Stompboxes: Kicking the Habit

Gregory Burlet
Distributed Digital Music Archives and Libraries Lab
CIRMMT, McGill University
Montréal, QC, Canada
gregory.burlet@mail.mcgill.ca

Marcelo M. Wanderley
Input Devices and Music Interaction Lab
CIRMMT, McGill University
Montréal, QC, Canada
marcelo.wanderley@mcgill.ca

Ichiro Fujinaga
Distributed Digital Music Archives and Libraries Lab
CIRMMT, McGill University
Montréal, QC, Canada
ich@music.mcgill.ca

ABSTRACT

Sensor-based gesture recognition is investigated as a possible solution to the problem of managing an overwhelming number of audio effects in live guitar performances. A realtime gesture recognition system, which automatically toggles digital audio effects according to gestural information captured by an accelerometer attached to the body of a guitar, is presented. To supplement the several predefined gestures provided by the recognition system, personalized gestures may be trained by the user. Upon successful recognition of a gesture, the corresponding audio effects are applied to the guitar signal and visual feedback is provided to the user. An evaluation of the system yielded 86% accuracy for user-independent recognition and 99% accuracy for user-dependent recognition, on average.

Keywords

Augmented instrument, gesture recognition, accelerometer, pattern recognition, performance practice

1. INTRODUCTION

Audio effects in the form of stompboxes—small pedals, typically housing a single audio effect, which are designed to be toggled by a musician's foot—have become prominent in the sonic arsenals of electric guitarists. Stompboxes are designed to be "daisy-chained" together, such that multiple audio effects may be applied to an input signal in series.

Many musical works employ a battery of different audio effects, which a guitarist must toggle at the correct times. Consequently, effect-driven music poses a new set of challenges to guitarists in the context of live performances. Managing an array of audio effects creates unnecessary or even physically unmanageable overhead [6] that detracts from other important aspects of live performance, such as audience interaction. For example, when a piece being performed dictates a change in audio effects, the guitarist must return to his or her array of stompboxes and concentrate on activating the correct combination of pedals. Although multi-effect pedals aim to alleviate this issue by allowing multiple audio effects to be activated by depressing a single pedal, one must still physically depress the pedal corresponding to the desired combination of effects to activate. Is it that guitarists have become complacent with the amount of overhead required to manage an array of audio effects, or is it that a better solution does not exist?

Sensor-based gesture recognition is investigated as a potential solution to this problem. Using direct gesture acquisition [16], an accelerometer is attached to the body of an electric guitar to measure the three-dimensional acceleration of gestures performed by a guitarist. Pattern recognition techniques are used to trigger the application of digital audio effects to the guitar signal in response to the gestural information captured by the accelerometer. The robustness of gestural control and methods for restoring the tactile and visual feedback of audio effect activation, which is lost when using gesture recognition in lieu of depressing a series of stompboxes, are investigated.

In the remaining sections of this paper, a review of relevant research is presented. An overview of the developed gesture recognition system is provided. The implemented pattern recognition algorithm is described in detail, followed by a discussion of the importance of recognition feedback. Finally, two experiments are performed to evaluate the implemented gesture recognition system.

2. PREVIOUS WORK

Differentiating between the type of input signals processed by gesture recognition systems, Mäntyjärvi et al. [8] define "discrete gesture commands" to be gestures with a user-defined start and stop time, while "continuous gesture commands" involve a continuous series of gestures with no explicit segmentation. This paper will focus on the complex task of continuous gesture recognition.

The first step in the automatic recognition of continuous gesture commands is temporal segmentation [9]. Specifying the start and stop times of a gesture may be manually performed by pressing a button on the gesture capturing device while performing a gesture [11]. Techniques for automatic gesture segmentation include locating abrupt changes in signal magnitude [5] or using a sliding-window representation of the input signal [14].

The second step in the gesture recognition problem is gesture classification. For this task, left-to-right hidden Markov models (HMMs) are predominantly used in the literature [9]. The underlying Markov chain represents the transitions between states of a gesture, where the left-to-right orientation enforces forward movement through the gesture. Using HMMs, Bevilacqua et al. [2] developed a gesture following system, which continuously outputs the likelihood of each trained gesture as the input signal is analyzed. This system was later extended to use hierarchical HMMs [4].

