Free-Space Gesture Mappings for Music and Sound

by

Gabrielle Odowichuk
BEng, University of Victoria, 2009

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Applied Science

in the Department of Electrical and Computer Engineering

Free-Space Gesture Mappings for Music and Sound

by

Gabrielle Odowichuk
BEng, University of Victoria, 2009

Supervisory Committee

Dr. P. Driessen, Co-Supervisor (Department of Electrical and Computer Engineering)

Dr. G. Tzanetakis, Co-Supervisor (Department of Computer Science)

Dr. Wyatt Page, Member (Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. P. Driessen, Co-Supervisor (Department of Electrical and Computer Engineering)

Dr. G. Tzanetakis, Co-Supervisor (Department of Computer Science)

Dr. Wyatt Page, Member (Department of Electrical and Computer Engineering)

Abstract

This thesis describes a set of software applications for real-time gesturally controlled interactions with music and sound. The applications for each system are varied but related, addressing unsolved problems in the field of audio and music technology. The three systems presented in this work capture 3D human motion with spatial sensors and map position data from the sensors onto sonic parameters. Two different spatial sensors are used interchangeably to perform motion capture: the radiodrum and the Xbox Kinect. The first two systems are aimed at creating immersive virtually-augmented environments. The first application uses human gesture to move sounds spatially in a 3D surround sound by physically modelling the movement of sound in a space. The second application is a gesturally controlled self-organized music browser in which songs are clustered based on auditory similarity. The third application is specifically aimed at extending musical performance through the development of a digitally augmented vibraphone. Each of these applications is presented with related work, theoretical and technical details for implementation, and discussions of future work.

Table of Contents

Supervisory Committee **ii**

Abstract **iii**

Table of Contents **iv**

List of Figures **vi**

Acknowledgements **viii**

1 Introduction **1**

1.1 Problem Formulation 2

1.2 Thesis Structure 4

2 Background And Motivation **6**

2.1 Contextualizing a Gesture 7

2.2 Data Mapping 8

2.3 Free-space Gesture Controllers 10

2.4 A Case Study 13

3 Capturing Motion **16**

3.1 Spatial Sensor Comparison 17

3.2 Latency 19

3.3 Range 20

3.4 Software Tools 22
3.5 Future Work with Motion Capture 24

4 Motion-controlled Spatialization **27**
4.1 Related Work 28
4.2 Sound Localization 29
4.3 Creating a Spatial Model 30
4.4 Implementation 31
4.5 Summary and Future Work 37

5 Gesturally-controlled Music Browsing **38**
5.1 Related Work 39
5.2 Organizing Music in a 3D space 40
5.3 Navigating through the collection 44
5.4 Implementation 45
5.5 Summary and Future Work 47

6 Hyper-Vibraphone **48**
6.1 Related Work 50
6.2 Gestural Range (Magic Eyes) 51
6.3 Adaptive Control (Fantom Faders) 54
6.4 Summary and Future Work 57

7 Conclusions **59**
7.1 Recommendations for Future Work 60

Bibliography **62**

List of Figures

2.1 Interactions between Sound and Motion 6
2.2 Data Mapping from a Gesture to Sound 9
2.3 Mickey Mouse, controlling a cartoon world with his movements in Fantasia 10
2.4 Leon Theremin playing the Theremin 11
2.5 Radiodrum design diagram 12
2.6 Still shots from MISTIC concert 15

3.1 Sensor Fusion Experiment Hardware Diagram 17
3.2 Sensor Fusion Experiment Software Diagram 18
3.3 Demonstration of Latency for the Radiodrum and Kinect 19
3.4 Captured Motion of Four Drum Strikes 21
3.5 Radiodrum Viewable Area 21
3.6 Kinect Viewable Area 22
3.7 Horizontal Range of both controllers 23

4.1 Room within a room model 31
4.2 Implementation Flow Chart 32
4.3 Delay Line Implementation 33
4.4 Image Source Model 35
4.5 OpenGL Screenshot 36

5.1 A 3D self organizing map before (a) and after (b) training with an 8-color dataset 42
5.2 3D SOM with two genres and user-controlled cursor 44
5.3 Implementation Diagram 46

6.1 Music Control Design 52
6.2 Audio Signal Chain 53
6.3 Virtual Vibraphone Faders 54
6.4 Computer Vision Diagram 55
6.5 Virtual recreation of the vibraphone 56

Acknowledgements

I'd like to begin by thanking my co-supervisors, Dr. George Tzanetakis and Dr. Peter Driessen, for their support, patience, and many teachings through my undergraduate and graduate studies at UVic. Peter's enthusiasm for my potential and my future has given me motivation and confidence, especially combined with the respect I have for his incredible knowledge and experience. Whenever I asked George if he was finally getting sick of me, he would assure me that could never happen. I'm still not sure how that's possible after all this time, but what a relief, and I will always strive to one day be as totally awesome in every way as George.

My first encounter with this field of research and much of my early enthusiasm came from sitting in the classroom of Dr. Andy Schloss. His dry sense of humour and passion for the material is what got me into this world. Thanks also to Kirk McNally, for helping me set up the speaker cube and teaching me some crucial skills with audio equipment, and to Dr. Wyatt Page for his help with my thesis and for showing me what an amazing academic presentation looks like.

Early on in my master's program, Steven Ness welcomed me into our research lab, and has helped me understand how to be an effective researcher. Many other friends and colleagues have helped me a long the way: Tiago Tiavares, Sonmez Zehtabi, Alex Lerch, and Scott Miller were all of particular importance to me.

A large chapter of this thesis is about a collaboration with Shawn Trail, who is a dear friend and the inspiration for what is, in my mind, the research with the most possible impact down the road. The use of this type of gestural control, when completely into music practice, has expressive possibilities that are still very much untapped. Thanks also to David Parfit for collaborating with me in the Trimpin

concert, which gave me more context and empirical proof that this type of control is rich with expressive possibilities.

