Interactive Music: Human Motion Initiated Music Generation Using Skeletal Tracking By Kinect

Tamara Berg, Debaleena Chattopadhyay, Margaret Schedel, Timothy Vallier

Abstract

This work experiments with human motion initiated music generation. Here we present a stand-alone system to tag human motions readily into musical notes. We do this by first discovering the human skeleton using depth images acquired by infrared range sensors and then exploiting the resultant skeletal tracking. This real-time skeletal tracking is done using the videogame console Microsoft Kinect™ for Xbox 360. An agent's bodily motion is defined by the spatial and temporal arrangement of his skeletal framework over the episode of the associated move. After extracting the skeleton of a performing agent by interfacing the Kinect with an intermediate computer application, various features defining the agent's motion are computed. Features like velocity, acceleration and change in position of the agent's body parts is then used to generate musical notes. Finally, as a participating agent performs a set of movements in front of our system, the system generates musical notes that are continually regulated by the defined features describing his motion.

1 Introduction

A full grown adult body has 206 bones and over 230 moveable and semi-moveable joints. The maximum number of degrees of freedom that any joint can have is three. However, the effect of adjacent joints may be summated to express the total amount of freedom between one part of the body and an area more distant to it. The more distant a segment, the greater the degrees of freedom it will possess relative to the torso. Jones et al. [4] cites the example of the degrees of freedom between the distant fingers of the hand and the torso amounting to 17.

Now, with such a wide choice of poses and possibilities the human body is capable of numerous moves and movements. And, as it happens, human beings use their bodily movements more than often as a mode to interact. But interaction needs the participation of more than one agent. Hence since not long before, interactions utilizing human motion were restricted only to human-human interactions. However with the recent developments in technology, the field of Human Computer Interaction has been exploiting human motion as one of the multimodal interaction possibilities.

Human Computer Interaction applications exploiting gesture recognition, full body tracking and motion detection has become a commoner in today's everyday world. Among the recent advances is the launch of the videogame console Kinect ™ for Xbox 360 in the late Fall of 2010.

In this work of Interactive Music, we have used the technology for skeletal tracking available with the Kinect™ videogame console and developed an application to perform tagging of human moves and movements with music. To use the Kinect videogame console for Xbox 360, we had to first interface it with a computer. For that we have used the OpenNI ™ framework [9] and NITE Middleware from PrimeSense™ [8]. Further we have used the Open Sound Control (OSC) Library to bridge between the Motion Capture Application and the Music Generator Application. The Motion Capture Software performs the Skeletal Tracking using the Kinect, computes a set of features defining the human motion and passes these features as OSC messages to the Music Generator Application. The Music Generator Application which is build with MAX/MSP software then generates musical notes depending upon how the passed features changes over time. Thus, the music created from our system is interactive, real-time and de-

fines a performing agent's movements.

2 Background

2.1 Music & Motion

"Is there a true perceptual experience of movement when listening to music, or is it merely a metaphorical one owing to associations with physical or human motion?"

Honing [3] gives an informal yet informative description on how the apparent relation between motion and music has been investigated in a considerable number of works. This article reviews kinematic models that create explicit relation between motion and music which can be tested and validated on real performance data. The key component behind the symbiotic relationship between dance and music is a series of body movements or human motion. In the computer music literature and the sensor system literature, different systems are proposed from time to time [12] to record different context of motion to better understand this relation.

There are existing sensor systems that capture various forms of gestures using spatial mapping for building interactive surface like smart walls as proposed by Paradiso et al. [7] or dance floors for tracking dance steps as described by Griffith et al.[2]. Paradiso et al. [6] designed an arrangement of tilted accelerometers and pressure sensors at various positions to capture high-level podiatric gesture and proposes an interface for interactive dance. The goal of their work had been to capture a collection of action-to-sound rules for improvisational dancers. Lee et al. [5] proposed a system to extract rhythmic patterns from movement of a single limb using accelerometers in real-time. Wechsler et al. [11] introduces a camera-based motion sensing system that is essentially an interactive video environment which permits performers to use their movements to control or generate sounds. In our work, we propose an interactive system that uses the depth sensors of Kinect™ for a whole body skeletal tracking. It is able to automatically generate musical notes based on the changes in velocity, acceleration and position of a set of skeletal joints in a performing agents body.

