

Gestural Controlled Virtual Percussion Instruments

José Fornari (Tuti)

University of Campinas (UNICAMP)
Interdisciplinary Nucleus of Sound Communication (NICS)
São Paulo - Brazil



Figure 1: (left) Computer model of virtual percussion instrument with controlling interface.
(right) Musician playing virtual percussion instrument.

Abstract

Nowadays, gestural interfaces of video-games, such as the nintendo wiimote, can be used as a powerful, yet low-budget, haptic sensor to capture in real-time keen movements of a musician performer, through its embedded tridimensional accelerometer. Here is described an ongoing development of computer models of real-time sound syntheses dynamically controlled by gesture captured with such sensor. This work presents the usage of this methodology to create virtual percussion instruments inspired by three real percussions: a Berimbau, a Shaker and a Bell. This paper presents these three virtual percussions and describes their usage, in remote musical interactivity; exploring extended techniques that can be applied to artistic, ludic and educational activities.

Keywords: gestural interface, sound synthesis, ecological modeling, extended techniques

Author' contact:

tutifornari@gmail.com

1. Introduction

Among all faculties of the human brain, one of the most amazing is its ability of simultaneously controlling all muscles and articulations of the human anatomy, in order to enable body expression. This extremely complex task seems to be done effortlessly, dynamically and, in most of the time, unconsciously. Body movements can be trained and refined by repetition, so that an artist can achieve the necessary fine-tuned movements to express his or her art.

Musical performances strongly rely on well-trained and fine body movements, so the musician can achieve the goals of his or her musical expressive performance while following the musical notation or convention.

Modern technology allowed access to many aspects of sound perception and consequently, made possible the retrieval of music information. With the computer aid, it is now possible to analyze, transform and synthesize musical aspects; in time and frequency domain. With that, many methods of sound synthesis were developed as computer models - each method exploring a different mathematical strategy to emulate sounds as being generated by physical processes and/or real acoustic musical instruments.

There are basically two types of classical sound synthesis methods: Linear and Nonlinear. Linear methods can be represented by linear equations systems. Nonlinear methods are given by nonlinear equations, which are not reducible to independent equations. Some nonlinear models are described as feedback systems. Linear methods normally allow full and independent control over all sound partials, however, its complexity (and so, the required computer processing) increases proportionally with the complexity of the generated sound. Nonlinear methods, on the other hand, can generate complex waveforms with less computation processing, but without independent control over each partial, which sometimes lead to unpredictable sound results; so the design of sounds using nonlinear methods are mostly done in an empirical, trial-and-error basis. Classical types of linear methods are: the Additive and the Subtractive syntheses. Classical types of nonlinear syntheses are: the FM (frequency modulation), the PM (physical modeling) and the Granular synthesis [Cook 2002].

The great richness of sounds generated by nonlinear methods of synthesis, together with its low computational processing required, make these methods quite enticing to be used in real-time computer music performances. However, its non-intuitive controlling is a major constrain, that was

probably the reason why methods such the FM synthesis were replaced by editing methods, such as the Wavetable [Moore 1990].

The computer technology unlocked the access to control several aspects of the sound, but the usual controlling interfaces of personal computers (the mouse and the keyboard) are reportedly not enough to handle several simultaneous aspects of the sound synthesis parametrization. As we said before, our brain has the ability of simultaneously controlling all aspects of our body movement. Therefore, the motor cortex is capable of simultaneously controlling all joints and body parts, through the nervous system, in order to perform gestures [Parncutt 2002].

Gestural data collected by dynamic sensors can be a way of controlling sound syntheses – linear the nonlinear ones - in a way that their parameters can be intuitively adjusted by the user, as he or she tries it on and learn new movements. With that, we are attempting to use the intrinsic incapacity of controlling and learning complexes body movements – given by the motor cortex - to spontaneously controlling complex parameters of sound synthesis methods. If we can use this innate capacity of the motor cortex to control models of sound synthesis, we may as well be able of truly explore extended techniques in computer models, emulating musical instruments.

This work describes the recent developments of computer models written in PD (PureData) controlled by gestural data, retrieved by the tridimensional accelerometer embedded within the video-game remote-control of nintendo wii, here referred by its nickname; the wiimote. As further described, we use this gestural sensor to control real-time computer models of virtual percussion instruments.

