The *KiOm*: A Paradigm for Collaborative Controller Design

Ajay Kapur, Adam R. Tindale, Manjinder S. Benning, Peter F. Driessen

Music Intelligence and Sound Technology Interdisciplinary Centre (MISTIC), University of Victoria
ajay@uvic.ca, art@uvic.ca, mbenning@engr.uvic.ca, peter@ece.uvic.ca

Abstract

This paper describes a paradigm for collaborative controller design, implementation and artistic expression. The KiOm is an inexpensive, robust, accelerometer-based wearable instrument used to expose many artists to the ideas and theory of computer music with the expansive capabilities of human computer interfaces. Novel methods for collaborative controller mapping development using ChucK are described. A variety of performance artists' use of the KiOm is also included.

1 Introduction

It is commonplace in the computer music community for an artist to design a novel controller which only they themselves can use. This does not allow other artists to take advantage of the variety of data obtained from their system. Further, the builder is alone in their pursuit of expression, without the help of other artists struggling with similar problems, ideas, and developmental criticism of that specific instrument. This has motivated the authors to design a controller which is inexpensive and robust, which many artists from a variety of disciplines can use together, while improving and evolving the novel, expressive instrument.

There are a variety of researchers whose work has influenced this project, especially in the area of wearable sensors for musical performance. One of the pioneers is Joe Paradiso and his work with wearable sensors for interactive media (Paradiso 2004). There is also early work using a host of sensor systems such as the BioMuse and bend sensors (Winkler 1995). Head tracking devices using a camera-based approach for musical signal processing is described in (Merrill 2003). Experiments using accelerometers in musical performance are presented in (Sawanda, Onoe et al. 1996; Cook 2001; Hahn and Bahn 2003; Ryan and Salter 2003; Kapur, Tzanetakis et al. 2004), placing them in various parts of the body including the head, feet and hands. The success of the *PikaPika* (Hahn and Bahn 2003) project was a huge influence to our project, in proving that designers should build simple technology in which performers use obvious gestures that the audience can easily relate to and understand.

There have been many successful projects which have built collaborative instruments for multiple performers to play including (Jorda 2002; Fels, Kaastra et al. 2004; Tanaka 2004; Kapur, Davidson et al. 2005; Weinberg, Driscoll et al. 2005) . The challenges and experience of these types of projects is well summarized in (Blaine and Fels 2003). Our goal is to explore the collaborative process on a developmental angle, from controller design, feature selection, sound mapping construction and software innovation.

In this paper, section 0 describes the design of the controller. Section 0 describes the process of redesigning the controller based on multiple user feedback, expanding features and use of the instrument. Section 0 describes a novel approach to collaborative controller mapping. Section 0 describes case studies of multiple users' incorporation of the controller into their own art form.

2 Controller Design

The design of the wearable sensor, known as the *KiOm* (Figure 1), is described in this section. The center piece of this controller is the Kionix KXM52-1050[1] three-axis accelerometer. The three streams of analog gesture data from the sensor is read by the internal ADC of a PIC Microchip 18F2320[2]. These streams are converted to MIDI messages for use with any musical hardware/software synthesizers and programs.

The dimensions of the *KiOm* are 3 inches by 3 inches by 2 inches. It weighs approximately 100 grams. A majority of the space and weight of the device is due to the 9-volt battery used to power the MIDI out port at 5-volts. The *KiOm* also has a power switch with LED to allow the user to know when the device is on/off. Two buttons are also built in for users to have control of different modes/controls/parameters of their compositions.

Figure 1 - *KiOm* encasement and inside circuit board.

[1] http://www.kionix.com/ (January 2006)

[2] http://www.microchip.com/ (January 2006)

3 Collaborative Redesign

In order to achieve our goal of multiple user collaboration, 12 *KiOms* were created and distributed to a variety of musicians, dancers, visual artists and computer music programmers.

In order to foster collaborative development, weekly meetings were held to discuss progress in use of the device. Each week there would be a theme. For example, the first weeks topic was MAX/MSP (Puckette 1991) in which able members discussed how they used the *KiOms* with this software. Following topics included Propellerhead Reason[3], ChucK (Wang and Cook 2003), and Jitter. Another meeting included looking at ICMC papers of the past to see how the community has used accelerometers on stage. One meetings topic was how to mount the wearable sensor on the body which generated solutions including velcro, elastic strips, duck tape and custom made clothing with pockets for *KiOms* and appropriate wires.