Dynamic time warping (DTW) algorithms have also been applied to the gesture recognition problem to measure the congruency of a performed gesture to a set of reference gestures which may vary in speed of execution. To process con-

NIME'13, May 27 – 30, 2013, KAIST, Daejeon, Korea.

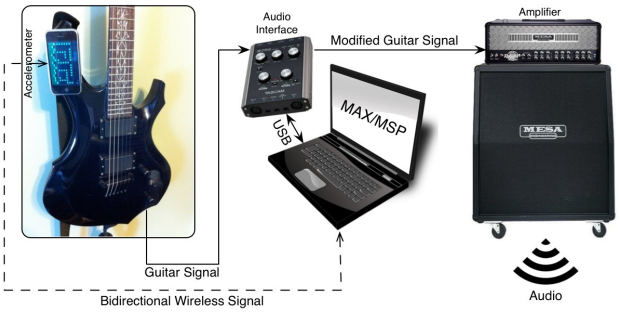

Figure 1: Hardware configuration and data flow of the gesture recognition system. Solid lines represent wired connections. Dashed lines represent wireless connections.

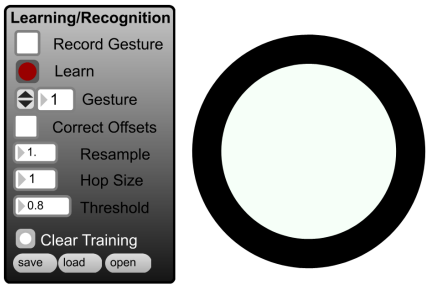

Figure 2: Graphical user interface (GUI) for the gesture recognition system (left). A GUI widget provides visual feedback to the guitarist upon successful gesture recognition (right).

tinuous gesture commands in realtime, Bettens and Todoroff [1] proposed a "multi-grid" DTW algorithm, which uses multiple shifted DTW grids that each posit a starting point of the performed gesture.

Another technique used for continuous gesture recognition is template matching. Thiebaut et al. [14] proposed the use of two metrics to compare the accelerometer samples of a reference gesture and a performed gesture: Euclidean distance and cosine similarity. A performed gesture is recognized when the distance or similarity to a reference gesture crosses a predefined threshold parameter.

3. SYSTEM OVERVIEW

A system has been developed to recognize, in realtime, the continuous gesture commands of a guitarist; it is not required for the musician to inform the system of the start and stop times of performed gestures. Moreover, a collection of digital audio effects have been developed in the Max/MSP visual programming language [10], which are activated in response to performed gestures and applied to the input guitar signal. The described system requires the configuration of both hardware and software, outlined in this section.

3.1 Hardware

The hardware configuration and data flow of the prototype gesture recognition system is illustrated in Figure 1. A Tascam US-144MKII audio interface digitizes the input guitar signal. An iPhone 4 is attached to the body of an electric guitar and the TouchOSC iPhone application[1] is used to access the acceleration signal of the device in the x, y, and z planes at a sampling rate of approximately 32Hz. Although the iPhone 4 also has a gyroscope sensor, which additionally measures the yaw, pitch, and roll of the device, only the digitized accelerometer signal is accessible through the TouchOSC application. Any choice of audio interface and accelerometer, provided it captures a three-dimensional acceleration signal at a reasonable sampling rate, is sufficient.

3.2 Software

Implemented in Max/MSP is a collection of digital audio effects which extend the Max 5 Guitar Processor Tutorial.[2] The following audio effects have been implemented: equalizer, compressor, distortion, whammy, phasor, reverb, granular synthesis, looping delay, and modulating digital delay. The aforementioned audio effects are connected in series to mimic the functionality of an array of physical stompboxes. Optionally, the parameters of each audio effect may be continuously modified by the x, y, or z component of the incoming iPhone accelerometer signal.

Effect presets may be created to store and recall the parameters of multiple audio effects. Effect presets may be manually activated from the iPhone TouchOSC interface, or activated by the gesture recognition system. Using the gesture recognition system, a Max/MSP interface allows guitarists to enter the sequence of effect presets required to perform a musical work. When a trained gesture is recognized, the effect preset required to perform the next section of the piece is automatically activated and the appropriate audio effects are applied to the input guitar signal.

