

ZATLAB: A Gesture Analysis System to Music Interaction

André Baltazar #, Luís Gustavo Martins # and Jaime S. Cardoso *
abaltazar@porto.ucp.pt, lmartins@porto.ucp.pt, jaime.cardoso@inescporto.pt

#UCP - School of Arts, Center for Science and Technology in the Arts - Porto, Portugal

*INESC TEC (formerly INESC Porto) - Faculdade de Engenharia da Universidade do Porto, Portugal

Abstract—The human gesture is an important means of expression and interaction with the world, and has an important role on the perception and interpretation of human communication. Over the years, different approaches have been proposed to capture and study human gestures and movements by various fields of study, namely Human Computer Interaction or Kinesiology (the scientific study of the human motion properties). This paper proposes a new modular system, named Zatlabs, that allows to control, in real-time, music generation through expressive gestures, allowing dancers and computer music performers (among others) to explore novel ways of interaction with music and sound creation computer tools. The system is based on real-time, non intrusive, human gesture recognition, which analyzes movement and gesture in terms of low level features (e.g. distance, velocity, acceleration) and high level features (e.g. quantity of movement), and uses machine learning algorithms to map them into various parameters in music generation algorithms, allowing a more semantically and artistically meaningful mapping of human gesture to sound.

Index Terms—Gesture analysis, real-time, music generation, kinect.

I. INTRODUCTION

The gesture is an important part of human communication, and it is used often - even unconsciously - as a means of expression and interaction with the world, having a strong impact on how humans perceive and interpret themselves [1]. There is an important distinction between gesture and movement [2]. Gesture presupposes an intention, a meaning and the movement is the physical action itself (e.g. a set of arm movements waving composes the gesture of saying goodbye).

Nowadays, different methods can be used to capture human movements using, for instance, video cameras, body wearable sensors or external sensors, such as infra-red Motion Capture (MOCAP) systems. All these processes allow capturing and gathering signal data, from where relevant gesture features can be extracted for further analysis and processing (e.g. estimating amplitude, periodicity, rhythm, diversity, etc., of a gesture or movement). Such features can then be used as inputs for real-time algorithmic music composition systems, paving the way for novel expressive and artistic works, where humans and machines interact in a more semantically and artistically meaningful dialog. The motivation behind this particular system, is the possibility of analyzing human gesture at a higher level. Hence, the musical output can present more abstract and complex relations with the human gestures, rather than the direct mapping of human movements.

This paper describes the Zatlabs, a modular system that allows the capture and analysis of human gestures in an unintrusive manner (using a Microsoft Kinect [3] video capture system and a custom application for video feature extraction and analysis developed using openFrameworks¹). The extracted gesture features are subsequently interpreted in a machine learning environment (provided by Wekinator [4]) that continuously modifies several input parameters in a computer music algorithm (implemented in Chuck [5]). Through this kind of gesture analysis one can perceive an higher level of interaction, where instead of a linear relation between actions and music, it presents a more intricate and complex result.

This paper is organized as follows. Section 1 presents a brief introduction to the proposed system. Section 2 discusses the previous work on this area, paving the ground for this work. The implementation and integration sections explains the tools used and how the system was developed. Section 4 presents and discusses some preliminary evaluation results. The paper concludes in Section 5 with some final remarks and plans for future work.

II. BACKGROUND

The field of human movements and gesture analysis has, for a long time now, attracted the interest of many researchers, choreographers and dancers. Since the beginning of the twentieth century, a significant corpus of work has been conducted on music psychology related to movement perception [1]. On the same topic, many approaches have been proposed to translate the human physical movement and gesture into digital signals that can then be used to control musical parameters in an algorithmic music composition system. During the 90s, Axel Mulder [6] distinguished three techniques that remain an important reference in relation to tracking human movement: Inside-In [7], Inside-Out [8], [9] (both approaches use sensors attached to the body) and Outside-In [10], [11], [12] (usually only uses video signals). Over the years, it has been possible to witness very interesting and remarkable achievements that followed the less-intrusive Outside-In technique. For instance, in 2004, Camurri [10], developed a study using video sequence analysis systems to compute the mass centre of a dancer as well as its evolution over space throughout a performance. In 2005, Guedes [11] realized that analyzing the number of pixels in video sequences whose luminance levels changed,

¹<http://www.openframeworks.cc>

due to repetitive movements, allowed to detect periodicities in the video signal. Using the Goertzel Algorithm, a variation of the common spectral analysis algorithms (such as the Fast Fourier Transform), permitted the efficient computation of the fundamental frequency from the video signal, and subsequently estimate the rhythm of the physical movements on the video stream. In 2008, Naveda [12] proposed a digital representation for Samba dance gestures. He developed tools to relate the music and the dance on a metrical level, proposing relevant heuristic methods to connect music and dance.