A tremendous amount of improvements and upgrades were made to the hardware as a result of these meetings. A simple but useful development was advanced button techniques in which a user could use the two buttons to send a variety of unique messages to control a number of parameters. This included differentiating between one click, two click and three clicks, as well as a hold down function in which time between start and finish is calculated turning the buttons into continuous controls.

Initial designs sent accelerometer messages only if the values had changed. Most users found this cumbersome as control rate was random and certain algorithms could not be implemented. Thus, the *KiOms* were upgraded to send continuous messages regardless of accelerometer values. This allowed some users to play with the idea of integrating to find position as discussed in (Young and Fujinaga 2004). Of course more sensors are necessary in order for position to have any accuracy and to eliminate drift which occurs for errors in two integrations.

A final group-enthused improvement, inspired by the radiodrum (Mathews and Schloss 1989), was to create a "Whack" mode in which quick changes in direction would trigger events similar to drums. These messages were set to the same messages of the radiodrum in order to let the *KiOm* make use of Max Mathews old radiodrum patches including his famous conducting software.

4 Collaborative Mapping

When designing an interface for artists, it is necessary to provide a simple way to write software that is flexible while relatively simple to develop. ChucK is a music programming language developed at the Princeton Soundlab that allows for sample accurate timing, flexible granularity of time and massive concurrency.

Due to the high level nature of ChucK, we were able to rapildly develop code to be distributed to artists. During the hardware manufacturing, a global software interface was designed. The interface handled all of the MIDI data, message passing, event handling and filtering necessary to use the *KiOm* effectively. A simple patch is provided below that receives the input from *KiOm* and maps the X axis to the amount of reverb on the instrument.

```
KiOm k
adc => JCRev reverb => dac;
while(true){
    k.x / 127. => reverb.mix;
    k.next_message=> now;}
```

Code Listing #1: A simple interface with the *KiOm* ojbect.

The artists were then able to develop their music and mappings without having to worry about event handling, as well as not having to write redundant processes for each patch they developed. The artists could start creating sound almost immediately using this interface. The following patch is a more complicated example of the use of the *KiOm* object.

```
//polling function
fun void x_axis(){
    while(true){
        10::ms => now;
        k.x_val /127. => fx.mix;}}
// event listening function
fun void y_axis(){
    while(true){
        k.y => now;
        k.y_val / 127. => fx1.mix;}}
// button 1 advances the notes in a sequence stored in the array notes
while (true){
    k.button_1 => now;
    theNote++ % notes.length() => theNote;
    notes.play(theNote);}
```

Code Listing #2: A demonstration of polling and event driven input.

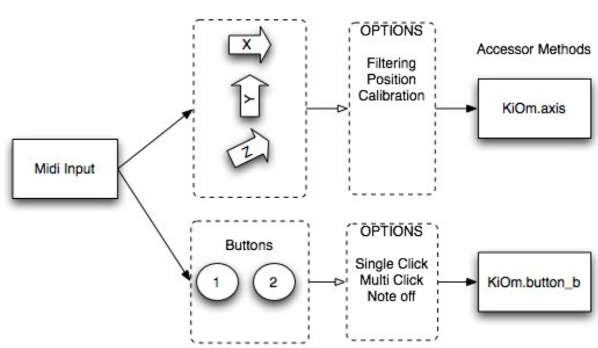

Figure 2 - *KiOm* Class written in ChucK.

[3] http://www.propellerheads.se (January 2006)

Having a separate *KiOm* object (described in Figure 2) allowed the programmers to address technical problems transparently. Accelerometers are very noisy sensors and in all cases the artists were looking for smoother data. Filtering algorithms were placed inside the object to smooth and control the data stream. A calibration feature was included to ground the *KiOm* for each player and environment. A rough estimate of velocity and position were also included by taking the integrals of the incoming data. These features were not used by all of the artists, but having them all wrapped within one object as options allowed the artists to observe the use of these features in the work of their peers and quickly implement them if they had been inspired. ChucK allowed the artists to work in parallel with the algorithm improvements without one creating a bottleneck for the other.

5 Case Studies

This section presents how the *KiOms* have been used to express musical ideas and meaning in a variety of different ways. Methods for preserving traditional technique for almost any musical instrument are described. The ways dancers and conductors have experimented with the *KiOms* is also described. Finally, visual artists use of the *KiOms* is included.