Paul Reimer is a close friend and my indispensable coding consultant. If I found myself spending more than a few hours beating my head against a wall with a technical issue, I need only ask Paul for help and my problem would soon be solved. Marlene Stewart has been another source of much support. It's so rare to have people in your life you can rely on so completely like Paul and Marlene.

Thanks mom 'n dad for being the proud supportive parents that raised the kind of daughter who goes and gets a master's degree in engineering.

And finally thank you to NSERC and SSHRC for supplying the funding for this research.

Chapter 1

Introduction

The ability for sound and human gestures to affect one another is a fascinating and useful notion, often associated with artistic expression. For example, a pianist will make gestures and motions that affect the sounds produced by the piano, and also some that do not. Both types of gesture are important to the full experience of the performance. A dancer, though not directly changing the music, is also creating a expressive representation of the music, or the ideas and emotions evoked by the music. In this case, sound affects motion. The connection between auditory and visual senses is a large part of what makes audio-visual performances interesting to watch and listen to.

Advances in personal computing and the adoption of new technologies allow the creation of new and novel mappings between visual and auditory information. A large motivator for this research is the growing capabilities of personal computing. The mapping of free-space human gesture to sound used to be a strictly off-line operation. A collection of computers and sensors were used to capture motion, and then calculations to produce a corresponding auditory output were synthesized and played back afterwards. Modern computers are able to sense motion and gesture and react almost instantaneously.

The ability to capture free-space motion, perform complex calculations, and pro-

duce corresponding audio in real-time is a fundamental requirement for the implementation of these systems. This type of control requires thought into how to use this control in many contexts. The secondary feedback of audio playback is an important aspect of what makes gesture-controlled sound and music useful, because accessing or manipulating aural information by listening to auditory feedback of that information is intuitive and natural.

Though there are many types of gestures used in human-computer interaction (HCI), in particular this work focuses on three dimensional motion capture of large, relatively slow, continuous human motions. The purpose of this work is not to classify these motions and recognize gestures to trigger events. Instead, the focus is on the mapping continuous human motion onto sonic parameters in three new ways that are both intuitive and useful in the music and audio industry.

1.1 Problem Formulation

The possible applications of these gesturally controlled audio systems span several different facets of HCI, and address a variety of music and audio-industry related problems, such as:

- Intuitive control in 2D and 3D control scenarios

  Intuitive and ergonomic control are an important consideration in the field of HCI. The use of 3D gesture-based sensors to interact with computers is a point of much research, with large companies like Microsoft and Apple investing heavily in the development of new free-space gesture sensors [23]. The traditional keyboard-mouse control is being challenged by controls capable of sensing higher dimensional data. The development of new sensors meant specifically for gesture-based human computer interaction has fuelled the invention of new ways to interact with aural information.

- Immersion in virtual-reality and augmented-reality based environments

Immersion in a virtual environment is something the video-game and movie industries are constantly striving for. Ideally, scientists and researchers dream of a virtual space that is indistinguishable from reality, in which those within the space are completely absorbed. Surround sound is a perfect example of efforts towards a realistic recreation of an auditory space. While enjoying an action movie in theatres, if an explosion happens to the left of the audience, and suddenly gun shots from behind, the overall experience is heightened. Spatialization of sounds, or the virtual placement of sounds in space, is an important aspect of an immersive auditory experience.

- Effectively and easily accessing aural media

  Effectively accessing information is another consideration of HCI. The growing amount of information available to computer users at increasingly quick rates has created a demand for novel methods of browsing data. While finding a specific piece of music is easy when the title or artist is known, browsing through new music or world music can be far more difficult. This can also be applied to a collection of sounds that do not necessarily have associated text. For example, say you are choosing sound effects for a movie and you need to pick the sound of a car revving its engine from a collection of hundreds of recordings of cars revving their engines. The text associated with these recordings are much less useful than the information found in the recording itself.

- Connecting a musical performer's intentions for expression and the resulting sounds

  The perceived expressiveness of a performance is tied to the perceived coupling between a performer's gestures and the resulting sounds. By

extending a musical performance with gesture-controlled augmentations, a new means of artistic expression is created. Since the captured gestures can be mapped to sound in many ways, the possibilities for expressive control are extensive, and can be expanded to suit certain instruments specifically.

1.2 Thesis Structure

This thesis presents three new systems for controlling sound with free-space human gestures. Chapter 2 will discuss the background, motivation, and previous work behind this project. In Chapter 3, a method is presented for expanding a 3D control paradigm previously developed on the *Radiodrum* (an electromagnetic capacitive 3D input device), using the *Kinect* (a motion sensing input device for the Xbox 360 video game console). The responsiveness and range of the sensors are compared to each other and to a fused data stream from both sensors.

In Chapter 4, gestural control is used to manipulate the perceived location of a sound source. Using a surround sound system with loudspeakers positioned in a 3D cube, captured human motions are mapped to the movement of a sounds in a virtual space. This control is intuitive, because the mapping is from one motion to another. While sound designers have previously moved sound sources with sliders and joysticks, capable of controlling one and two dimensions respectively, moving a sound in 3-dimensions is a perfect case for the use of 3D motion control.

Chapter 5 introduces a gesture-based content-aware music browser. This system places sounds virtually in a 3D space, organized automatically based on auditory similarities. Representations in 3D have the potential to convey more information but can be difficult to navigate using the traditional ways of providing input to a computer such as a keyboard and mouse. Utilizing sensors capable of sensing motion in 3-dimensions, we propose a new system for browsing music in augmented

reality. Expanding on concepts from the previous chapter, this augmented reality is heightened by placing the sound files spatially using concepts from Chapter 4.