Figure 1: The Kinect™ Game Console.

2.2 The Kinect™

The recent advances on imaging hardware and computer vision algorithms had led to the emerging technology of markerless motion capture using a camera system. The commercial solution for markerless motion capture currently available in the market is the Microsofts Kinect videogame console. The technology associated with the Kinect™ console discovers the 3D skeleton for a human body and gives us a robust tracking output [10]. The Kinect essentially uses a range camera technology developed by PrimeSense™ that interprets 3D scene information from a continuously-projected infrared structured light. The depth sensors in Kinect consist of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. After recording the 3D scene information, the Kinect first evaluates how well each pixel fits certain features for example, is the pixel at the top of the body, or at the bottom? This gives each pixel a certain score. The score for each feature is then combined with a randomized decision forest search. A randomized decision forest search is essentially a collection of decisions that asks whether a pixel with a particular set of features is likely to fit a particular body part. The Kinect technology has already been trained on a collection of motion capture data (around 500,000 frames). Once the body parts have been identified, the system then calculates the likely location of the joints within each one to build a 3D skeleton. The Microsoft Xbox runs this algorithm 200 times per second, which is around ten times faster than any previous body-recognition techniques ensur-

ing players can easily be tracked fast enough for their motions to be incorporated in to games.

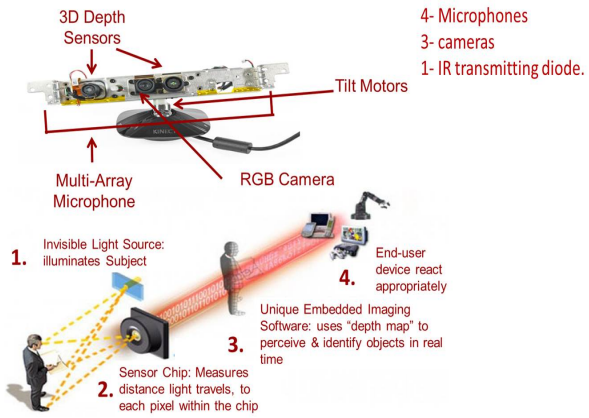

Figure 2: The Kinect™ Sensors in Play

3 Human Motion Capture System with Kinect™

The idea of tagging moves with musical notes originated from a preliminary discussion on how the sense of depth can be communicated through the change in musical notes. So, we believed that given a steady flow of information regarding the change of body parts in terms of relative position, velocity and acceleration, it shall be interesting to generate musical notes trying to express a performing agent's bodily movements. To enable a robust skeleton tracking we used markarless motion capture system of Kinect and communicated the features defining that motion as OSC messages to the Music Generator System. As mentioned before, to make use of the Kinect videogame console for Xbox 360, we interface it with a computer using the OpenNI ™ framework [9] and NITE Middleware from PrimeSense™ [8].

Now, with the Kinect™ console interfaced with the computer using proper interfaces, what was required was to make a bridge between the Kinect™ Console and Open Sound Control. This would enable us to actually use bodily movements (in real time) to generate musical signatures. So essentially we could tag certain moves and movements into musical notes. We built a system to make this possible using the OpenNI, the NITE Middleware, the Open Sound Control Library and the Open Frameworks Library. Using all this available frameworks, we built an Interactive Music system that can essentially permit human agents to interact with the application using their motion and create music seamlessly. This system uses the Kinect™ and a computer as its hardware components and hence is very portable and inexpensive to use.

We present all our software systems at [1]. We also present a work-flow of the final system that we have used to tag moves with music in Figure 3.