5.1 Preserving Traditional Technique

In the literature presented, sensors are used to drive synthesis algorithms directly, completely separating the sound source from the gesture. One of our paradigms, is to keep traditional instrument performance technique, modifying the amplified acoustic signal with sensor data controlling a number of audio effect parameters. A similar paradigm is that of the hyperinstrument (Machover and Chung 1989) where an acoustic instrument is augmented with sensors. In our approach, the performer wears a low-cost sensor while keeping the acoustic instrument unmodified, allowing for a more accessible and flexible system.

Our first experiment was to place a *KiOm* on the hands of the drummer during performance. The drummer was told to play with traditional technique. Because of the rhythmic nature and movement of the drummer's hands during the performance, using the gesture-captured data to effect the sounds of the drums was successful. Our favorite algorithms were controlling parameters of the comb filters and the flanger. Similar results were obtained by placing the sensors on the feet of the drummer while playing bass drum.

Our next experiment was with hand drumming on the traditional North Indian Tabla. Again, a traditional performance obtained rhythmic gesture capture data which musically combined as parameters to the various synthesis algorithms. Another method was to place the sensor on the head of a performer, so that the performer could sing and control the DSP parameters with head motions, thus leaving the hands of feet free to gesture to the audience. Another method is for the performer to play a traditional instrument wearing the headset, replacing the need for foot-pedals, knobs and buttons to control synthesis parameters. An example where this might be useful is during Sitar performance, in which the musician traditionally sits on the floor, and whose hands are occupied, leaving only the head to control parameterization. This was the initial experiment administered by the first author described in (Kapur, Lazier et al. 2004), which initiated this research.

More experiments include performances with a turntablist who was scratching vinyl records with the *KiOm* placed on the hand, similar to the drum experiments. By the second author's interest, we administered experiments on a computer music performance, in which a performer uses a keyboard and mouse of a laptop, with a *KiOm* to capture gestures to control parameters of synthesis algorithms.

5.2 Dancers and Conductors

Other artists have used the *KiOms* to generate sound using synthesis algorithms to create soundscapes and music from only their gestures. The two most popular artists that used this technique were a dancer who performed a piece called Hopelab and a classically trained cellist who conducted a synthesized version of the Swan.

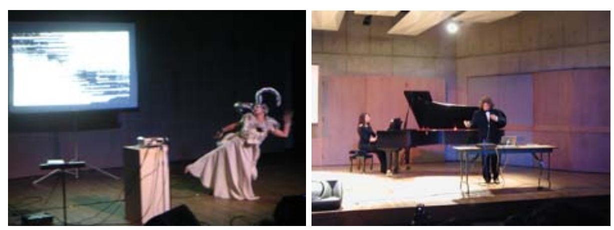

Figure 3 – (left) Erin RobinSong performing Hopelab with two *KiOm* sensors attached to her wrists. (right) Graham Percival performing The Swan with two *KiOms*.

Hopelab. Hopelab is a collaborative visual/music/dance piece in which the dancer wears a specially built costume embedded with midi cabling and two *KiOms*. The 3 dimensions of acceleration of the dancer's wrist movements are mapped to various synthesis controls: The right wrist x and z acceleration to lowpass filter cutoff frequency of two synthesizers; the left wrist z to synthesis noise amplitude, and x to lowpass filter cutoff frequency of a third synthesizer. It was found that the correlated movements of the dancer in space led to expressive parameter control in the virtual synthesis space. A feat that could not be created with conventional keyboard based midi controllers.

The Swan. Another artist used the *KiOm* to play Saint-Saens' The Swan originally composed for solo cello. The artist used two *KiOms*, one in each hand, in order to control the volume and vibrato of the tones. The pitches were sequenced in the correct order and triggered individually by the buttons on the *KiOm*. The artist had been searching for a way in his own playing, as well as in his teaching, to focus on the subtleties of the music such as time, volume and timbre without having to deal with the difficulty of intonation. The KiOm, in conjunction with ChucK, allowed this artist to make a simple and effective manner to realize his vision.

5.3 Visual Graphic Control

Many of the described experiments and performances such as Hopelab also had accompanying visual elements in which the *KiOm* data was used to control various visual effects and synthesis parameters of image/video sequences within the Jitter software framework.