More recently, there were several articles at the NIME 2011 Conference that covered work developed on this topic. For instance, Polloti et al. [13] studied both sound as a means for gesture representation and gesture as embodiment of sound. Bokowiec [14], proposed a new term, “Kinaesonics”, to describe the coding of real-time one-to-one mapping of movement to sound and its expression in terms of hardware and software design. And Yoo [3] described the use of a Microsoft Kinect to directly map human joint movement informations to MIDI.

In summary, the corpus of research developed in this area in recent years has made available a new basis for the creation of computational models inspired on the human expression and perception of movement, gesture and rhythm. These models have been used to test and refine existing theories, and to create interactive systems that are able to perform perceptually and artistically relevant tasks in real-time. Nevertheless, we are still looking for more satisfactory approaches and solutions to understand, interpret and creatively use human gestures in algorithmic musical composition contexts. In this sense the Zatlab system intends to fill some of the gaps still unexplored, such as the ability of machine learning algorithms to give a more high level relation between the Human gesture and the music output.

III. SYSTEM DESCRIPTION

The Zatlab combines the use of a Microsoft Kinect camera (which, in addition to the VGA video stream, also maps a depth field of the scene) and its software libraries, with the visual representation developed, using openFrameworks, that enables gesture feature mapping and analysis. The gesture features extracted are then sent to a Machine Learning module (implemented using Wekinator) and ultimately to an algorithmic music composition module (implemented using ChucK [5]). The diagram in Figure 1 presents the system main building blocks, along with some information about what is being performed in each module, as well as the type of information passed among modules.

All modules in the system communicate using Open Sound Control (OSC) [15]. This allows to implement a highly modular system, where any module can be reused for a different purpose or in a different scenario (e.g. use the extracted gesture features to control a video generation module, or add additional controllers to the system such as smart phones or touch tablet devices).

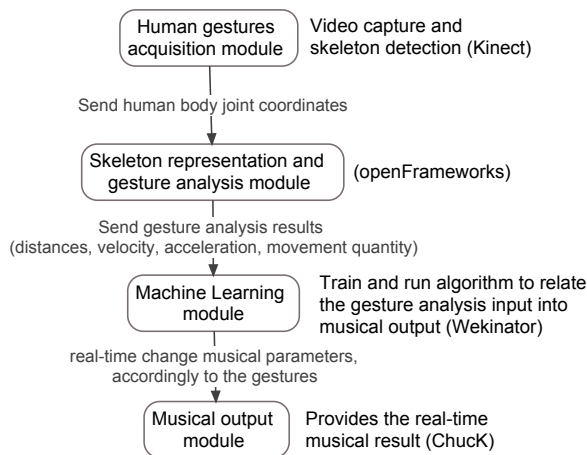


Fig. 1. System Block diagram.

The following sections will present a more detailed description of each system block and its integration into the system.

A. Human gesture acquisition module

The acquisition of human gestures should be as accurate as possible, to ensure a proper analysis of their features. In a previous version of the system, this module was implemented using a 2D webcam, whose output was then analyzed using image segmentation algorithms (as described in [16]). More recently, with the Microsoft Kinect, it became possible to easily obtain a full-body detection using the depth information from the video signal. When compared with the previous webcam version, one can affirm that it becomes much more simple to detect and track a foreground object/person. The “traditional” computer vision tracking problems, such as light constraints or background/foreground separation can now be easily solved using this new technology. Along with Kinect’s launch, various drivers and software modules were developed to provide skeleton detection, namely the drivers by Primesense², OpenNI³ and SenseBloom⁴. There is a particular software module known as “OSkeleton” (developed by Sensebloom) that extracts and transmits all the human body joint coordinates, in a text list, through OSC.

