DESIGNING AN EXPRESSIVE VIRTUAL PERCUSSION INSTRUMENT

Brian Dolhansky
Drexel University
bdol@drexel.edu

Andrew McPherson
Drexel University
apm@drexel.edu

Youngmoo E. Kim
Drexel University
ykim@drexel.edu

ABSTRACT

One advantage of modern smart phones is their ability to run complex applications such as instrument simulators. Most available percussion applications use a trigger-type implementation to detect when a user has made a gesture corresponding to a drum hit, which limits the expressiveness of the instrument. This paper presents an alternative method for detecting drum gestures and producing a latency-reduced output sound. Multiple features related to the shape of the percussive stroke are also extracted. These features are used in a variety of physically-inspired and novel sound mappings. The combination of these components provides an expressive percussion experience for the user.

1. INTRODUCTION

With the widespread use of mobile devices by the general public, musical applications and games have been particularly popular. On the Apple App Store there are several percussion applications that attempt to give the user the ability to play a full or partial drum set on their phone. However, the degree of expressivity granted by these percussion applications is limited by simple trigger-type input methods. Playing a stored drum sample via a trigger does not allow the user to modify the qualitative aspects of a single drum hit.

While, to some, percussion instruments appear to lack expression, in reality a drummer is able to change their playing style by not only modifying their drumming rhythm, but also by adjusting parameters such as the energy they impart to each drum hit, the stick type used and the location of the hit on the drum.

Unlike the aforementioned drum applications, we present an implementation that replicates some of these expressive qualities by extracting features from an accelerometer profile generated when a user moves a mobile device like a physical drum stick. Several percussive gesture features regarding the timing and shape of the preparation and stroke are extracted from the accelerometer signal and mapped to an output sound. A predictive triggering method substantially reduces the latency between gesture and output sound. This system provides a musical experience that better replicates playing a physical instrument.

Figure 1: The virtual percussion system presented in this paper aims to provide a system that closely replicates an actual instrument. It was inspired by analyzing actual drum strokes.

2. PREVIOUS WORK

Several studies have focused on using accelerometers and gesture recognition in an attempt to replicate the motions used when playing real physical instruments. Dahl [1] described the correlation between stroke shape and strike velocity through user studies. Hajian et al. [2] studied human drumming styles by recording the accelerometer profiles of actual drum strikes. Tindale et al. [3] assessed various stick augmentation techniques, including using accelerometers and gyroscopes to aid in drum gesture detection. The AoBachi system by Young et al. [4] replicated Japanese taiko drumming by using accelerometers placed in the large "bachi" drum sticks. Bott et al. [5] used a standard Wii Remote to simulate playing a number of instruments, including drums, by making gestures corresponding to the typical motions involved in playing those instruments.

Cook explored expressive synthesis of percussion sounds with the Physically Informed Stochastic Event Modeling (PhISEM) algorithm [6] and detailed its use in software- [7] and hardware-based [8] applications. Accelerometers were attached to simple controllers, such as a user's foot for a toe-tapping instrument, and players were able to shape pre-recorded drum loops. Cook argued for simple instruments that are intuitive for an inexperienced user to play.

Heise et al. [9] used a Wiimote to integrate the PhISEM model and a physical controller. This system was able to simulate various percussion instruments, including the maraca and rainstick.

There has also been research conducted on the viability of replicating drum playing on a mobile device. The Shoogle system proposed by Williamson et al. [10] used haptic feedback to notify the user of certain events. This system could be extended to provide a realistic feeling when playing a virtual drum instrument. Tanaka [11] presented a collaborative musical creation system that used mobile devices augmented with a suite of sensors so that a group of users could compose a piece of music. The ShaMus system, developed by Essl et al. [12], used phone tilt information calculated by an on-board accelerometer to control a virtual drum. The user made a strike gesture by tilting the device. Some expressive control was afforded by measuring the rate at which the device was tilted past the horizontal plane. Weinberg et al. [13] implemented the music creation tool ZooZBeat, which used gestural input as a composition tool. Accelerometer onset detection was used for note input. The energy of the onset was calculated to allow the user to change the pitch of the input note.

Our system extends the work presented in these studies by capturing percussive motions on a mobile device and mapping these gestures to sound synthesis. This system uses an intuitive input and output method, as the user simply has to move the mobile device like a drum stick. We propose the extraction of multiple independent features per swing. The output sound is produced exactly when the user expects it, and with modifications that are directly related to the extracted features.