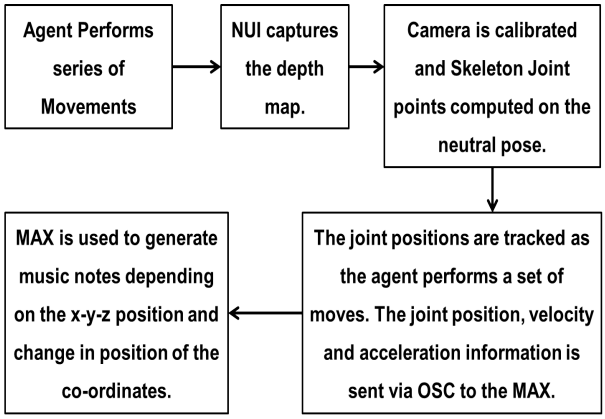

Figure 3: Motion To Music Working Flow Chart.

4 Music Generator System

When approaching the OSC Skeleton application we wanted a solution that was accessible and interesting. The goal was that each joint should have its own customizable sound source and that the performer and audience should easily be able to discern the sound changing and have a general idea of which sounds are coming from which joints. The entry point of this project is an application called Max/MSP or Max for short. Max is a visual object oriented programming language which has three cores. The first core is the Max core which handles mathematic functions. The second core is MSP which is used for signal processing to generate sound and manipulate existing sound. The thirds core is Jitter which is used for video processing. All of the cores are fully accessible from application which makes Max a very powerful multimedia visual language. The software OSC Skeleton [1] sends Open Sound Control or "OSC" data packets through the local network. OSC is an ideal method of passing data because unlike MIDI, it can be passed very easily over the local network connection. The first step in building the Max patch receiver for OSC Skeleton is the

Joint name
User Id
”confidence of the joint position co-ordinate”
/xjoint
/yjoint
/zjoint
9 values of the joint orientation matrix.
”confidence of the joint orientation”

Table 1: Skeletal Joint Information as sent over OSC

unpacking process. OSC Skeleton sends data in a particular way. Information for all joints sent from the Kinect to the OSC is as shown in Table 1.

The first function seen in Figure 4 tells the program to receive all UDP data on port 3333 and route everything under the joint heading along the path of j which stands for joint.

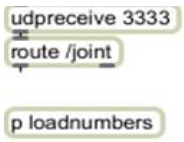

Figure 4: Joint Information Routing Building Block of Motion-to-Music Appication

Next, Figure 5 shows a visual aide that is constructed to assist in organizing the unpacking of each of the 15 joints. This hand drawn stick figure helps to better visualize how the Kinect is tracking the agent, and where the joints are located on the body. Each one of the boxes seen in Figure 6 receives the *joint* data and unpacks it in a *sub-patch*, which allows users to create programs or *patches* inside of an existing patch. The sub-patches that unpack the *joint* data look like what is in Figure 6. As we can see, the only data being unpacked for this project is the position of the X, Y, and Z coordinates of the joints, and not the orientations. These values are then packed into the range of 0 and 127 which is the standard range for MIDI. This is done for simplification purpose and to allow better interaction with components inside of Max and also for quick redirecting of data to programs outside of Max.

The last part of the Figure 6 sends the three values (X,Y,Z) to another sub-patch, seen in Figure 7. This sub patch receives the XYZ data and filters it through switches, which can be globally and locally activated and deactivated. This allows one to easily turn on or off and join with one click. Next, the XYZ values are visualized to give the composer and performer feedback.

