocity values. Thus, sucking was deemed redundant.

Using PVC tubing gives static airflow and allows the user (if they are able) to circular breathe and/or perform techniques such as flutter tonguing.

3.3 Designing the Display

How hard the user is breathing (or how loud the note is sounding) will be shown through the brightness of the LED's. A second PIC16F88 chip is used to control the display by containing an array of values for the brightness of each LED, and outputting these in sequence. The PIC is connected to 3 analogue switches, with 12 outputs each, giving a total of 36. The PIC turns each output of the switches on in sequence, and as it does so, the value of the corresponding LED in the array is sent through the DAC port. The DAC output is sent to the input of the analogue switches, and hence output to the corresponding LED. The human eye is able to detect blinking lights below around 50Hz, therefore each LED must flicker at or above this rate. There are 36 LED's in the display, therefore each value in the array must refresh at a rate of:

$$50\text{Hz} \times 36 = 1.8 \text{ kHz}. \quad (1)$$

Changing the lighting in front of the user's eyes will cause pupils to dilate. This may cause a problem when tracking the pupils; however programming the surrounding LEDs to compensate will solve this. If the user breathes as hard as they can, one LED will be lit on its own, at 'full' brightness, while if the user were to not be breathing at all, all the LEDs would shine at a level of brightness that - when they are all added together – equals that of one LED at 'full' brightness. Therefore the same level of total light will always be emitted by the display. This is accomplished by loading the background value as part of the array used in the display circuit described previously. The value of the background LED's will simply be the maximum brightness level, minus the brightness value for the LED being controlled by the pressure sensor. Using this method, the background LED's will constantly change with the reciprocal value of the selected LED.

There are to be three rows of LEDs, each one being a different octave. Accordingly, if the user looks at an LED, and then subsequently looks at the one below, the instrument will sound a note that is an octave lower. Likewise, if the user looks at an LED directly above the previous, the sounding note will move up an octave.

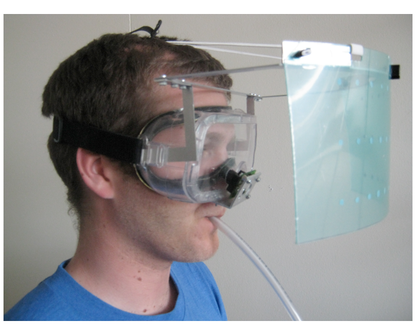

Figure 3. Image of the first prototype.

3.4 Designing the Software

The software is predominately written in Max/MSP/Jitter. C-code was used to program the two PIC16F88 chips used for the analogue to digital conversion on the breath pressure values, and the setting up of an array to output the correct values for each of the LED's.

As each user's eyes will be in a different position on the screen, it is necessary to calibrate the software for each user. Using GUI components the user must click on the calibration button. The user is asked to look at the top left LED, blow, then look at the top right LED and blow, so that mathematics objects can then be used to divide this space into twelve sections for each note of the chromatic scale.

Using the cv.jit.track object in Max/MSP, a program was formed to track a given point. Although successful it was not efficient enough due to the point being tracked often aliasing. Another program was written to track colour, but this also became problematic as the pupil is black and there was spill from the surrounding area of the eye. Research showed that using an infrared LED reflected off-axis into the eye makes the pupil stand out. As the retina does not see near-infrared you can not rely on subjective brightness and precautionary measurements have to be tested before shining infrared LED's into the user's eye. *Babcock* [1] states that an irradiance level less than 10mW/cm^2 is considered safe for IR exposure. A suitable LED was chosen that produced an irradiance level of less.

An infrared LED has very low levels of emission compared to an IR laser and at these levels is not unhealthy for the human eye. With this information a suitable firewire camera was purchased.

4. CONCLUSION AND NEXT STEPS

Both hardware and software are integrated to create the Lumiselo. All the elements of the hardware have been designed and developed into one piece which captures all of the data. The software program, which processes this data, has been developed using Max/MSP/Jitter and allows for two-way communication between the user and the computer. The camera interprets the position of the user's pupils; this is communicated to the computer along with the intensity of the user's breath. This information is then transferred from the computer to the display.

Eye.Breathe.Music allows people to express themselves musically with limited movement. Users can create music with the voice of any instrument, through a combination of moving their eyes and breathing.

Future development will work towards the inclusion of a completely wireless interface as this the logical next step. The user will then be free to move without being restricted by cables.