The gesture recognition software is implemented as a Max/MSP patch, which uses the FTM library of complex data structures for Max [12] and the Gabor library of Max/MSP externals for digital signal processing functions [13]. A custom interface has been designed for TouchOSC, which enables accelerometer and interface data to be wirelessly transmitted between the iPhone and the Max/MSP patch over a UDP connection using the Open Sound Control (OSC) protocol [17].

4. GESTURE RECOGNITION

The gesture recognition system is controlled through an interface (see Figure 2), which allows the guitarist to train gestures, modify parameters of the gesture recognition algorithm, and alternate between training and recognition mode. Visual feedback is provided to the guitarist when a gesture is successfully recognized. This section will describe the signal preprocessing, training, recognition, and feedback phases of the gesture recognition system.

4.1 Signal Preprocessing

The three-dimensional accelerometer signal

$$A[n] = (x[n], y[n], z[n]) \quad (1)$$

is optionally preprocessed by an offset correction or signal resampling function before being stored as a training instance or used for recognition. Offset correction considers the values of the x, y, and z components of the accelerometer signal and removes the offset from zero to create a new origin for future samples. The signal may also be upsampled or downsampled to account for the sampling rate of the accelerometer hardware in use. Features are not extracted from the accelerometer signal.

4.2 Training

Excluding sign language, gesture recognition does not have a standardized vocabulary [7]. Especially in the case of ges-

[1] http://hexler.net/software/touchosc
[2] http://cycling74.com/2008/07/28/max-5-guitar-processor-part-1

tures intended for artistic purposes, it is desirable for musicians to define personalized gestures to use in the recognition process. User-dependent gesture recognition promotes the use of gestures that are natural to the guitarist and conform with their performance style.

To recognize personalized gestures, the system requires the user to first form a gesture dictionary by recording one training instance for each desired gesture. Training mode is toggled by clicking the *learn* button on the gesture recognition interface (Figure 2). The training process involves clicking the *record* button, performing the gesture, and clicking the *record* button again to define the start and stop times of the gesture command. The user must then select one gesture from the gesture dictionary to be used in the recognition process, which determines the presence or absence of this gesture in the incoming accelerometer signal.

4.3 Recognition

The first step of the gesture recognition algorithm is signal segmentation. A sliding-window representation of the accelerometer signal is used to segment the continuous gesture commands into analysis frames. Formally, analysis frames are created by multiplying time-delayed versions of a rectangular window function

$$w[n] = \begin{cases} 1 & \text{if} \quad n \in \{0, 1, \ldots, N-1\} \\ 0 & \text{else} \end{cases} \tag{2}$$

with each component of the three-dimensional accelerometer signal (1). The length of the window $N \in \mathbb{N}^+$ corresponds to the number of samples comprising the user-selected reference gesture from the gesture dictionary. In practice, three delay lines of length N—one for each dimension of the accelerometer signal—are used to represent the current analysis frame. Each delay line is updated as new samples are received from the accelerometer.

The second step of the gesture recognition algorithm is binary gesture classification on the current analysis frame—either the user-selected reference gesture has been performed or not. The gesture classification algorithm uses template matching to compare a vector representation of the current analysis frame $G_a \in \mathbb{R}^{3N}$ to a vector representation of the single reference gesture $G_r \in \mathbb{R}^{3N}$ using the technique of cosine similarity [14]

$$S = cos(\theta) = \frac{G_r \cdot G_a}{||G_r||\,||G_a||}, \tag{3}$$

which measures the cosine of the angle between two feature-space vectors. When the performed gesture and the reference gesture are similar, the angle between the vectors is small and the cosine of this angle is close to one, since $\cos(0) = 1$. A user-defined threshold parameter $\gamma \in \{\mathbb{R} : 0 \leq \gamma \leq 1\}$ determines the gesture recognition sensitivity, such that a gesture is recognized when $S \geq \gamma$. To accommodate computers with varying computational power, a user-defined hop size parameter $H \in \mathbb{N}^+$ determines the frequency in which analysis frames are presented to the gesture classification algorithm. By default $H = 1$: with each new accelerometer sample, the gesture classification algorithm processes the current analysis frame.