3. OVERVIEW

To better understand the acceleration characteristics of a percussion stroke, we first recorded the movements of musicians playing real instruments using the accelerometer in an iPod Touch. For comparison, we carried out perceptual tests where users swung only the mobile device in a motion mimicking a percussive stroke. We identified several key features that affect the quality of the sound of a drum hit and designed a system to extract these features in real time on a mobile device.

The implementation of an expressive virtual percussion instrument requires several integrated components that must operate quickly enough to provide an experience that mimics playing a real instrument. The system must be simple to reduce computation time, but it must also provide enough expressive control to make playing the instrument interesting for more than several minutes.

The proposed system has three subsystems: hit prediction, feature extraction and feature mapping. Each time the device receives a new accelerometer sample, it checks to see if a hit is imminent. If so, the device examines the past motion information to determine what type of hit will occur and extract other related gestural features. The device then maps these features to an output sound.

4. DEFINING INSTRUMENT EXPRESSIVENESS

4.1 Expression of Physical Instruments

The term "expression" is difficult to define. Any musical piece can be played in different manners by changing not only what notes are played, but *how* the notes are played. An instrument's ability to qualitatively modify notes or sounds constitutes its expressiveness.

Generally, the degree of expression afforded by an instrument depends on the physical properties of that instrument. For example, it is possible for a violin player to continuously manipulate the pitch and timbre of the notes they play, allowing for a wide variety of techniques, such as slides or vibrato.

Although most drums lack melodic expression, they are not entirely "one-dimensional" instruments. A percussion instrument (or set) has several musical degrees of freedom: *what* drum a drummer plays, *when* they play it, and *how* they play it. A drummer can modify how they play a particular percussion instrument by using different sticks or hitting different parts of the drum. Changing the stick type or hit location affects the timbre of the sound.

Implementing these qualitative aspects is especially important when designing a virtual percussion instrument, as it would give the musician using the virtual instrument nearly the same amount of control and expression as if they were playing a real instrument. In addition, it is important to intuitively provide access to these qualities so that the user immediately understands the effects their gestures have on the produced sound.

4.2 Replicating Drum Expression on a Mobile Device

In order to replicate these degrees of freedom, a mobile device can record a user's gestures using an accelerometer. The most intuitive motion for a user is to swing the device as though they are using an actual drumstick. In order to avoid breaking immersion, the system must not only detect each stroke, but the output sound must be produced concurrently with its apparent impact. Latency must therefore be addressed to provide a consistent experience.

The accelerometer magnitude profile of a drum hit is similar even among various instruments (see Section 5). Therefore, it is possible to extract certain features from the percussive gesture to modify the output sound accordingly. For instance, if the user swings the device quickly, a louder sound should be produced, mimicking the behavior of a physical drum. Other non-physical mappings were explored, such as triggering different instrument types based on the shape of the stroke (a discrete mapping) or affecting the pitch based on the stroke time (a continuous mapping).

5. REAL PERCUSSIVE ACCELEROMETER PROFILES

A pre-study was conducted to record the accelerometer profiles of the movements required to play different percussion instruments. Accelerometer data of over 300 percussive motions spanning 8 different instruments was recorded.

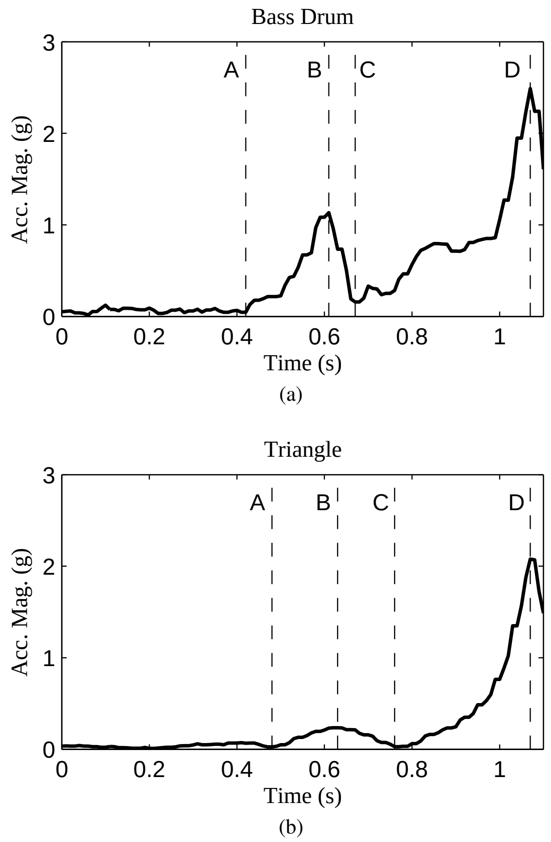

Figure 2: Accelerometer profiles for two percussion instruments (each of which requires a different playing style) played at the same velocity. The regions of the stroke are labelled, with point A corresponding to the start of the back swing, point B being the point of maximum acceleration of the back swing, point C being the midpoint of the back stroke, and D occurring at the point of impact.