2. Related Work

There are several examples, in the literature, of studies using different types of interactions and sensors for musical expressivity. In the 18th century, there was the advent of the first (dated) electric musical instrument; the Denis d'or, created by Václav P. Diviš [Tübingische 1754], using electrified wires to synthesize and control sounds in real-time. In 1920, there was the invention of Theremin, by Lev S. Termen [Glinsky 2000]. This instrument has a loop antenna working as a proximity sensor in which the musician (normally by the proximity of their hands, without actually touching the antenna) dynamically controlled the frequency of a continuous sine-wave. Probably inspired by this instrument, Max V. Mathews conceived the Radio-Baton¹; also a proximity (capacitive) sensor device which tracks the motions of the tips of 2 batons, in a tridimensional bounded space.

Nowadays, the research on this field is known by different nomenclatures, such as: Haptic interaction, Input devices and Gesture interfaces; gathered by international congresses, such as NIME (New Interfaces for Musical Expression) and research

centers, such as the IDMIL² (Input Devices and Music Interaction Laboratory, at the McGill University).

Specifically similar works with the one presented here are: BodyMusic, a cyber instrument that enables musicians to interactively and intuitively control musical tonality and the melody, through body motions [Horace 2005], and the system developed by [Peng 2009]; a computer model that uses a wiimote as gestural interface to capture the movements of a conductor, in real-time.

3. Developing Virtual Percussions

In order to achieve the development of gesture-controlled virtual percussion instruments, we sought a way of dynamically capturing the gestural data and transmit it wirelessly to a laptop, in real-time. This data dynamically controlled the sound synthesis models programmed in PD. The next sessions explain this developments.

3.1 Gestural Information Retrieval

As said before, we opt to use the remote-control of the nintendo video-game wiimote. This is an inexpensive, wireless (bluetooth), portable interface that is equipped with one tridimensional accelerometer, one infrared camera that tracks up to four reflexion points simultaneously, as well as several buttons (A, B, -, +, 1, 2, Up, Down, Left, Right and Home).

The accelerometer embedded in this device is the Analog Devices ADXL330; a 3-axis linear accelerometer that provides the wiimote with motion-sensing capability. It has a ± 3 g sensitivity range, 8 bits per axis, at a 100 Hz update rate.

The data is transmitted over a wireless bluetooth connection that uses a Broadcom 2042 chip; designed for devices that conform to the Bluetooth Human Interface Device standard, such as keyboards and mouses.³ The accelerometer data is transmitted in seven data streams: the raw acceleration data: x, y and z; and their related rotational angles: pitch, yaw and roll; as well as a general acceleration data stream, as described in the Figure 2:

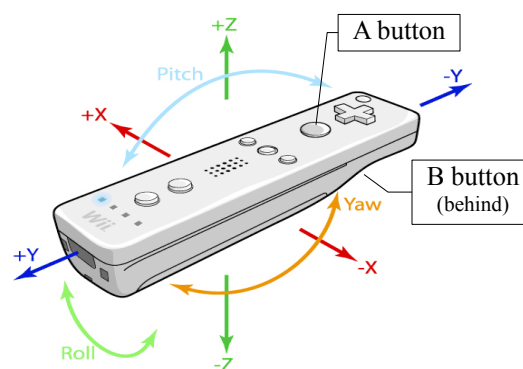


Figure 2: The tridimensional accelerometer data and buttons, used in the virtual percussion instruments.

1 <http://emfinstitute.emf.org/exhibits/radiobaton.html>

2 <http://www.idmil.org>

3 <http://johnnylee.net/projects/wii/>

This data is retrieved by the software OSCulator⁴, a MAC OS X program that links bluetooth devices (such as the wiimote) with musical softwares, via OSC (Open Sound Control) networking protocol⁵. According to its designer, OSCulator is capable of handling up to 8 wiimotes simultaneously. We have successfully tested it with 4 wiimotes, each one controlling a different virtual percussion patch.

3.2 Real-time Sound Syntheses Models

Once we have the gestural data represented by the accelerometer data streaming, in OSC format, the next step is to use it to control our computer models of sound syntheses. The control must be done continuously, and in real-time, without any perceived delay, so the musician feel comfortable playing a virtual instrument in which the sound is immediately synthesized and controlled by his or her gesture.

We used the programming environment of PureData, also known as PD, to develop such models. PD is a multi-platform, open-source visual programming language, specially created for media data controlling and processing [Puckette,1997].

Similar to MAX/MSP⁶, PD is also programmed by connecting building blocks, such as functions (the “objects”), and other boxes, as the: “message”, “number”, “bang”, “array”, and so forth. These interconnected structures are called “patches” and resemble a data-flow diagram, throughout which the real-time processing of “pure” data (audio, image, video, MIDI, text, control, etc.) occurs. PD has two versions: the official one, maintained by its designer, and an extended one, known as PD-extended⁷, developed by a large community of programmers, in which several major expansions were included, as object libraries. We used this version to program the virtual percussion instruments presented here. The next Figure shows an example of a PD patch; an excerpt of the additive synthesis, used in one of our models.