Using this information, it becomes relatively simple to implement a visual representation of the skeleton information received from the Kinect, as explained in the following section.

B. Skeleton representation and gesture analysis module

This module (developed in openFrameworks) is responsible for receiving the skeleton joints information extracted from

²<http://www.primesense.com>

³<http://openni.org/Documentation>

⁴<https://github.com/Sensebloom/OSkeleton>

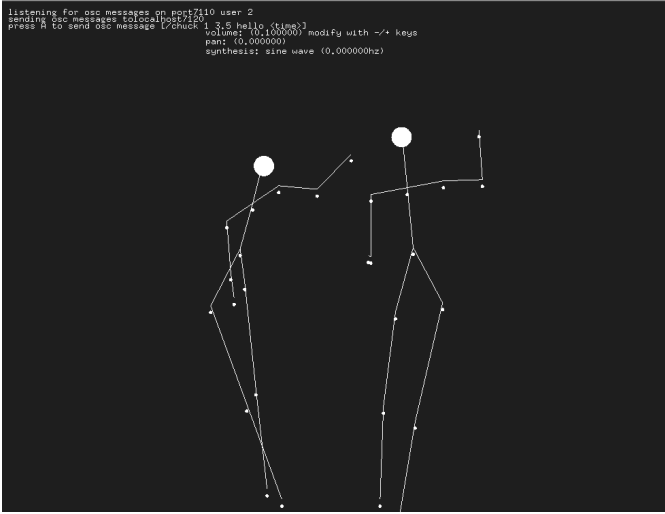


Fig. 2. Human skeleton representation and gesture analysis (implemented in openFrameworks).

the Kinect, and subsequently interpreting and drawing in real-time a visual representation of the main joints and segments of a human skeleton (see Figure 2).

The gesture analysis is implemented by computing several low level features, such as: euclidean distances of joints in consecutive frames, velocities and accelerations, as well as some high level features, such as the estimate of “quantity of movement” (QOM) of the subject.

The QOM feature is computed using a running average method in which a circular vector (whose size can be specified by the user, e.g. window size, WS , of 100 samples), is updated by replacing the oldest distance measures with the most recent ones and proceeding with the average calculation at a refresh rate (RR) established also by the user (e.g. at 5 new samples) (see eq. 1). This way, the user can determine if he prefers a fast rate of actualization and a more responsive, “hysteric”, system, or a slow rate and a more slow passed system.

$$avg = \frac{\sum_{n=0}^{WS} d_n}{WS} \quad (1)$$

These computed features are then passed to the following machine learning module.

C. Machine learning module

This module receives the gesture analysis data, and uses it to control different music parameters, based on a Machine Learning approach. For that purpose, Wekinator, a simple to use and open source software package that uses machine learning techniques to build real-time, interactive systems, is applied [4].

For the current system, a Neural Network (NN) was trained using four different gestures (repeated around 100 times each), which correspond to the output of four different sets of parameters (that will be used to control the musical composition algorithm, described in the next section). For instance, if the