The use of gestural control in music is particularly interesting because of the artistic and expressive possibilities. In chapter 6, a collaboration with percussionist and vibraphone player Shawn Trail is presented, in which the use of non-invasive gesture sensing is integrated into practices for musical performance with the vibraphone. Two specific digital augmentations were implemented. The performer's motions are first mapped to filter parameters that modify the sounds produced by the vibraphone, and another extension of this work used gesture sensing to turn each bar of the vibraphone into a virtual fader.

Each of these systems is described in detail, with theoretical background, instructions for implementation, demonstrations of working models, and recommendations for future work and evaluations.

Chapter 2

Background And Motivation

The relationship between sound and movement is intrinsic and dance is a perfect example of this. Enjoying music often involves some sort of dancing and motion, a sort of personal expression of the music. In the case of dancing, the transfer of information is one-directional, the sound affects human movement. Though dance is an obvious example of how humans express music through movement, a performer also reacts to sound through movements and these movements may or may not have some effect on the sounds produced.

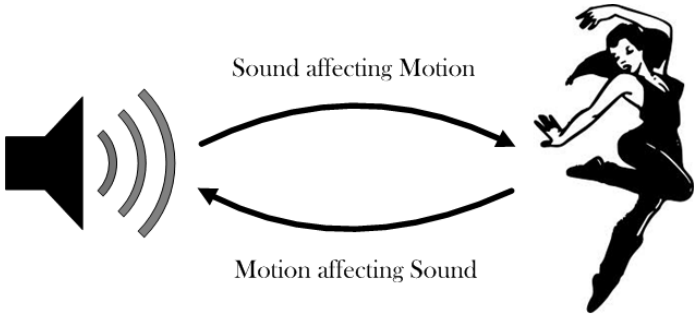

Figure 2.1: Interactions between Sound and Motion

What is of more interest here is when this process is opened up to include the reverse interaction. That is to say, the motion of the performer or dancer could affect and also be affected by the sounds. The result is a feedback loop with useful and expressive possibilities. The ergonomic advantages of gesture-based control are broad and span far beyond controlling sound and movement, but dance makes this

relationship so natural to us that using gestures to control sound and music is intuitive and effective in a variety of different scenarios.

2.1 Contextualizing a Gesture

"Gesture" is a loaded word with many definitions, even within the specific domain of music technology. Cadoz and Wanderley published work on this topic [16], discussing the different definitions of gesture within human-computer interaction and music. While the authors of this work admit that there is no single correct definition, for these purposes it's important to provide a context.

> *The Scribner-Bantam English Dictionary*, 1979 Bantan Books Inc.
>
> **gesture** [**ML** gestura posture, bearing] *n* **1** bodily movement expressing or emphasizing and idea or emotion; **2** act conveying intention. ... SYN *n* attitude, action, posture, gesticulation

This definition includes concepts that carry significant weight in a musical context, like movement and expression. A lot of different types of gestures fall into this definition, and a gesture can still mean many things. For example, the movement of the hand as it puts pen to paper and the transfer of information that is the primary goal of the written word. So, writing can be considered a gestural act, and the act of writing is also being used to control sound /cite.

Movement is an important aspect of gestures in music, as is the difference between posture and gesture. A posture is a single stationary position, while a gesture is a dynamic movement between postures. Although a posture can convey information, like how a stationary sign-language posture can convey a specific letter or word, combinations of postures and movements are required to convey more complex ideas.

The type of gesture used in this work is intended for control, and can therefore also be described as part of the semiotic function of the gestural channel [1], which encompasses most free-handed or empty-handed gestures. Other functions of the gestural channel require interactions with an instrument, and the semiotic function is unique in that it is not instrumental.

From the perspective of the senses involved, the lack of contact with a physical object removes the haptic feedback available with instrumental gestures. Most musicians can use their sense of feeling as an additional source of information that helps to properly control their instrument. If desired, this type of feedback can still be added to free-space systems through the use of wearable haptics [23]. The gestures used for control in this work are continuous, so gesture and sound co-exist. When haptics are not present, the auditory system, which temporally is a secondary form of feedback, becomes even more crucial to proper control.

In musical contexts, gestures can be intentional and the performer is consciously choosing to perform a gesture. A gesture can also be completely unintentional. After that, regardless of intention, the gesture may or may not result in any sound cues. While the mappings between motion and sound presented in this thesis are mostly intentional gestures that cause sounds, it is important to understand that this is not always the case. It is safe to say performers generally want to minimize unintentional gestures that cause sound, as they may lead to unwanted sonic events.

2.2 Data Mapping

Data mapping is the process of connecting the elements of two distinct models. In this context, we are taking data from sensors that capture data, and creating a corresponding sonic event [28]. Mappings can be very simple or very complex, and deciding what mappings are most effective is somewhat arbitrary. When mappings

[1] The *semiotic* function is used in this context to classify gestures with an intended communication of information [16]

are explicit and deliberate, it can also be seen as an algorithmic composition [29]. Choosing how to map these parameters has a lot to do with human perception, and specifically perception of sound. These parameters may have to do with physical properties of the sound, signal properties, psychoacoustic properties, or extracted meta-data [12].