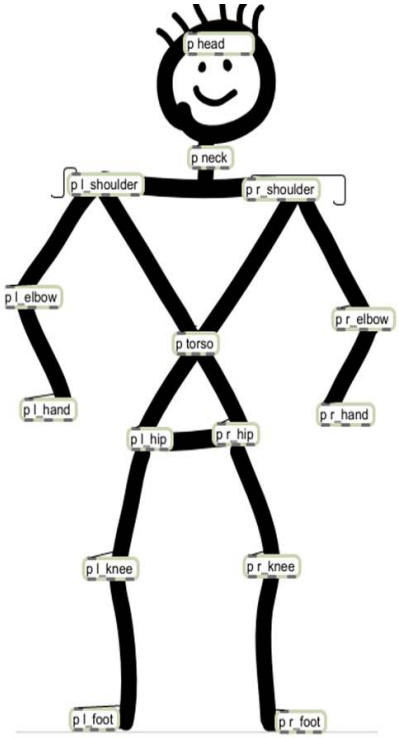

Figure 5: Visual Aide of Motion-to-Music Appication

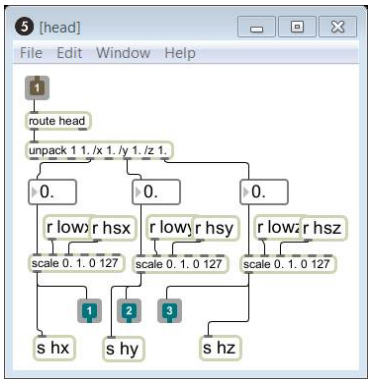

Figure 6: Motion To Music Application Sub-patch.

One of the visual panels as shown in Figure 8 is created and labeled for each of the 15 joints. The blue box in the top left is a toggle switch. When the box is empty, the joint is inactive. When one clicks the box, the joint becomes active. The final part of the patch is passing the data from the sliders to another sub patch which takes the values and

generates sound. The type of sound being generated is called *Frequency Modulation* which takes a carrier frequency, modulator frequency, and amplitude to generate a complex waveform.

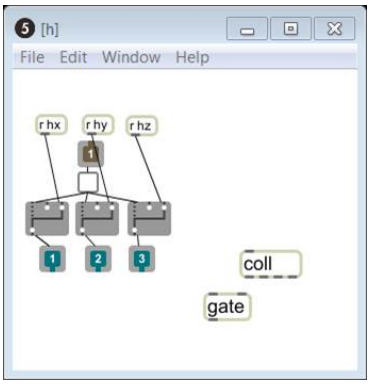

Figure 7: Another Motion To Music Application Sub-patch.

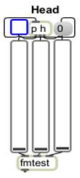

Figure 8: Visual Panels for each Joints

As shown in Figure 9, the three values of X-Y-Z are assigned respectively to carrier frequency, modulator frequency, and amplitude. Each joint has a dedicated frequency modulation sound generator allowing them to act as unique instruments. After the sound is generated, it is passed to two sliders which act as stereo volume control. Also, all of the scaling for the incoming XYZ values and respective carrier frequency, modulator frequency, and amplitude can be scaled easily in the final patch along clearly labeled along the side walls. The final patch (with the sub patches hidden) looks like in Figure 10.

5 Results

To check out a demo of this application and generate musical notes as you perform a set of movements using the Kinect™ and our systems, please visit our project website [1].

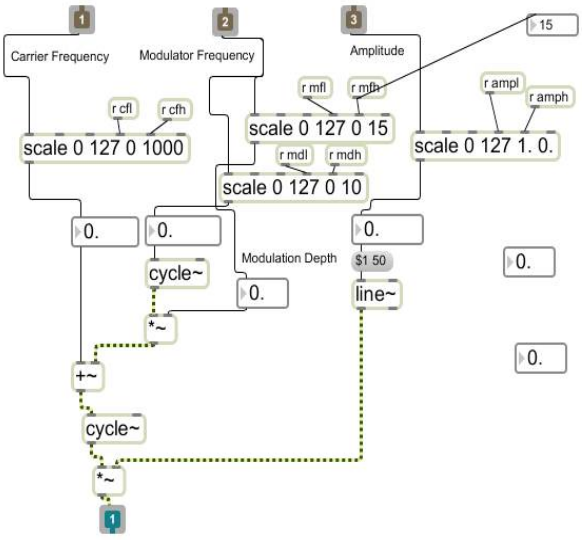

Figure 9: Frquency Modulation

References

[1] D. Chattopadhyay, T. Vallier, T. Berg, and M. Schedel. Multimodal tagging of human motion using skeletal tracking with kinect™. November 2011.

[2] Fernstrm M. Griffith, N. Litefoot: A floor space for recording dance and controlling media. In *Proceedings of the 1998 International Computer Music Conference, International Computer Music Association*, pages 475–481, 1998.

[3] Henkjan Honing. Computational modeling of music cognition: A case study on model selection. *Music Perception: An Interdisciplinary Journal*, 23(5):pp. 365–376.