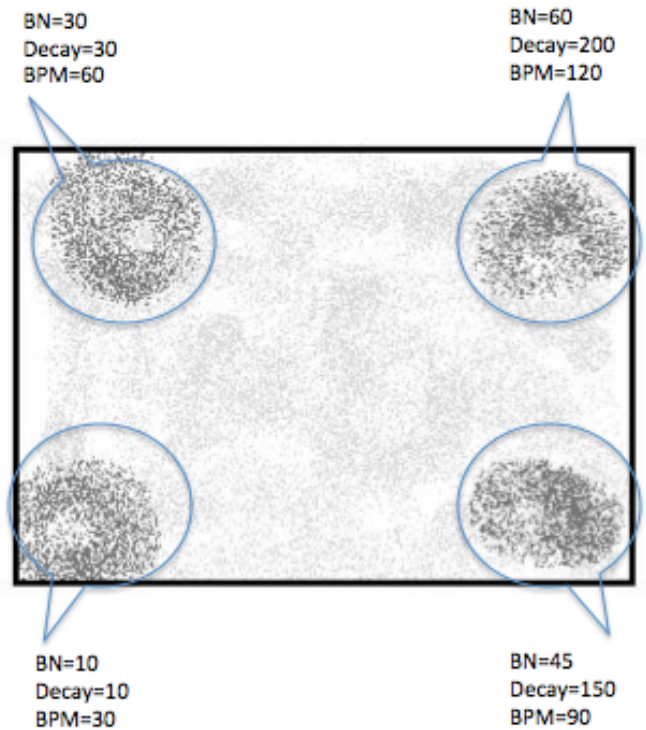


Fig. 3. Visual feedback of the samples used to train the NN. The four high lined clouds of samples and their respective output.

gesture under analysis is waving the hand on the top right corner (above your head), the output should be a set of features corresponding to a MIDI note number 30, with a decay value of 30ms and a tempo value of 60 beats per minute (BPM). If you then wave your hand on the top left corner this should output a MIDI note number 60, a decay value of 200ms and a BPM of 120, and so on (Figure 3).

After this initial training of the NN, the system should be able to interpolate the output for any state that falls in-between the trained gestures. This set of features which depend on the gesture analysis performed by the NN trained using Wekinator are then sent to the Algorithmic Composition module, described in the next section.

D. Algorithmic Composition Module

The musical output module consists on a musical composition algorithm implemented using the ChucK software package [5]. ChucK is a programming language for real-time audio synthesis and composition. It is a time-based programming model that allows high precision and expression. One of the main qualities is the ability of adding and modifying code on-the-fly, as well as being able to dynamically control rates and parameters.

The sound synthesis used in this module is based on a physical model of a set of plucked strings, along with a simple percussion feedback. The percussion rhythm consists of a list of shaker instruments that are randomly chosen to

```
[[1, 15], [2, 18], [1, 21], [0, 6], [1, 13], [2, 18], [1, 11], [0, 9],
[1, 18], [2, 21], [1, 24], [0, 11], [1, 16], [2, 21], [1, 13],
[1, 21], [2, 18], [1, 15], [0, 13], [1, 12], [2, 9], [1, 8], [0, 0]
] @=> int PlaySequence[];
```

Fig. 4. Note scale, each pair of values represents the string to be plucked and the note to add to the Base Note (in midi).

be played at random beats. There is a pre-programmed scale of notes to be plucked along with a choice of which strings to pluck for each note (Figure 4). This note sequence is controlled by the gesture features extracted from the video signal. This mapping is implemented in a simple way, where the extracted gesture features influence the base MIDI note (BN) of the scale, the decay of the plucked strings and the BPM of the overall music piece. As a result, and depending on the gestures, the system allows going from a really bass and calm environment (BN=10; decay=10; BPM = 30) to a really high register, intricate and frenetic environment (BN=60; decay=200; BPM=180).

Since the interaction occurs in real time and there is a musical structure to respect, there was the need of implement a system lock at the time a note is being played. This was assured by only allowing external interruptions at the end of the decay of each note.

IV. RESULTS

The presented system is permanently being developed, but it is already possible to report on some of the key strengths of the proposed approach. Taking into consideration previous studies and proposals, many of them are still quite limited to a direct and linear relation in what regards the mapping of gestures into sound and music. Having that in mind, the main contribution of the Zatlabs system is the inclusion of a Machine Learning module after the gesture analysis module. This inclusion allows the creation of intricate and semantically meaningful relations between the extracted low- and high-level gesture features and the output sound and music parameters, avoiding the limitations of a direct gesture-sound correspondence. The objective is to use this platform to continue to explore other Machine Learning based algorithms that, by means of learning of data extracted from human gestures, may result in novel and more elaborated ways of interaction between human gesture and algorithmic sound and music producing systems.

The skeleton representation and gesture analysis modules will be available as open source software. There's already some video footage of experimental performances using the system at <http://andreabaltazar.wordpress.com/artech-2012/>.

V. CONCLUSION

This paper describes a modular system, named Zatlabs, that allows to capture and analyze human gestures in a very low intrusive manner, using a Microsoft Kinect video capture system and a custom application for video feature extraction and analysis developed using openFrameworks. The extracted gesture features are subsequently interpreted in a Machine Learning environment that continuously modifies several input

parameters in a computer music algorithm. Through this kind of gesture analysis one can perceive an higher level of interaction, where instead of a simple direct mapping of human movement to sound, the performer is able to interact in a more semantically and artistically meaningful dialog.