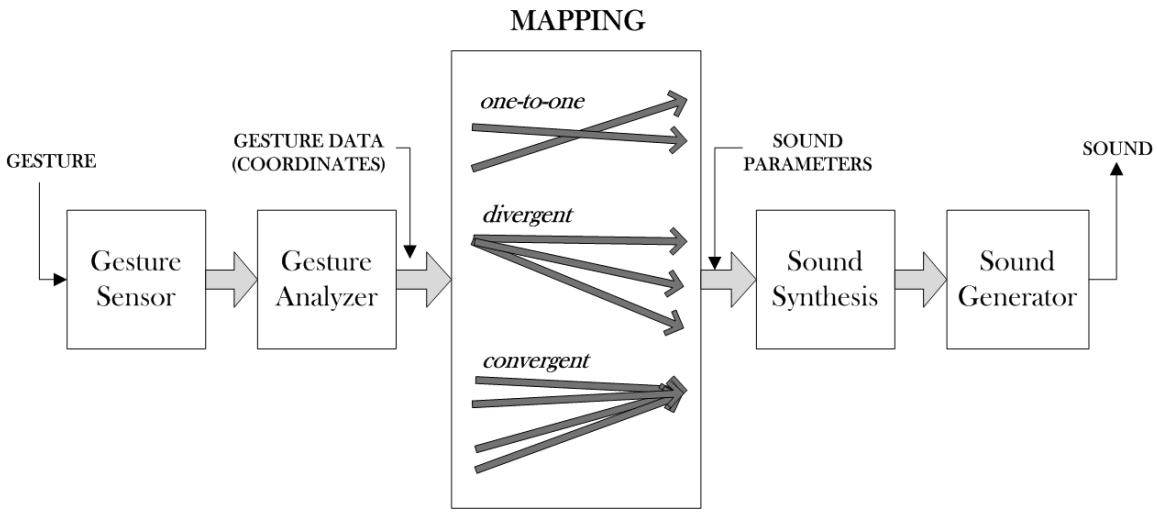

Figure 2.2: Data Mapping from a Gesture to Sound

Gesture mapping strategies have been broken up into three groups [53] based on the number of gestures and parameters that are mapped together. In the first case, *One-to-One* mappings, a single captured gesture affects a single musical parameter. In the second case, *Divergent* mappings refer to a single captured gesture affecting multiple musical parameters. And in the third case, *Convergent* mappings refer to many captured gestures control one parameter. The notion and effectiveness of one-to-many and many-to-one is also shown in [28], a study of the effectiveness of real-time musical control.

An example of a more complex mapping is presented in [20], where explicit mappings are not required and adaptable neural networks are used to map between gestural and musical data. This type of mapping is fuelled from the metaphor of conducting, and attempt to turn gestural data from hand motions into acoustic signals

[36].

Some mappings involved recognition and higher level computer learning algorithms. Many natural gesture mappings have been suggested in previous works, like "drag-and-drop" and "catch-and-throw" for example [68].

2.3 Free-space Gesture Controllers

The interest in controlling sound with motion came early, as stated earlier, with traditions like dance. In fact, the same technologies used here to map motion to sound have been used to analyze dance as a musical gesture [17], [18].

Figure 2.3: Mickey Mouse, controlling a cartoon world with his movements in Fantasia

Another obvious metaphor for this type of control is conducting. One person stands in front of dozens of instruments, and controls the speed and progression of the music with movements. Of course, the conductor doesn't have anywhere near complete control over the sounds produced, and really the members of the symphony have control to play whatever they want. Using technology, a more tightly coupled form of auditory control can be obtained.

Music and gestures are similar in that they are both expressive mediums that do not require the use of language [41]. The development of free-space controllers that do not physically restrict motion in any way have allowed for the creation of a new set of virtual control surfaces [47].

2.3.1 Theremin

The Theremin is the first gestural based sound synthesizer. Its invention has had a major impact on the evolution of computer music and development of modern day electric instruments.

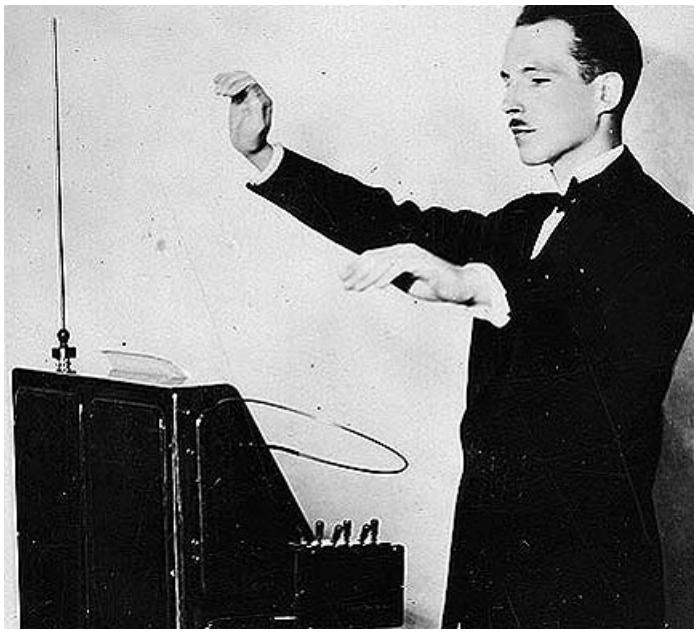

Figure 2.4: Leon Theremin playing the Theremin

The theremin was the first "hands-free" musical instrument [59]. The movement of the performer's hands through the space near the metal disrupts the electromagnetic field, acting as grounding plates to two variable capacitors. One hand affects the frequency of an analog waveform output, and the other controls the volume.

The impact of the theremin is still of interest to researchers today. The basic principles of the theremin have been used in the creation of new virtual instruments that incorporate modern gesture sensors [25]. There has also recently been projects to train a robot to play a theremin [45], demonstrating the need for the secondary-feedback of the oscillators to play the instrument.

2.3.2 The Radiodrum

The radiodrum is a gestural control system created at Bell Laboratories in the late 1980s [44]. It was originally meant to be a replacement for the mouse as a control for computers, giving the users an added dimension of control. Instead, the radiodrum is now mostly used as a musical instrument, played in live concert settings.

