WEARABLE SENSORS FOR REAL-TIME MUSICAL SIGNAL PROCESSING

Ajay Kapur *Eric L. Yang* *Adam R. Tindale* *Peter F. Driessen*

Music Intelligence & Sound Technology Interdisciplinary Centre (MISTIC)
University of Victoria, Victoria, BC, Canada

ABSTRACT

This paper describes the use of wearable sensor technology to control parameters of audio effects for real-time musical signal processing. Traditional instrument performance techniques are preserved while the system modifies the resulting sound based upon the movements of the performer. Gesture data from a performing artist is captured using three-axis accelerometer packages that is converted to MIDI (musical instrument digital interface) messages using microcontroller technology. ChucK, a new programming language for on-the-fly audio signal processing and sound synthesis, is used to collect and process synchronized gesture data and audio signals from the traditional instrument being performed. Case studies using the wearable sensors in a variety of locations on the body (head, hands, feet, etc.) with a number of different traditional instruments (tabla, sitar, drumset, turntables, etc.) are presented.

1. INTRODUCTION

New methods of musical expression using modern technology and signal processing are creating a new age of artistic expression. Using low-cost microcontrollers and sensor technology, a musician can now have an armada of signals describing details of gestural movements that can be used to enhance the acoustic audio signal. This paper describes the use of wearable sensors that are used to control parameters of digital audio effects for real-time musical signal processing.

There are researchers in computer music laboratories that have been experimenting with wearable sensors for performance. One of the pioneers is Joe Paradiso and his work with wearable sensors for interactive media [6]. There is also early work using a host of sensor systems such as the BioMuse and bend sensors [14]. Head tracking devices using a camera-based approach for musical signal processing is described in [5]. Experiments using accelerometers in musical performance are presented in [1,2,3,9,10], placing them in various parts of the body including the head, feet and hands. In the literature presented, sensors are used to drive synthesis algorithms directly, completely separating the sound source from the gesture. Our paradigm, and the key novelty of this work, 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 [3,4] 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.

In this paper, the design schematics of the wearable sensor will be described in section 2. In section 3, the software designs created to integrate both audio and gestural streams of data will be discussed, along with a brief description of synthesis techniques administered. Section 4 will describe case studies with a variety of different instruments and techniques, showing how the system can be used in practice and on stage.

2. WEARABLE SENSOR DESIGN

The design of the wearable sensor, named the *KiOm*, is described in this section. A Kionix KXM52-1050[1] three-axis accelerometer is used. The three streams of analog gesture data from the sensor is read by the internal ADC of the Microchip PIC 18F2320[2]. These streams are converted to MIDI messages for use with most musical hardware/synthesizers.

A microphone capturing the acoustic signal of the instrument is also part of the system. This way, synchronized gesture data and audio signals are captured in real-time by the system for signal processing described in section 3. Figure 1 shows a diagram of our system.

[1] http://www.kionix.com/ (February 2005)
[2] http://www.microchip.com/ (February 2005)

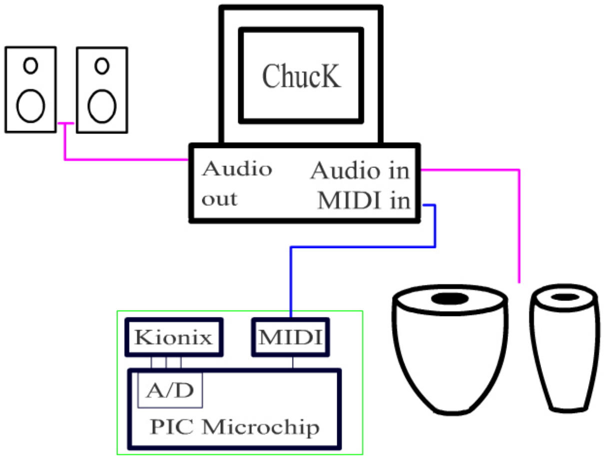

Figure 1. Diagram of synchronized audio and gesture capture system

3. AUDIO SIGNAL PROCESSING

This section will present ChucK, a software used for audio signal processing using both streams of data. The different synthesis algorithms that we use for experimentation will also be described.

3.1. ChucK

ChucK [11,12,13] is a real-time audio signal processing language built for musical applications at the Princeton Soundlab. ChucK is a text-based language that offers expressivity of other text-based languages along with the ease of visual patching languages like Max/MSP [7] and Pure-Data [8]. ChucK is well suited for signal processing applications because of its strong timed nature. ChucK is able to guarantee sample accurate processing of different threads, and allows for these different processes to run at any given sampling rate. Thus, it is easy to capture and synchronize two streams of data sampled at different rates as the case in our design.

3.2. Synthesis Algorithms

For the initial experiments, a number of traditional synthesis algorithms and digital audio effect processors were implemented.

The first algorithm was a FIR comb filter. A FIR comb filter adds a delayed version of the input signal with its present input signal. There are two parameters to tune the filter: T that is the amount of delay, and the g the amplitude of the delayed signal. The difference equation is given by [15]:

$$y(n) = x(n) + gx(n-M)$$
$$M = T / f_s$$
$$H(z) = 1 + gz^{-M}$$

The acceleration data from the wearable sensor can be used to control values of T and g on the acoustic signal *x(n)*.