The instruments included the bass drum, glockenspiel, slapstick, snare drum, tam-tam, toms, triangle and wood block. Three different users held an accelerometer-equipped mobile device (protected by a foam covering) and a drumstick in their dominant playing hand (Figure 1). This setup was used because playback occurs an iPod device, and it is important to analyze accelerometer data from the same perspective in both recording and playback (from the base of the user's hand, rather than the tip of the drumstick.)

The users were asked to hit each instrument 5 times at 3 different velocities. Some extra profiles were recorded, including quick successive hits, rolls and swells. All subjects had musical experience, although only one was an expert percussionist.

A generalized drum stroke was observed. A user first made a back swing away from the drum to prepare for the hit. The user then began accelerating the device in the opposite direction to strike the drum. The length of these motions depended on the instrument and the playing style.

Figure 2 shows the accelerometer magnitude profiles of two instruments that have different playing styles, specifically the bass drum and triangle. The magnitude is shown for one second preceding the drum strike. Note the distinguishable points in both examples:

- Point A: the point in time when the user begins the percussive back swing
- Point B: the peak acceleration of the first half of the back swing
- Point C: the midpoint of the back stroke, or where the user began accelerating the device towards the drum
- Point D: the point where the drum stick impacts the drum surface

Some of the feature mappings examined later were inspired by these physical trials. For instance, the bass drum generally has a longer wind up when compared to other percussion instruments. The mallet used to strike a bass drum is significantly larger than typical wooden drum sticks and the percussionist requires a longer time and distance to accelerate the mallet to the appropriate velocity. Smaller instruments such as the triangle use a smaller stick that requires less energy to move and therefore require a smaller wind up. In addition, the slope of the forward swing used to strike these smaller instruments tends to be steeper, as hitting a triangle involves a whipping motion as opposed to the grandiose swing required for a concert bass drum.

6. LOW-LATENCY HIT DETECTION

Percussion instruments are often used to keep rigid time. It is especially important to minimize or eliminate the lag between the detection of a hit and sound production, as any perceptible amount of latency diminishes both the rhythmic accuracy and the playing experience. We therefore developed a system to accurately predict a hit before its actual point of impact.

6.1 Platform Limitations

The accelerometer installed on the 4th generation iPod touch used for this implementation is software limited to 100 Hz in order to conserve battery life. This is non-ideal, especially for high frequency time-sensitive applications. In addition, the accelerometer data is very noisy, as it is mainly meant to ascertain the orientation of the device as opposed to measuring movement or displacement.

The latency inherent in the platform is partially due to the time it takes to sample the accelerometer. A larger amount is due to the architecture of iOS's sound API, Core Audio. Empirically, the latency between when a hit was detected and when the sound was produced ranged from 10 to 20 milliseconds.

6.2 Determining the Characteristics of a Virtual Hit

It was expected that the acceleration profile of a user swinging only the mobile device would differ from when the user swung both the mobile device and a large drum mallet. It was important to not only determine the physical characteristics of a hit, as in Section 5, but the perceptual, user-defined qualities of a virtual hit. For instance, users may think that a sound should be triggered slightly before or after the actual peak in acceleration magnitude.

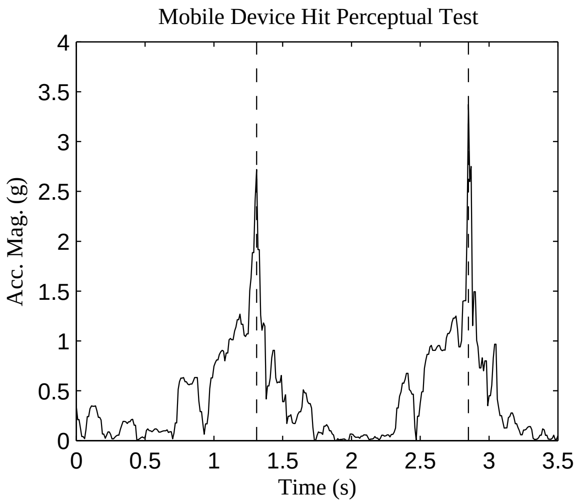

Figure 3: The locations of user triggered hits versus the acceleration profile of a percussive motion made with the mobile device.

