# SoniMime: Movement Sonification for Real-Time Timbre Shaping

Jesse Fox
CCRMA
Dept. of Music, Stanford University
Stanford, CA 94305
jrobfox@ccrma.stanford.edu

Jennifer Carlile
CCRMA
Dept. of Music, Stanford University
Stanford, CA 94305
jcarlile@ccrma.stanford.edu

## ABSTRACT

This paper describes the design of SoniMime, a system for the sonification of hand movement for real-time timbre shaping. We explore the application of the tristimulus timbre model for the sonification of gestural data, working toward the goals of musical expressivity and physical responsiveness. SoniMime uses two 3-D accelerometers connected to an Atmel microprocessor which outputs OSC control messages. Data filtering, parameter mapping, and sound synthesis take place in Pd running on a Linux computer.

## Keywords

Sonification, Musical Controller, Human Computer Interaction

## 1. INTRODUCTION

SoniMime is a new system for the sonification of human hand movement. SoniMime uses a pair of 3-D accelerometers to track hand gestures, while an Atmel microprocessor formats the serial accelerometer output into Open Sound Control (OSC) messages. All data filtering and sound synthesis is implemented using Pure Data (Pd) running on a Linux computer.

While SoniMime was originally conceived as a new method to capture gestural data from dancers in real-time, its applications can be far more diverse. Due to the precision and flexibility of the sensors used, we can deduce several different types of gestural movement, and thus several different schemes for the sonification of gesture data are explored.

In general, it is found that direct mappings of tilt and jerk to various synthesis parameters are more successful than memory-based, comparative pattern-matching schemes. Perhaps the most exciting implementation of SoniMime involves a gesturally-controlled tristimulus timbre synthesizer [5] [6], resulting in the generation of speech-like vowel formants.

## 2. SYSTEM

SoniMime uses a pair of ADXL ADC digital accelerometer boards produced by Procyon Engineering [1]. Each board houses a pair of bidirectional accelerometers that, when mounted perpendicularly, sense acceleration in 3 dimensions. The change in voltage associated with acceleration in any direction is sent through an A/D converter that outputs a single serial data stream based on the I2C data protocol. This data stream is interpreted by an Atmel ATMega16 microprocessor that outputs OSC messages via serial port. A computer running Pure Data (PD) under Linux houses all of our data filtering, mapping and sound synthesis software.

**Figure 1: Accelerometer attached to hand**

The physical design of the system is meant to be small and unobtrusive, attributes important to a dancer or other potential user. Each accelerometer board is attached to the hand via an elastic band [see fig. 1] and a small foam pouch, to ensure user comfort. Wires are run up the sleeves to an AVRmini development board [1] housed in a belt-pouch, and powered using a 9V battery. The AVRmini board houses our microprocessor, and a serial cable directly connects the board to the Linux computer.

All data is sent to PD as a single 30-bit stream of OSC messages, running at 115200 bits per second. Within PD, each 30-bit number is split into its 3x10-bit directional components (x, y, and z). DC offset is removed, and the data are filtered in three ways: First, a series of high-pass filters output accelerometer data corresponding to sudden jerks or impacts. Second, a series of low-pass filters output accelerometer data corresponding to tilt. Third, tilt data is differentiated (using a one-sample memory buffer) to output jerk, or change in acceleration. Our final OSC data stream then consists of three separate outputs for each direction, or nine outputs total for each hand. Finally, a "stillness detector" is implemented that changes state based upon continual analysis of tilt data from each hand. The end result is a Pd abstraction that has a total of twenty outputs (9 for each hand, plus one "stillness detector" for each hand).

## 3. SONIFICATION AND CONTROL

SoniMime translates acceleration data with high resolution and low latency into a data stream that reflects three specific kinds of hand movement–tilt, jerk, and impact. This feature allows us to easily adapt SoniMime for use in a wide array of applications.

### 3.1 Tristimulus Timbre Model

An interesting application of SoniMime is a physical implementation of the tristimulus timbre model for sound shaping. In Pd, we created a patch that maps sensor data to control frequency, amplitude, and timbre of a synthesized sound.