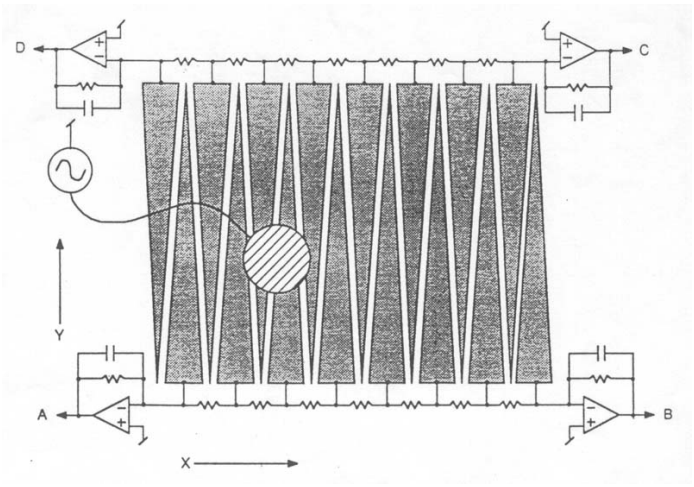

Figure 2.5: Radiodrum design diagram

This instrument has two sticks, each with a metallic coil at the tip driven by an electric RF signal. The x,y and z position of these sticks are determined by the point of greatest capacitance on the surface below. From the user's perspective, the resulting tool reports the position of two drum sticks in 3D space. Recent improvements to the radiodrum have increased the accuracy of the xyz position data thus enabling a more refined gestural control. One of these improvements was developed by [49], creating a new version of the radiodrum with the x,y,z, and dz data for each drumstick as analog waveforms instead of MIDI. The temporal and quantitive limits of MIDI made this type of data undesirable for mappings to continuous parameters.

2.3.3 The Kinect

The Kinect is a video game controller released by Microsoft in 2010. The controller includes various sensors including a camera, microphone array, and an active infrared

sensor. The infrared sensor provides a major development in control due to its ability to perform 3D imaging and tracking [72].

A major factor in the popularity of the Kinect was the low price, standard USB connector, and open availability of software libraries for the sensor. This allowed computer users to access not only the hardware, but also the data streams provided the Kinect, as well as high-level access to the detection and tracking algorithms associated with this sensor. The PrimeSense [1] skeleton tracking algorithms provide relatively easy access to computer vision algorithms that allow us to detect human form and identify different users, as well as track human motion. The techniques used probably resemble work by [54] and [71], however the underlying algorithms are copyrighted and only the output is publicly available.

A recent and significant use of the Kinect is in the recreation of a dance notation in cartesian coordinates proposed by Joseph Schillinger in 1934 [55]. The main goal of Schillinger's work was cross-disciplinary, and his writings on deriving data from one art form and applying them to another is a good analogy for controlling sounds through the use of motions and gestures.

2.4 A Case Study

A major motivation for this work lies in the possibility to use these systems in musical performance. And so, while the main focus of this research has been on the development implementation of software systems, this case study describes a related composition that features the use of 3D gestures for musical control.

The Music Intelligence and Sound Technology Interdisciplinary Collective (MISTIC) at the University of Victoria is a group of engineers, computer scientists and musicians dedicated to the research and development of music technologies.

This group presents concerts of new music bi-yearly, one of which coincided with another related event this year. Open Space, an art gallery in Victoria, BC, com-

missioned an interactive installation from sound sculptor Trimpin. This installation, entitled *4:33 + CanonX = 100*, was a celebration of the 100th birthday of famous composers John Cage and Conlon Nancarrow. The installation consisted of five modified pianos, given new life through the addition of motorized scrapers, files, hammers, ball bearings, among others. The sounds of the pianos were reminiscent of John Cage's prepared piano works, in which he modified pianos by attaching foreign objects to the piano strings to modify the instrument and create new timbres. The robotic nature of the installation and the ability to pre-program the pianos of each component reflected Nancarrow's compositions with player pianos. On April 28, 2012, MISTIC held a concert in which each performance was composed specifically for these robotic pianos.

Gesturally-controlled music requires a human made motion and a sonic reaction to that motion. In a collaboration with composer David Parfit, the author of this thesis used a Kinect to track her movements, creating corresponding sounds from the pianos and projected shapes. The infra-red based Kinect sensor required no physical components to be attached to the musician, and movements were tracked in complete darkness.

The perceived connection between the performers intentions and the resulting sounds has a great effect on the audience's reaction to a musical performance [56]. Within the computer music community, controlling music with the Kinect sensor is well-known and has been quickly adopted. However, other attendees of this concert were less familiar with this type of control. The reactions were from some were positive, and others were truly amazed. The questions received after were surprising and interesting. Even after being told specifically that the movements were modifying the music, some thought that the performer was simply dancing to the sounds. Nonetheless, the feedback from the audience and their general sense of wonder confirmed the possibilities for expressive and interactive control offered by this type of performance.