As future work, one should consider the development of gesture classifier algorithms to make possible for even more accurate and creative gesture to music relations. For the music generation there is also more development to be made on the ChucK programming language, in order to have an even more interesting music piece.

REFERENCES

- [1] P. Fraisse, "Rhythm and Tempo," in *The Psychology of Music*, ser. Springer Handbook of Auditory Research, D. Deutsch, Ed. Academic Press, 1982, pp. 149–180. [Online]. Available: <http://www.springerlink.com/index/10.1007/978-1-4419-6114-3>
- [2] R. I. Godøy and M. Leman, *Musical Gestures: Sound, Movement, and Meaning*, R. I. Godøy and M. Leman, Eds. Routledge, 2009. [Online]. Available: <http://www.amazon.jp/dp/0415998875>
- [3] M. Yoo, J. Beak, and I. Lee, "Creating Musical Expression using Kinect," *visualcomputing.yonsei.ac.kr*, no. June, pp. 324–325, 2011. [Online]. Available: <http://visualcomputing.yonsei.ac.kr/papers/2011/nime2011.pdf>
- [4] R. Fiebrink, D. Trueman, and P. Cook, "A metainstrument for interactive, on-the-fly machine learning," in *Proc. NIME*, vol. 2, 2009, p. 3.
- [5] G. Wang, P. Cook, and Others, "ChucK: A concurrent, on-the-fly audio programming language," in *Proceedings of International Computer Music Conference*, 2003, pp. 219–226. [Online]. Available: <http://nagasm.org/ASL/icmc2003/closed/CR1055.PDF>
- [6] A. Mulder, "Human movement tracking technology," *Hand The*, no. July, pp. 1–16, 1994.
- [7] C. Dobrian and F. Bevilacqua, "Gestural control of music: using the vicon 8 motion capture system," in *Proceedings of the 2003 conference on New interfaces for musical expression*. National University of Singapore, 2003, pp. 161–163. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1085753>
- [8] M. Feldmeier, M. Malinowski, and J. A. Paradiso, "Large group musical interaction using disposable wireless motion sensors," *Computer*, pp. 83–87, 2002. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.7463>
- [9] U. Enke, "DanSense: Rhythmic Analysis of Dance Movements Using Acceleration-Onset Times," Ph.D. dissertation, RWTH Aachen University, 2006. [Online]. Available: <http://hci.rwth-aachen.de/materials/publications/enke2006.pdf>
- [10] A. Camurri, C. L. Krumhansl, B. Mazarino, and G. Volpe, "An Exploratory Study of Anticipating Human Movement in Dance," *Stimulus*, no. i, pp. 2–5, 2004.
- [11] C. Guedes, "Mapping Movement to Musical Rhythm: A Study in Interactive Dance," Ph.D. dissertation, New York University, 2005.
- [12] L. Naveda and M. Leman, "Representation of Samba dance gestures, using a multi-modal analysis approach," in *5th International Conference on Enactive Interfaces*. Edizione ETS, 2008, pp. 68–74. [Online]. Available: <http://hdl.handle.net/1854/LU-503783>
- [13] P. Polotti and M. Goïna, "EGGS in Action," in *NIME 2011 Proceedings*, 2011, pp. 64–67.
- [14] M. A. Bokowiec, "V ! OCT (Ritual): An Interactive Vocal Work for Bodycoder System and 8 Channel Spatialization," in *NIME 2011 Proceedings*, 2011, pp. 40–43.
- [15] M. Wright and A. Freed, "Open sound control: A new protocol for communicating with sound synthesizers," in *Proceedings of the 1997 International Computer Music Conference*. International Computer Music Association San Francisco, 1997, pp. 101–104. [Online]. Available: <http://en.scientificcommons.org/26532287>
- [16] A. Baltazar, C. Guedes, F. Gouyon, and B. Pennycook, "A Real-time Human Body Skeletonization Algorithm for MAX/MSP/JITTER," in *ICMC 2010*, 2010.