[4] K. Jones, K. Jones, and K. Barker. *Human movement explained*. Physiotherapy practice explained.

[5] Eric Lee, Urs Enke, Jan Borchers, and Leo de Jong. Towards rhythmic analysis of human motion using acceleration-onset times. In *Proceedings of the 7th international conference on New interfaces for musical expression*, NIME '07, pages 136–141, 2007.

[6] J. A. Paradiso, K. Hsiao, A. Y. Benbasat, and Z. Teegarden. Design and implementation of expressive footwear. *IBM Systems Journal*, 39(3.4):511 –529, 2000.

[7] J. A. Paradiso, K. Hsiao, J. Strickon, J. Lifton, and A. Adler. Sensor systems for interactive surfaces. *IBM Syst. J.*, 39:892–914, July 2000.

[8] PrimeSense. NITE™. November 2010.

[9] PrimeSense. OpenNI™. November 2010.

[10] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. June.

[11] F. Weiss Wechsler, R. and P. Dowling. Eyecon: A motion sensing tool for creating interactive dance, music, and video projections. In *Proceedings of the AISB 2004 COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems*, AISB '04, pages 74–79, 2004.

[12] Todd Winkler. Making motion musical: Gesture mapping strategies for interactive computer music. *Computer*, page 261264, 1995.

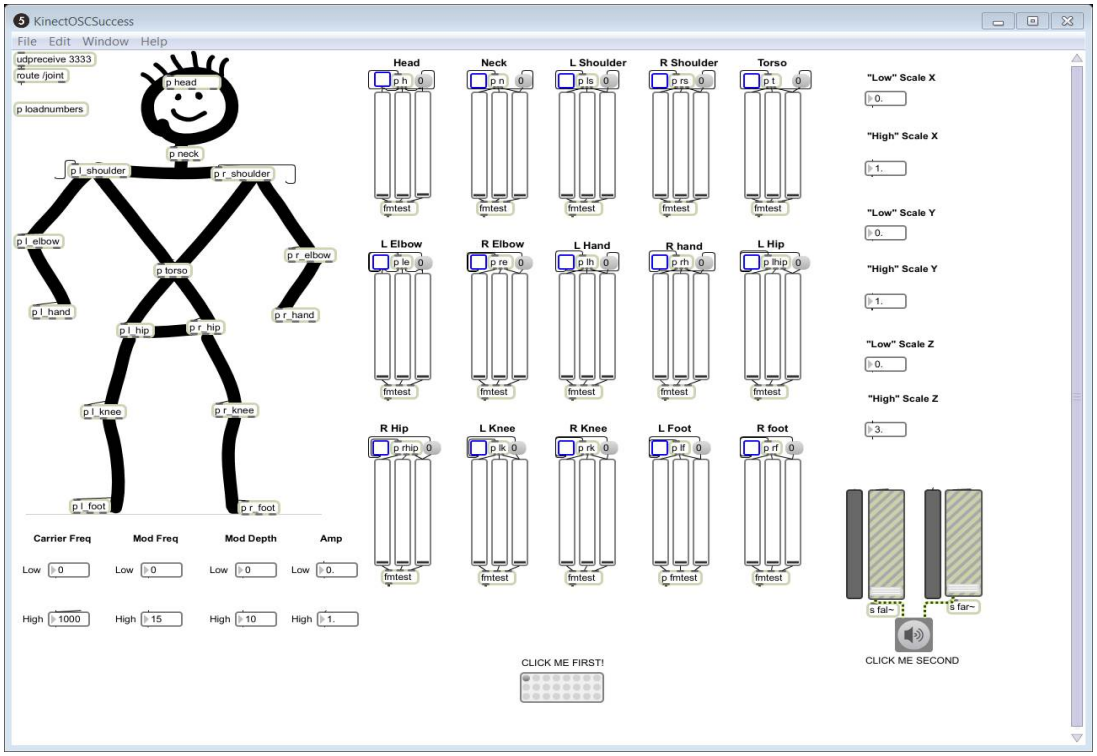

Figure 10: Final patch for the MAX application.