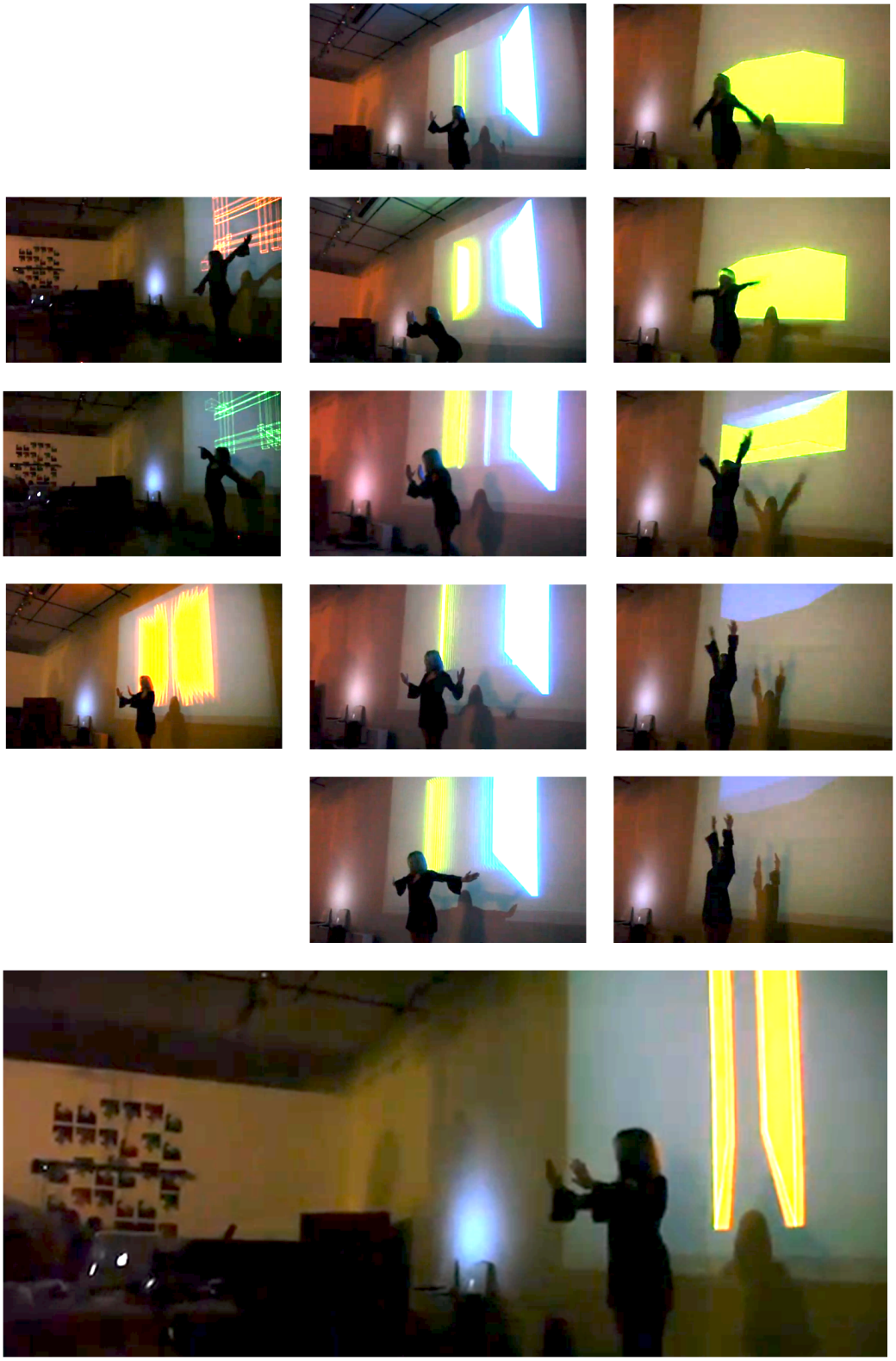

Figure 2.6: Still shots from MISTIC concert

Chapter 3

Capturing Motion

Several sensors have been used to capture motion and create resulting sounds, and of course the data from any sensor can be used to control and manipulate sound with many unique possibilities.

Motion sensors can be separated into two categories: *body sensors* which are attached to the body, and *spatial sensors* which detect the location in space relative to a specified projection grid [69]. *Body sensors*, such as accelerometers or gyroscopes, measure force and orientation. Position tracking can be accomplished with accelerometers and gyroscopes attached to the object of interest, and integration of the higher order motion information, however there is still a tendency for this position data to drift.

A focus of this work has been *spatial sensors* that have the ability to sense position. This three dimensional positional control in especially interesting, because it allows for direct immersion into a virtual world of sound. The MISTIC research lab has a long history of work related to the radiodrum, a musical controller that senses the positions of the tips of two sticks in three-dimensional space. The similarities between the radiodrum and the Kinect lead to quick adoption of this new technology for this type of gestural mapping, and the result is a set of mapping paradigms in which the two sensors can be used interchangeably.

3.1 Spatial Sensor Comparison

In many ways, the Radiodrum and Kinect are similar controllers. However, there are some major differences between these two pieces of technology. In this experiment, we present some early experiments that aim to demonstrate some of the major differences between these sensors.

Capacities of the human motor system regulates the type of movement that is possible to capture. The Kinect aims to capture body movements, and there are kinetic limitations to how quickly we can move our limbs. The Radiodrum aims to capture the tip of drum sticks, which are an extension of the human body and can be moved and adjusted with much greater speed.

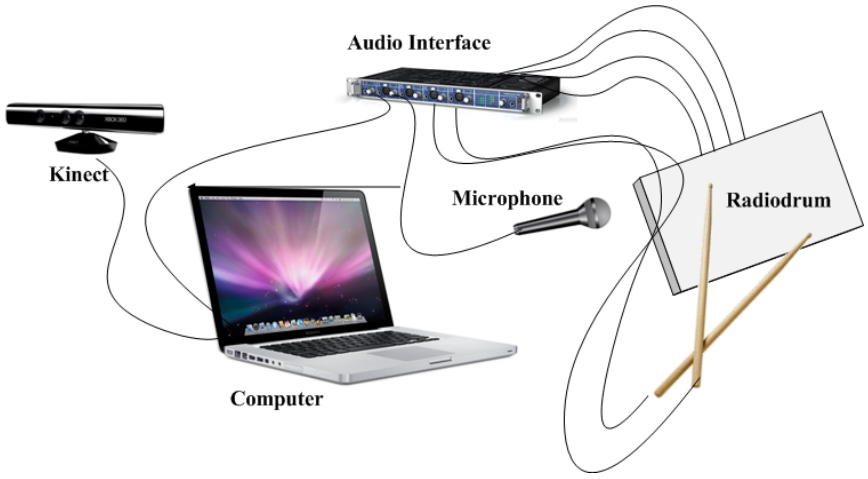

Figure 3.1: Sensor Fusion Experiment Hardware Diagram

Figure 3.1 shows the basic layout of the hardware. The Kinect connects to the computer via USB, and the Radiodrum via firewire through an audio interface. A microphone is also connected to the audio interface, which will be used as a reference when comparing the reaction of the sensors, much like an experiment performed by Wright et al [70].