We are currently working on the validation of the overall system. Numerous quantitative tests have been taken, measuring inaccuracies such as the latency of the system. Various designs have been drawn to create a more comfortable and fashionable headset. A group of musicians have been invited to test out the instrument to provide qualitative feedback.

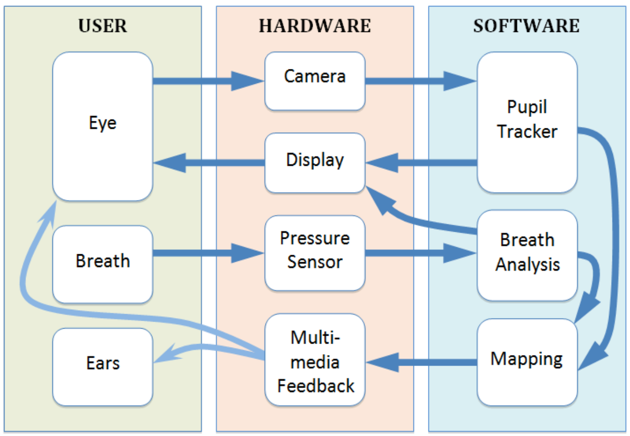

Figure 2. System Architecture Flow Diagram.

In its current state, some parts of the device, such as selecting a voice for the instrument and opening or closing the patch, must be done with the use of a computer mouse. This will be developed so it is completely "hands free".

Although a display gives the user feedback, this still does not emulate the feeling of playing an acoustic wind instrument. Physical vibration is a major part of playing any instrument, because of this, vibrations will be produced electronically and triggered using the breath sensor.

Additional systems could also be built without the use of a headset. Using a projected display and a fixed camera would provide a very different experience. If a computer program could be calibrated to recognize the user's face, the program could take into account the position of the user's head and from this, locate the coordinates of the pupils in reference to the location of the face. The primary function of this system would be identical to the current one; however the user would not be restricted within the confines of the headgear. Whether or not this system would be as rewarding will be shown through qualitative testing.

Carrying on with the current theme of a facially controlled instrument an additional control could be implemented within the system. An extra parameter could be controlled by facial expressions, such as smiling.

The potential for controlling musical systems using the eyes and breathing has room for even further development. Supplementary ideas will be explored such as using it to compose, or draw. Composition could be done through operating a cursor with the eyes and writing notation, or in real time using a system that works like a sequencer. A visualizer could be created from the eye movement that takes place while the instrument is being played. This would provide an audience with something to look at, improving the merit of the instrument being performed live.

We look forward to future work on the Eye.Breathe.Music system, and with succeeding revisions, hope to continue to research and implement further methods of human interaction with music.

5. REFERENCES

[1] Babcock, J. Li, D. Parkhurst, D. (2006) openEyes: a low-cost head-mounted eye-tracking solution. *The Human Computer Interaction Program* Iowa State University, Iowa.

[2] Camurri, A. Coletta, P. Varni, G. Ghiso, S. "Developing Multimodal Interface Systems With EyesWeb XMI", NIME, New York, USA, 2007 pp. 3305-308

[3] Cwol <http://www.cwol.com/firewire/firewire-vs-usb.htm>, December 2009

[4] Ellis, Phil, "The Development of Interactive Multi Sensory Environments For Expressions", in Luxembourg Society of Music Therapy 2006

[5] Eyewriter. http://www.eyewriter.org/ (24 March 2010).

[6] Holly Bank Trust <http://www.hollybanktrust.com/>, October 2009

[7] Hornof, A. J., Rogers, T. Halverson, T. "EyeMusic: Performing Live Music and Multimedia Compositions with Eye Movements" 2007 http://www.cs.uoregon.edu/~hornof/publications.html

[8] Hughes, S. Cannon, C. O Modhrain, S. "Epipe: A Novel Electronic Woodwind Controller," NIME, Japan, 2007, pp 199-200

[9] Juno Kim, Greg Schiemer & Terumi Narushima, "Oculog: Playing with Eye Movements", NIME, New York, USA, 2007 pp.50-55

[10] Scavone, G "THE PIPE: Explorations with Breath Control," NIME, Montreal, Canada, 2003, pp. 15-18

[11] Scavone, G & da Silva, A "Frequency Content of Breath Pressure and Implications for Use in Control," NIME, Vancouver, Canada, 2005, pp.93-96

[12] UniBrain <http://www.unibrain.com/products/visionimg/fire_i_bc.htm>, November 2009

[13] Young Won, S. Chan, H. Liu, J. "Light Pipes: A Light Controlled MIDI Instrument," NIME, Japan, 2004 pp. 209-210