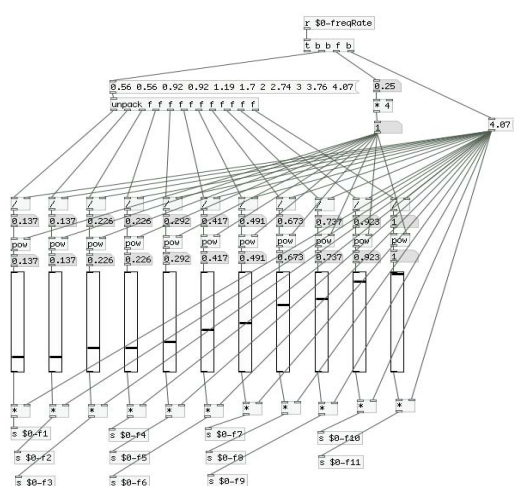


Figure 3: Example of PD patch

4 <http://www.osculator.net/>

5 <http://opensoundcontrol.org/>

6 <http://cycling74.com/products/maxmsp/jitter/>

7 <http://puredata.info/>

4. Virtual Percussions

Among other additions, PD-extended has objects to receive and transmit OSC data. The object “dumpOSC” receives this protocol from an specified port / UDP, and outputs all data received, throughout its only outlet. We used this object to receive data from OSCulator; the gestural information retrieved by the wiimote accelerometer, that controls the computer models of these sound syntheses.

Each model is written as a PD patch using a different synthesis method to generate percussion sound. There is no audio samples of real (acoustical) musical instruments used in these patches, such as in a Wavetable synthesizer or a Sampler machine. Each sound is generated in real-time, emulating the features of the actual acoustical musical instruments, in which they were inspired. This way, the system may easily allow the user to explore extended techniques of music that would otherwise be impossible to achieve by real acoustical instruments.

These virtual percussion instruments have some movement features in common. The rotational “Pitch” - as shown in Figure 2 - was used to control the musical pitch. “Roll” was used to control sound localization ITD and ILD functions [Blauert 1983]. Button “A” was used to enable pitch change, through the Pitch accelerometer data, and button “B”, was used to pause the amplitude envelope sound decay.

4.1 vBerimbau

Berimbau is a typical Brazilian percussion instrument, used in sessions of *Capoeira*, a Brazilian dance and martial art. It has one single string, that is hit by a wooden stick and modulated by a stone. A filtering sound effect is achieved by a gourd, by pressing it against the body, while playing. The next Figure shows a *berimbau*, its compounding parts, and a wiimote with its extension, giving to it an extra accelerometer, that creates our virtual *berimbau*, here named as: vBerimbau.

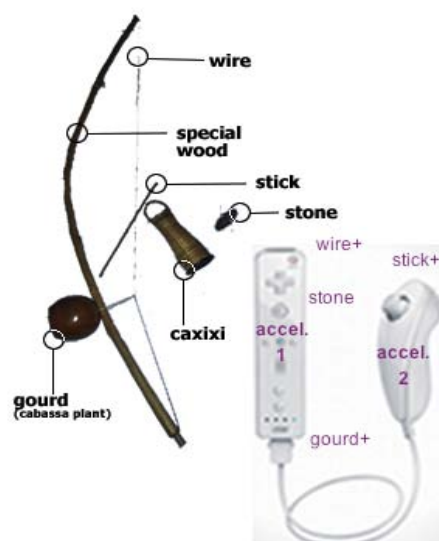


Figure 4: *Berimbau* and the vBerimbau.

4.2 vShaker

Shaker refers to a group of percussion instruments without defined pitch that generates sound by shaking movements. This implementation uses each coordinate of the tridimensional accelerometer to control distinct shaker timbres. Subtractive synthesis is the method used to create it, mostly because it allows the easy creation of rich stochastic sounds, rather than deterministic ones (with clear pitch). There is a type of shaker also used together with *Berimbau*. It is named *Caxixi*, also shown in Figure 4.

We used a white noise generator, as provided in the PD environment, by the object “noise~”, to create stochastic sound, then, we tweak it, with band-pass filter objects, to create and control each intended shaker timbre. Figure 5 shows a typical waveform and spectrum of the virtual shaker; the vShaker.

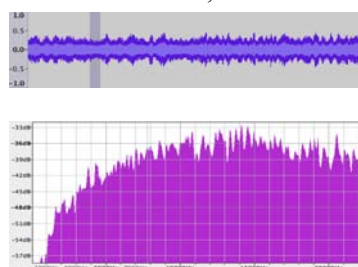


Figure 5: Example of vShaker waveform and spectrum.