Custom software was developed to record and compare data received from both sensors. The program flow is shown in Figure 3.2. A software program was written

that takes the motion tracking data from both the Radiodrum and the Kinect, as well as data from the audio interface, and saves all the data to a file.

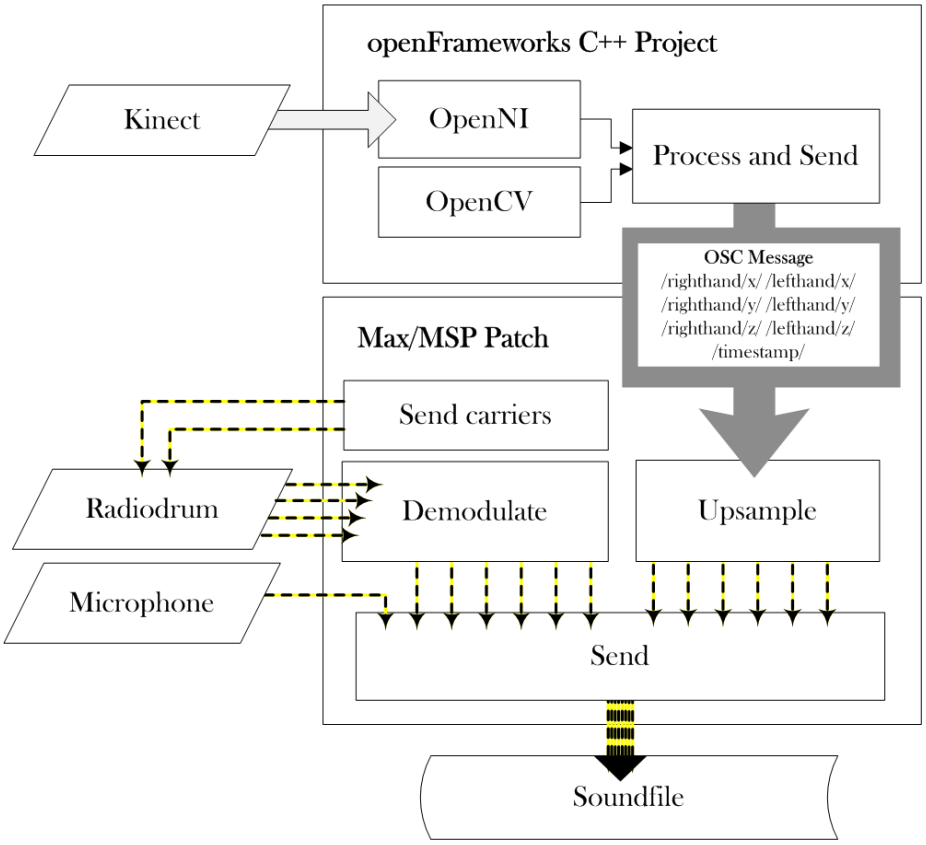

Figure 3.2: Sensor Fusion Experiment Software Diagram

Various movements were captured in an attempt to demonstrate some of the observed effects we have come across when using these sensors in a musical context. This work is not an attempt to show whether one device was superior to the other. Instead, we are more interested in comparing the accuracy and latency of the sensors so that data can be more intelligently fused for better control over the instrument. Fusing these data streams could produce interesting results, and there is some research fusing Kinect data with body sensors [62].

3.2 Latency

Humans can trigger transient events at a relatively high speed. This is demonstrated in the percussive technique known as the flam, where trained musicians can play this gesture with a 1ms temporal precision [67]. It is also important to look at the temporal accuracy of events. Delays experienced by the performer will change the perceived responsiveness of the musical instrument, a major consideration for musicians.

A basic difference between these two sensors is the vast difference in the update rate of the captured data. The radiodrum sends continuous signals to an audio interface, and the sampling rate of the data is determined by the audio interface. For this experiment, we used a frequency of 48000 Hz, but higher rates are possible. Conversely, the Kinect outputs position data at approximately 30Hz, the most stark difference in the capabilities of the sensors.