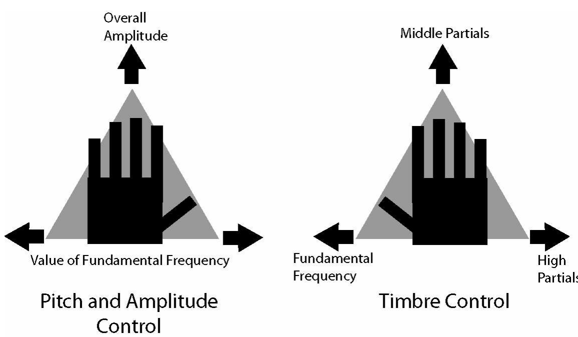


**Figure 2: Parameter Mapping of X and Z tilt**

X and Z tilt data from the "timbre" hand (which could be either the left or right hand) determine the weight given to each of the three timbre stimuli: the fundamental frequency, the middle partials (2-5), and the high partials (6-16). The sum of the weight given to all three stimuli is always equal to 1. If the hand is tilted all the way to the left along the X axis, the fundamental frequency is given a weighting of 1; if the hand is tilted all the way to the right, the high partials are given a weighting of 1; if the hand is tilted all the way forward along the Z axis, the middle partials receive all of the weighting. Everywhere in between, the X and Z tilt data is combined, and the timbre is determined by a weighted sum of all three stimuli [see fig.2].

X-tilt data from the "frequency" hand determines the value of the fundamental frequency, ranging from approximately 50Hz to 615Hz. The Z tilt controls the overall amplitude of the sound.

An emergent property of our application of the tristimulus timbre model is that it sounds like a variable formant synthesizer. Through careful control of the "timbre" hand, one can create a wide variety of vowel-like sounds, for which the human auditory system is innately tuned. In practice, a performer could learn to control the SoniMime system to create melodies with distinct vocal qualities.

The software patch for the tristimulus timbre model (along with the complete SoniMime software package) is available online [2].

## 4. RELATED WORK

Fels and Hinton in Glove-TalkII [4] explore speech synthesis using specific learned hand gestures in an adaptive interface. Instead of a static gesture-to-consonant mapping, SoniMime utilizes a truly continuous sensing environment that depends upon the user's fine motor control and ability to learn to manipulate a virtual timbre space, dictated by our tristimulus timbre model implementation.

## 5. FUTURE WORK

At the time of this writing, SoniMime is limited in its appeal to dancers based upon its reliance on a hard-wired serial connection with a computer. Future work will focus upon implementing wireless communication between the Atmel microprocessor and a host computer.

The sonification of movement can be a powerful learning tool for physical activities. Sonification introduces auditory feedback in situations in which audible stimuli would otherwise be absent. Other future projects include expanding the spectrum of useful applications for SoniMime to include the sonification of specific learned motions, perhaps as diverse as choreographed dance sequences and even a golf swing.

## 6. ACKNOWLEDGMENTS

SoniMime is the result of a project developed during the Autumn quarter in 2004 at CCRMA as part of a course taught by Bill Verplank called "Human Computer Interaction Theory and Practice" [3]. Many thanks to Bill, Jonathan Berger, and Max Mathews. The authors would also like to thank Matt Wright for the use of his accelerometer data filtering scheme, and Wendy Ju for her invaluable advice during the development process.

## 7. REFERENCES

[1] Procyon Engineering
`http://www.procyonengineering.com`.

[2] SoniMime web site, with software examples:
`http://ccrma.stanford.edu/~jcarlile/250a/sonimime.html`.

[3] Course web site for Music 250a, Stanford University
`http://ccrma.stanford.edu/courses/250a/`.

[4] S. Fels and G. Hinton. Glove talk ii: A neural network interface which maps gestures to parallel formant speech synthesizer controls. *IEEE Transactions on Neural Networks*, 9(1):205 – 212, 1998.

[5] H. Pollard and E. Jansson. A tristimulus method for the specification of musical timbre. *Acustica*, 51:162 – 171, 1982.

[6] A. Riley and D. Howard. Real-time tristimulus timbre synthesizer. 2004.
`http://www-users.york.ac.uk/~dmh8/tristimulus.htm`.