4.3 vBell

We implemented a virtual Bell by using the Risset's mapping of inharmonic partials of real bells [Risset, 1969]. This is implemented as an additive synthesis with eleven partials. Thus, there are eleven sinusoidal oscillators in this patch, controlled in amplitude, frequency and phase; each one related to one partial of the Risset's bell model. Figure 3, shows an excerpt of this patch implementation, with eleven parameters.

Additive synthesis are linear models inspired in the Fourier series, in which any sound can be represented by a sum of partials, given by sine waves. The caveat is that, the more complex is the sound synthesized, the large is the number of oscillators required in an additive synthesis model.

The PD patch implemented for vBell was able to easily process and control in real-time the parameters of this additive synthesis. The Figure 6 shows an excerpt of its waveform and respective spectrum.

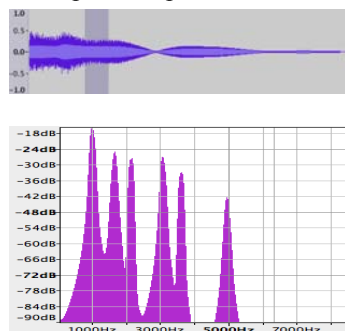


Figure 6: Example of vBell waveform and spectrum.

5. Conclusion

The creation of a family of virtual percussion instruments, synthesized and controlled in real-time, by gestural data, seems to be an interesting way of exploring new and truly extended techniques, as well as allowing ecological modeling, by transmitting the gesture data retrieved throughout the internet cyberspace, which can be of great interest in forthcoming applications for games, interactive musical performances, as well as for educational purposes. Some of these models were selected as finalist for the File Prix Lux 2010 festival, by the name of CybeRITMO (<http://www.fileprixlux.org>).

It is important to notice that the design of these virtual musical instruments, by no means intended to replace real acoustical percussions, as its sonic richness seems to be unsurpassable. These patches were solely inspired by these instruments and only attempted to model part of its timbre complexity, in a computational environment so it could be controlled and used to explore extended techniques impossible to be reached by actual real instruments. There are several videos of professors, researchers and musicians playing with these virtual instruments. They can be watched at the following link: www.youtube.com/nicsunicamp.

Acknowledgements

The author would like to thank the Interdisciplinary Nucleus for Sound Communication, where he is researcher; the Faepex/Unicamp, for the financial support and all musicians and academics who spent their time testing and playing with these virtual percussion instruments.

References

- TUBINGISCHE, 1754. *BERICHTE VON GELEHRTEN SACHEN*, XXX, p. 395.
- GLINSKY, A., 2000. *THEREMIN: ETHER MUSIC AND ESPIONAGE*. URBANA, ILLINOIS: UNIVERSITY OF ILLINOIS PRESS.
- PENG, L., GERHARD, D., 2009. *A WII-BASED GESTURAL INTERFACE FOR COMPUTER-BASED CONDUCTING SYSTEMS*. INTERNATIONAL CONFERENCE ON NEW INTERFACES FOR MUSICAL EXPRESSION.
- HORACE, H. S., KWONG, B., LAW, K. C. K., 2005. *BODYMUSIC: A NOVEL FRAMEWORK DESIGN FOR BODY-DRIVEN MUSIC COMPOSITION*. IN *ACE'05: PROCEEDINGS OF THE 2005 ACM SIGCHI. INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER ENTERTAINMENT TECHNOLOGY*, PAGES 342-345, NEW YORK, NY, USA, 2005. ACM PRESS.
- PUCKETTE, M. S., 1997. *PURE DATA*. IN: *PROCEEDINGS OF THE INTERNATIONAL COMPUTER MUSIC CONFERENCE*, PP. 224-227. INTERNATIONAL COMPUTER MUSIC ASSOCIATION.
- RISSET, J. C., *AN INTRODUCTORY CATALOG OF COMPUTER-SYNTHESIZED SOUNDS*, MURRAY HILL, NEW JERSEY: BELL LABORATORIES, 1969.
- COOK, P. R., 2002. *REAL SOUND SYNTHESIS FOR INTERACTIVE APPLICATIONS*. A K PETERS, LTD. 263 PAGES. ISBN: 978-1-56881-168-0.
- MOORE, F. R., 1990. *ELEMENTS OF COMPUTER MUSIC*, PP. 159-169, PRENTICE-HALL, ENGLEWOOD CLIFFS, NJ.
- PARNCUTT, R., MCPHERSON, G., 2002. *THE SCIENCE & PSYCHOLOGY OF MUSIC PERFORMANCE: CREATIVE STRATEGIES FOR TEACHING AND LEARNING*. OXFORD UNIVERSITY PRESS US.