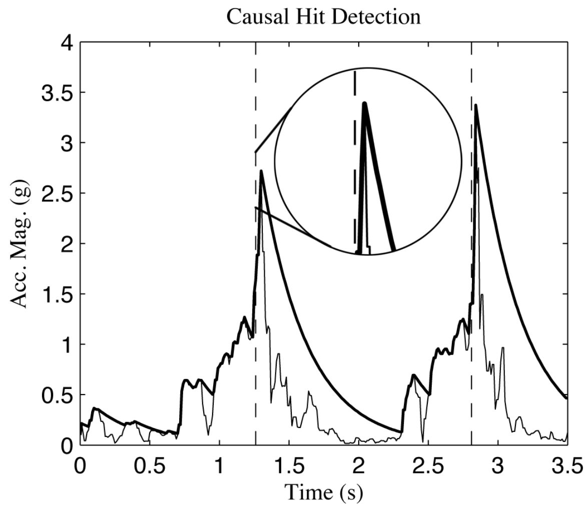

Figure 4: A demonstration of the hit prediction system. The thick line is the hit detection threshold, while the vertical dashed lines indicate where a hit prediction was made.

A program was developed to record accelerometer and screen touch data. Users were asked to touch the screen when they thought a hit should occur. As can be seen in Figure 3, users correlated a peak in acceleration magnitude with when the percussion sound should be produced. The act of reversing the device direction, which mimics striking a drum, corresponds to a peak in acceleration magnitude. Accordingly, users associate peaks in acceleration with an expected sound.

6.3 Difficulties of Real-Time Hit Detection

Because sound should be produced when there is a peak in acceleration magnitude, the system should possess a robust peak-picking algorithm. However, the proposed system is causal and cannot implement a traditional peak picking algorithm that attempts to find maxima using information on both sides of the peak. In addition, even if latency were ignored, it is possible that a detected peak was a local maximum that occurred slightly before the larger hit acceleration peak.

6.4 Hit Prediction via Onset Detection

Our solution for causality and latency is a variation of onset detection that performs hit *prediction*. This solution was developed based on the characteristics of the accelerometer profiles produced by the virtual drum user tests. Any large increase in acceleration magnitude means that a hit is imminent. A person can only swing the device at a high rate for a limited amount of time before the extent of possible motion is reached.

The system should also detect hits of variable magnitude. A dynamic threshold is implemented with an envelope follower so that both hard and soft hits are recognized.

The threshold follows the leading edge of large onsets and decays from peaks with a per-sample decay rate of 3%. The decay accounts for the accelerometer peaks produced by the rebounding motion of the user after they make a hit gesture. A noise floor is also utilized in order to ignore small fluctuations produced by the accelerometer hardware and accidental user input. This floor is set at a magnitude of 0.1 g. iOS's Core Motion API filters out the acceleration due to gravity automatically, so the gravity vector can be ignored.

If, in the current time step, the accelerometer magnitude exceeds the envelope's previous value at a high enough rate, the system determines that a hit is imminent. The slope needed for hit detection at the threshold crossing can be varied to create different sensitivities. A small rate, such as 0.25 g per sample, produces a very sensitive onset detector that can be triggered with very small movements. Increasing this rate to 0.4 g per sample requires large movements. This rate can therefore be tweaked to provide a balance between sensitivity and noise robustness.

Figure 4 shows the data recorded from user tests along with the causal hit prediction. Note that the onset detections occur several samples before the actual peak, giving our system extra computation and sound synthesis time. The mean prediction time afforded by our system was 2.09 samples (around 20 ms), i.e. our system predicted a hit 20 ms before a user expected the hit to occur. This is greater than the inherent system latency.

7. FEATURE EXTRACTION AND SOUND MAPPING

After a hit has been detected, features describing the hit must be extracted. These percussive stroke features can be extracted deterministically from the buffered accelerometer samples. Features of the future hit peak must also be estimated. Finally, these extracted features must be mapped to the output sound.