4.4 Feedback

Stompboxes provide tactile, visual,[3] and auditory feedback to the performer. By alternatively using gesture recognition to switch audio effects, the same problem as open-air controllers is encountered: the tactile and visual feedback channels are lost [15]. Effort is made to restore these lost feedback channels. To restore visual feedback, a GUI widget flashes on both the Max/MSP and iPhone interface when a gesture is successfully recognized. Effort was made to provide vibrotactile feedback by vibrating the iPhone when a gesture is successfully recognized; however, the propagation of the vibration through the body of the guitar to the musician was too dampened to serve as reliable feedback.

[3]Stompboxes typically have an LED that indicates audio effect activation.

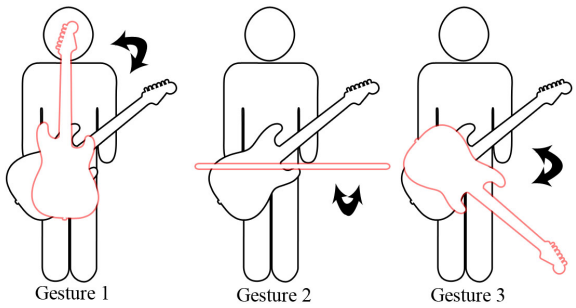

Figure 3: Three gestures used in the experiments evaluating the gesture recognition system.

Table 1: Recall metric for the user-independent and user-dependent gesture recognition experiments.

	User-independent	User-dependent
Gesture 1	94%	97%
Gesture 2	99%	100%
Gesture 3	84%	100%
Average	86%	99%

5. EVALUATION

To evaluate the gesture recognition system, two experiments were conducted on ten right-handed test subjects, half of which considered themselves a guitarist. For each experiment, offset correction was performed on the accelerometer signal as the test subject held the guitar naturally in the home position. No accelerometer signal resampling was performed and the hop size parameter was $H = 1$.

The first experiment measures the recall of the gesture recognition system under the condition that the reference gestures are trained by a different user (user-independent recognition). Prior to conducting the experiment, the gestures illustrated in Figure 3 were demonstrated for each test subject. The first gesture involves raising the neck of the guitar, such that it is parallel with the guitarist, and back to the home position; the second gesture involves rotating the guitar 90 degrees, such that the fretboard is facing upwards, and back to the home position; and the third gesture requires the guitarist to lower the neck of the guitar before returning to the home position. The subject was asked to test the first gesture three times. In these trial runs, the recognition threshold γ was adapted accordingly. The subject was then asked to perform the gesture ten times and this process was repeated for the remaining two gestures. The results of this experiment are presented in Table 1.

The second experiment measures the recall of the gesture recognition system under the condition that the reference gestures are trained by the test subject (user-dependent recognition). The subject was asked to train the same three gestures as in the first experiment (the *record* button was clicked for them). The experiment then continued in an identical manner as the first experiment. The results are presented in Table 1.

According to Mäntyjärvi et al. [8], nearly perfect accuracy is required to ensure that the user does not abandon the gesture recognition system. Although a training dataset containing the gestures illustrated in Figure 3 is provided with the system, the results suggest that guitarists should train personalized gestures for use with the implemented recognition system.

6. CONCLUSION

Sensor-based gesture recognition is investigated as a possible solution to the problem of managing a large array of audio effect stompboxes in the context of a live guitar performance. An open-source collection of digital audio effects and a gesture recognition system have been developed in Max/MSP.[4] The system allows guitarists to define the structure of audio effects occurring in a piece of music and to train personalized gestures for recognition. When a gesture is successfully recognized, the appropriate audio effects are automatically applied to the guitar signal and visual feedback is provided to the guitarist.

An evaluation of the gesture recognition system on ten test subjects yielded 86% recall for user-independent recognition and 99% recall for user-dependent recognition. These results suggest that gesture recognition is a viable alternative to manually activating audio effects during a live guitar performance. With the development of this system we hope that guitarists will consider kicking aside their stompboxes and usher in the power of gestural control. We hypothesize that the use of realtime gesture recognition to toggle audio effects, versus the traditional method of depressing a sequence of stompboxes, will increase guitarists' stage mobility and opportunity for audience interaction.