Vibrato[15] is an algorithm which periodically varies the time delay at a low rate to produce a periodic pitch modulation. Typical values are 5 to 10 ms for average delay time, and 5 to 14 Hz for the low-frequency oscillator; parameters which two axes of acceleration from the *KiOm* control.

When a comb filter is combined with a modulating delayline, a flanger, chorus, slapback and echo effects are produced. If an FIR comb filter and a delay between 10 and 25 ms are used, a doubling effect known as slapback occurs. If an FIR filter with a delay greater then 50 ms is used, an echo is heard. If the delay time is continuously varied between 0 and 15 ms, an effect known as flanging occurs. If the delay line is varied sporadically between 10 and 25 ms, a chorus effect occurs [15]. The *KiOm* is used to control parameters to all these different algorithms.

4. CASE STUDIES

This section describes different experiments with a variety of instruments to show the versatility and evolvement of our system.

Figure 2 shows our first experiments with a drumset performance. The wearable prototype sensor was first placed 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 as seen on the left of Figure 3. 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, as shown on the right of Figure 3. Here it is attached to a headset (headphones with boom microphone) so that the performer can 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 [3], which initiated this research.

Figure 2. Wearable sensors used for a drumset performance

Figure 3. Wearable sensor used for a Tabla performance (left). Set up for using on head of performer (right).

Figure 4. Wearable sensor used to capture scratching gesture of turntablist (left). Wearable sensor used in conjunction with live Computer Music Performance using ChucK (right).

More experiments include performances with a turntablist who was scratching vinyl records with the *KiOM* placed on the hand (Figure 4 (left)), similar to the drum experiments. By the third 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 as shown on the right of Figure 4.

We have also experimented with musicians playing guitar, bass, saxophone and trumpet.

5. CONCLUSION

We have described the development of a framework for wearable sensor technology to control parameters of audio effects for real-time musical signal processing. Gesture data from a performing artist is captured using three-axis accelerometer packages that are converted to MIDI messages and control parameters of audio effects modifying the acoustic signal.

There are various directions for future work. We have planned to evolve to a wireless system, but are wary because problems described in [1]. The system needs to be full proof for performance on stage in which wires offer the desired reliability. We are interested in integrating other sensors like gyroscopes and magnetometers, to get position accurate data. We are interested in the effect of dancers wearing the sensors to control parameter of DSP algorithms and modifying the music. We are also interested in the possibility of our sensor package being used to sonify movements of patients with neurological motion disorders, in order to help diagnosis.

6. ACKNOWLEDGEMENTS

Thanks to Bernie Till for his discussions on wearable sensors. 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. Also thanks to the musicians who took part in our initial experiments.

7. REFERENCES

[1] P. Cook. "Principles for Designing Computer Music Controllers," *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, Seattle, 2001.

[2] T. Hahn and C. Bahn, "Pikapika—the Collaborative Composition of an Interactive Sonic Character." *Organised Sound: An International Journal of Music Technology*. vol. 7, no. 3., 2003.

[3] A. Kapur, A. Lazier, P. Davidson, R.S. Wilson, and P.R. Cook. "The Electronic Sitar Controller," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Hamamatsu, Japan, pp. 7-12, 2004.

[4] T. Machover, "Hyperinstruments: A Progress Report 1987-1991," *Media Lab Document*, MIT, 1992.

[5] D. Merrill, "Head-Tracking for Gestural and Continuous Control of Parameterized Audio Effects," *Proceedings of the Conference on New Instruments for Musical Expression (NIME)*, Montreal, Canada. May 2003.

[6] J. Paradiso. "Wearable Wireless Sensing for Interactive Media," *First International Workshop on Wearable & Implantable Body Sensor Networks*, London, April 2004.

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

[8] M. Puckette, "Pure Data." *In Proceedings of the International Computer Music Conference (ICMC).* Thessaloniki, Greece, pp. 269-272, 1997.

[9] J. Ryan and C. Salter, "TGarden: Wearable Instruments and Augmented Physicallity," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Montreal, Canada. pp. 87-90, May 2003.

[10] H. Sawada, N. Onoe, and S. Hashimoto, "Acceleration sensor as an input device for musical environment," *In Proceedings of the. International Computer Music Conference (ICMC)*, pp. 421-424, 1996.

[11] G. Wang and P. R. Cook. "ChucK: A Concurrent, On-the-fly Audio Programming Language," *In Proceedings of the International Computer Music Conference (ICMC)*, Singapore, September 2003.

[12] G. Wang and P. R. Cook. "On-the-fly Programming: Using Code as an Expressive Musical Instrument." *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Hamamatsu, Japan, June 2004.

[13] G. Wang, A. Misra, A. Kapur, and P.R. Cook. "Yeah, ChucK It! => Dynamic, Controllable Interface Mapping," *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Vancouver, Canada, May, 2005.

[14] T. Winkler, "Making Motion Musical: Gesture Mapping Strategies for Interactive Computer Music." *In Proceedings for International Computer Music Conference (ICMC)*. San Francisco, CA. 1995.

[15] U. Zolzer, *DAFX: Digital Audio Effects,* John Wiley & Sons, 2002.