7.1 Causal Feature Extraction

When an imminent hit is detected, the system has access to past accelerometer samples. Several features describing the stroke can be extracted from these samples. Specifically, the following features are calculated:

- Length of entire stroke
- Length of back and forward swings
- Ratio of times spent in back and forward swings
- Velocity estimate (integral of acceleration over each segment) of back and forward swings

Each of these features must be calculated automatically in a short amount of time. Our system uses the accelerometer profile generalizations observed in Section 5. We first determine points A, B and C, shown in Figure 2, and then segment the percussive stroke for feature extraction.

First, the location of the largest maximum that occurred during the first half of the back (denoted B in Figure 2) swing is calculated by using a general noise-robust peak picking algorithm.

Next, the minimum sample between point B and the current accelerometer sample is determined. This is point C and is where the user begins accelerating the device towards the drum, or where the application of force to the device changed direction. Finally, the start of the stroke (point A) is calculated by finding the first accelerometer magnitude sample that is less than 10% of the value at point B.

The length of the entire stroke, in samples, is the difference between point A and the current accelerometer sample. Similarly, the length of the back swing is related to the difference between points point A and point C. The velocity estimate of the back swing is calculated by summing the magnitudes over the segment between A and C. Because the accelerometer is noisy, calculating the velocity directly is prone to drift. In addition, there is no information about the initial velocity from the accelerometer magnitude.

7.2 Non-Causal Feature Extrapolation

The characteristics of a drum hit that determine the volume of the produced sound are mainly the velocity of the hit, the type of stick or mallet used, and the drum type. The type of stick or drum used is a programmatic choice made when designing a specific implementation of our system. However, it is important to accurately emulate the correlation between the velocity of the drum stick and the energy of the output sound, as this is one of the key techniques a drummer uses when playing a piece expressively.

Due to our use of hit prediction, we do not have immediate access to the actual peak accelerometer magnitude value. Therefore, we must use another predictive step to estimate the peak acceleration. Two causal features were used in this prediction and were motivated by the accelerometer pre-study. First, a large slope in the forward swing region (the samples immediately preceding D) corresponded to a sharper play style and louder output sounds. In addition, a large amount of forward swing acceleration corresponded to a very high strike velocity, specifically for mallet strikes on the bass drum. The features used for prediction were:

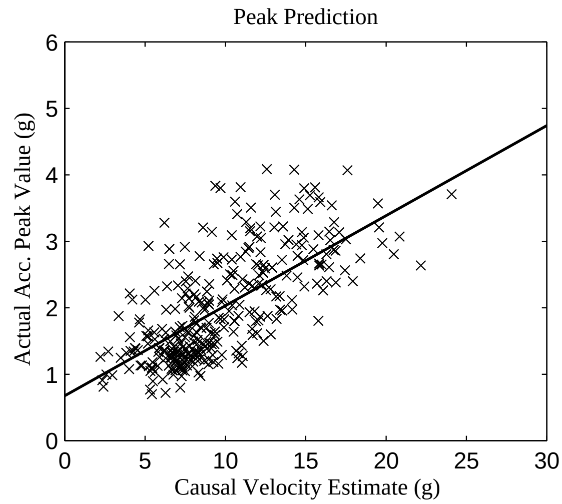

Figure 5: Visual representation of the correlation between the velocity estimate (Equation 1) and the peak acceleration.

- Slope (derivative) estimate of forward swing
- Forward swing velocity (integral) estimate

The slope estimate was determined by finding the slope between the accelerometer sample at the time of hit detection and the accelerometer value k samples prior. Different values of k were tested and $k = 10$ (or 0.1s) provided the best peak estimate. The forward swing velocity estimate was calculated via Equation 1, where $m[n]$ is the magnitude of the acceleration at time n, and i is the current sample.

$$\hat{v} = \sum_{0}^{k} m[i-k] \tag{1}$$

In order to prove the validity of applying these empirical observations, we calculated the above features for 331 specific hits and calculated the correlation between each feature and the actual accelerometer peak value Figure 5. A strong correlation between the features and the actual peak value was found, with $r = 0.74$ for the slope estimate feature and $r = 0.80$ for the velocity estimate feature.

7.3 Feature Mapping

Once the percussive stroke and peak estimate features have been calculated, they must be mapped to the output sound. Several mappings were explored, including those that used some of the physical characteristics of a drum as inspiration, and those that were based on novel (non-physical) mappings.

7.3.1 Physically-Inspired Mappings

Even for those with little or no percussion experience, it makes intuitive sense that hitting a drum harder should produce a louder noise. It is obvious that the mapping between the velocity estimate feature and the output sound should be proportional. However, the amount of energy imparted to a surface by a moving object is not directly proportional to the moving object's velocity. A moving drum stick possesses a certain amount of kinetic energy, given by $E_k = \frac{1}{2}mv^2$. Although kinetic energy is not a direct measure of loudness, mapping the squared velocity estimate to the output sound's volume provided a natural continuum of sound.