Work is currently being done to extend the gesture recognition system to extract features from the accelerometer signal using the time and frequency-domain features proposed by Dargie [3]. We also plan to extend the binary gesture classification algorithm to differentiate between multiple trained gestures and to allow multiple training instances for each gesture. With multiple training instances, the recognition threshold parameter γ could be estimated from the training data. For the continuous control of audio effects, we plan to investigate alternative mappings between the accelerometer signal and effect parameters to take into consideration the natural and expressive gestures of a guitarist while performing. Furthermore, the use of a vibrotactile belt will also be investigated to restore tactile feedback.

7. ACKNOWLEDGMENTS

This research was generously supported by the Social Sciences and Humanities Research Council of Canada. Special thanks are owed to the participants of the McGill Music Hack Day who kindly partook in the experiment.

8. REFERENCES

[1] Bettens, F., and T. Todoroff. 2009. Real-time DTW-based gesture recognition external object for Max/MSP and Puredata. In *Proceedings of the Sound and Music Computing Conference*, Porto, Portugal, 30–5.

[2] Bevilacqua, F., B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. 2010. Continuous realtime gesture following and recognition. In *Gesture in Embodied Communication and Human-Computer Interaction*, Volume 5934 of Lecture Notes in Computer Science, 73–84. Berlin: Springer-Verlag.

[3] Dargie, W. 2009. Analysis of time and frequency domain features of accelerometer measurements. In *Proceedings of the International Conference on Computer Communications and Networks*, San Francisco, CA, 1–6.

[4] Françoise, J. 2011. Realtime segmentation and recognition of gestures using hierarchical Markov models. Master's thesis, Université Pierre et Marie Curie.

[5] Hofmann, F., P. Heyer, and G. Hommel. 1998. Velocity profile based recognition of dynamic gestures with discrete hidden Markov models. In *Gesture and Sign Language in Human-Computer Interaction*, Volume 1371 of Lecture Notes in Computer Science, 81–95. Berlin: Springer-Verlag.

[6] Lähdeoja, O. 2008. An approach to instrument augmentation: The electric guitar. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Genova, Italy, 53–7.

[7] Liu, J., Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, Galveston, TX, 1–9.

[8] Mäntyjärvi, J., J. Kela, P. Korpipää, and S. Kallio. 2004. Enabling fast and effortless customisation in accelerometer based gesture interaction. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia*, College Park, MD, 25–31.

[9] Mitra, S., and T. Acharya. 2007. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics* 37 (3): 311–24.

[10] Puckette, M. 2002. Max at seventeen. *Computer Music Journal* 26 (4): 31–43.

[11] Schlömer, T., B. Poppinga, N. Henze, and S. Boll. 2008. Gesture recognition with a Wii controller. In *Proceedings of the International Conference on Tangible and Embedded Interaction*, New York, NY, 11–14.

[12] Schnell, N., R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller. 2005. FTM—complex data structures for Max. In *Proceedings of the International Computer Music Conference*, Barcelona, Spain.

[13] Schnell, N., and D. Schwarz. 2005. Gabor, multi-representation real-time analysis/synthesis. In *Proceedings of the International Conference on Digital Audio Effects*, Madrid, Spain, 122–6.

[14] Thiebaut, J., S. Abdallah, A. Robertson, N. Kinns, and M. Plumbley. 2008. Real time gesture learning and recognition: Towards automatic categorization. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Genova, Italy.

[15] Vigliensoni, G., and M. Wanderley. 2010. Soundcatcher: Explorations in audio-looping and time-freezing using an open-air gestural controller. In *Proceedings of the International Computer Music Conference*, New York, NY, 100–3.

[16] Wanderley, M., and P. Depalle. 2004. Gestural control of sound synthesis. *Proceedings of the IEEE* 92 (4): 632–44.

[17] Wright, M. 2005. Open Sound Control: An enabling technology for musical networking. *Organized Sound* 10 (3): 193–200.

[4] `http://github.com/gburlet/gesture-recognition`