It was challenging to come up with meaningful visual-virtual-space mappings. Our main constraint was to make it obvious to an audience that the movements of the dancer or performer were manipulating the visuals in real-time. Through experimentation it was discovered that zoom, rotation, brightness, and contrast were noticeably and elegantly controllable by wrist mount *KiOms*.

6 Conclusion

Many artists and computer musicians worked together to explore the creative potential and limits of the *KiOm*. Artistic ideas and concepts using the *KiOm* were easily auditioned and tested within the ChucK programming environment providing non-computer musicians an opportunity to explore the creative potential of gesture controlled digital audio effects and synthesis. Through collaborative use by a variety of people, we were able to realize the possibilities that the *KiOm* presents in the discipline of multimedia art and performance.

7 Acknowledgements

Our thanks to Eric Yang for all his help in working with the team to design the *KiOm*. Many thanks to Ge Wang and Perry R. Cook for designing the ChucK programming environment. Thanks to Eric Singer for his expertise on using the PIC microchip. Thanks to Andrew Schloss, George Tzanetakis and the University of Victoria MISTIC team for their support. A special thanks to Jimmy Olson, Justin Love, Erin Robinsong, Arvand Tiv, and Graham Percival for their contributions. Thanks to Tiffany Jenkins for help with photography.

References

Blaine, T. and S. Fels (2003). Contexts of Musical Collaboration. International Conference on New Interfaces for Musical Expression, Montreal, Canada.

Cook, P. R. (2001). Principles for Designing Computer Music Controllers. International Conference on New Interfaces for Musical Expression, Seattle, WA.

Fels, S., L. Kaastra, et al. (2004). Evolving Tooka: from Experiment to Instrument. International Conference on New Interfaces for Musical Expression, Hamamatsu, Japan.

Hahn, T. and C. Bahn (2003). "Pikapika - The Collaborative Composition of and Interactive Sonic Character." Organized Sound **7**(3).

Jorda, S. (2002). Afasia: the Ultimate Homeric One-Man-Multimedia-Band. International Conference on New Interfaces for Musical Expression, Dublin, Ireland.

Kapur, A., P. Davidson, et al. (2005). "Preservation and Extension of Traditional Techniques: Digitizing North Indian Performance." Journal of New Music Research.

Kapur, A., A. Lazier, et al. (2004). The Electronic Sitar Controller. International Conference on New Interfaces for Musical Expression, Hamamatsu, Japan.

Kapur, A., G. Tzanetakis, et al. (2004). Audio-Based Gesture Extraction on the ESitar Controller. International Conference on Digital Audio Effects, Naples, Italy.

Machover, T. and J. Chung (1989). Hyperinstruments: Musically Intelligent and Interactive Performance and Creativity Systems. International Computer Music Conference.

Mathews, M. and A. Schloss (1989). The RadioDrum as a Synthesizer Controller. International Computer Music Conference.

Merrill, D. (2003). Head-Tracking for Gestural and Continuous Control of Parameterized Audio Effects. International Conference on New Interfaces for Musical Expression, Montreal, Canada.

Paradiso, J. (2004). Wearable Wireless Sensing for Interactive Media. First International Workshop on Wearable & Implantable Body Sensor Networks, London.

Puckette, M. (1991). "Combining Event and Signal Processing in the MAX Graphical Programming Environment." Computer Music Journal **15**(3): 68-77.

Ryan, J. and C. Salter (2003). TGarden: Wearable Instruments and Augmented Physicality. International Conference on New Interfaces for Musical Expression, Montreal, Canada.

Sawanda, H., N. Onoe, et al. (1996). Acceleration Sensor as an Input Device for Musical Environment. International Computer Music Conference.

Tanaka, A. (2004). Malleable Moble Music. International Conference on Ubiquitous Computing, Tokyo, Japan.

Wang, G. and P. R. Cook (2003). ChucK: A Concurrent, On-the-fly Audio Programming Language. International Computer Music Conference, Singapore.

Weinberg, G., S. Driscoll, et al. (2005). Haile - A Preceptual Robotic Percussionist. International Computer Music Conference, Barcelona, Spain.

Winkler, T. (1995). Making Motion Musical: Gesture Mapping Strategies for Interactive Computer Music. International Computer Music Conference, San Francisco, CA.

Young, D. and I. Fujinaga (2004). Aobachi: A New Interface for Japanese Drumming. International Conference on New Interfaces for Musical Expression, Hamamatsu, Japan.