Different drums require different playing styles. As Figure 2 shows, the concert bass drum requires the use of a large mallet and large strokes, while the triangle uses a small stick and small, quick movements. These observations can be used to select different instrument types based on the type of stroke used. A mapping was explored that switched between a triangle and bass drum output sound based on the length of the entire stroke. The type of instrument selected was independent of the final output sound volume, so a user was able to switch between instruments at will and play each with variable loudness.

7.3.2 Non-physical Mappings

Several other unique mappings were created that specifically utilized the stroke features. For this initial work, we felt it was important to use simple mappings so that it was possible for the user to discover the relationship between their gestures and the effects on the output sound. One mapping used a pitched drum output sound and changed the pitch based on the length of the stroke, with longer strokes producing lower pitches. This mapping used the strokes required to play large instruments as inspiration. Large melodic drums produce lower pitches and require large impulses.

Another mapping adjusted the frequency of a low pass filter based on the ratio between the segments A-C and C-D. This allowed a user to produce bright sounds with relatively quick forward swings, and muffled sounds with relatively slower forward swings.

These implementations are simple examples. It is possible for a developer to create many other unique percussive mappings using this system.

8. FUTURE WORK

We plan to incorporate gyroscope data for noise reduction and for access to 3 dimensional position features. These position features would allow a user to play different drum types by making strike gestures in different locations.

Other avenues for expressive control will be explored. For instance, physical modelling, specifically the percussive convolution synthesis detailed by Aimi [14], will be examined. Because the general acceleration magnitude envelope is already available, it is possible to match this envelope to prerecorded audio-rate drum impulses and convolve these with the impulse responses of different drums. This will provide a more realistic and expressive output sound because it is based on the real response of a drum.

9. REFERENCES

[1] S. Dahl, "On the beat: Human movement and timing in the production and perception of music," Ph.D. dissertation, KTH Royal Institute of Technology, 2005.

[2] A. Z. Hajian, D. S. Sanchez, and R. D. Howe, "Drum roll: Increasing bandwidth through passive impedance modulation," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 1997.

[3] A. R. Tindale, A. Kapur, G. Tzanetakis, P. Driessen, and A. Schloss, "A comparison of sensor strategies for captruing percussive gestures," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '05)*, 2005.

[4] D. Young and I. Fujinaga, "Aobachi: A new interface for Japanese drumming," *Proceedings of the Conference on New Interfaces for Musical Expression (NIME '04)*, pp. 23–26, 2004.

[5] J. N. Bott, J. G. Crowley, and J. J. LaViola, Jr., "Exploring 3D gestural interfaces for music creation in video games," *Proceedings of the 4th International Conference on Foundations of Digital Games*, 2009.

[6] P. R. Cook, "Physically informed sonic modeling (PhISM): Synthesis of percussive sounds," *Computer Music Journal*, vol. 21, no. 3, pp. 38–49, 1997.

[7] ——, "Toward physically-informed parametric synthesis of sound effects," *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999.

[8] ——, "Principles for designing computer music controllers," *Proceedings of the Conference on New Interfaces for Musical Expression (NIME '01)*, 2001.

[9] J. L. Sebastian Heise, "A versatile expressive percussion instrument with game technology," *Multimedia and Expo, 2008 IEEE International Conference on MultiMedia and Expo (ICME '08)*, pp. 393–396, 2008.

[10] J. Williamson, R. Murray-Smith, and S. Hughes, "Shoogle: Excitatory multimodal interaction on mobile devices," *Proc. of the Conference on Human Factors in Computing Systems (CHI '07)*, 2007.

[11] A. Tanaka, "Mobile music making," *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME '04)*, 2004.

[12] G. Essl and M. Rohs, "ShaMus - a sensor-based integrated mobile phone instrument," *Proceedings of the International Computer Music Conference (ICMC '07)*, 2007.

[13] G. Weinberg, "ZooZBeat: a gesture-based mobile music studio," *Proceedings of the 2009 Conference on New Interfaces for Musical Expression (NIME '09)*, 2009.

[14] R. M. Aimi, "Hybrid percussion: Extending physical instruments using sampled acoustics," Ph.D. dissertation, Massachusetts Institute of Technology, February 2007.