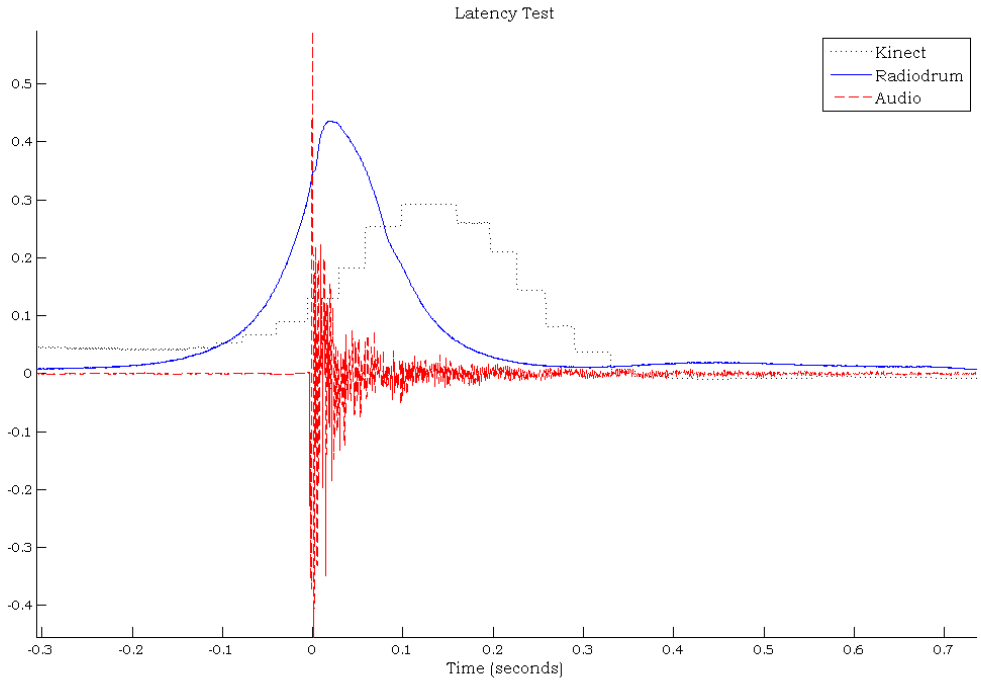

Figure 3.3: Demonstration of Latency for the Radiodrum and Kinect

We begin by demonstrating this latency by holding the tip of the radiodrum stick, and hitting the surface of a frame drum that has been placed on the surface of the radiodrum. We now have the output of three sensors to compare. The microphone has very little delay (between 50 and 100 milliseconds), and the auditory event of the frame drum being hit will occur before these events are seen by the gesture capturing devices. For our purposes, the audio response is considered a benchmark. The radiodrum will capture the position of the tips of each drumstick during this motion, and the Kinect will capture the position of the user's hands. We performed simple piece-wise constant up-sampling to the Kinect data, so that the difference in the frame rates is evident.

As seen in Figure 3.3, the Kinect displays a significant amount of latency. This latency occurs for a number of reasons, but there are two main temporal attributes that set the Kinect apart from the other two sensors. The first is the low frame rate of the sensor mentioned earlier, and the slow frame rate makes it nearly impossible to detect sudden events like the whack of a mallet. The second reason for this delay is the software-driven human skeleton tracking algorithms. With this type of human motion tracking, there is a trade-off between temporal accuracy and the accuracy of the spatial predictions.

Although rapid movement will not be detected by the Kinect, capturing slower movements is still possible. The following plot shows four discrete hits of the drum. Although the Kinect would not be able to use this information to produce a responsive sound immediately, we could still perform beat detection to determine the tempo a performer is playing at.

3.3 Range

The choice of mapping for gestures onto or into audio data has also been a source of significant attention. How much perceived change in sound should a movement

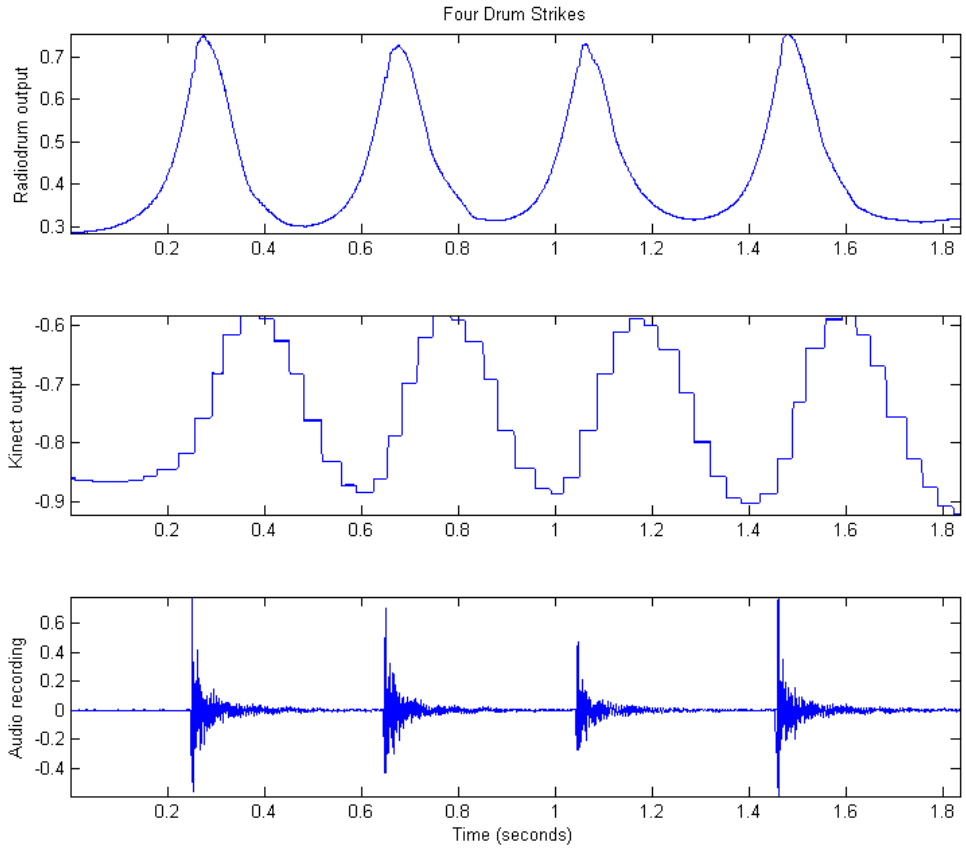

Figure 3.4: Captured Motion of Four Drum Strikes

produce?

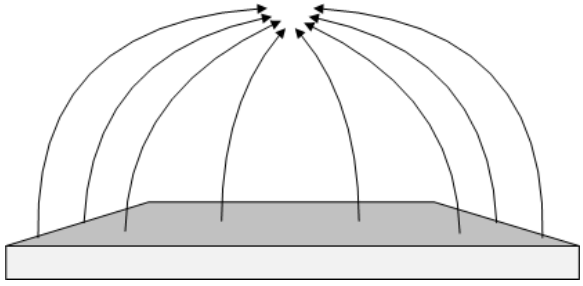

Figure 3.5: Radiodrum Viewable Area

First, we examine the range of motion for both sensors. The radiodrum will only return consistent position data while the drum sticks are in an area close above the

surface of the sensor. It will also tend to bend the values towards the centre as the sticks move farther above the surface.

Perspective viewing gives the Kinect a much larger viewable area. The Kinect's depth sensor's field of view is 57 degrees in the horizontal direction and 43 degrees in the vertical direction. This means that at closer ranges, the Kinect cannot detect objects far to the sides of the camera whereas when depth is increased, objects far from the centre of view may be detected.

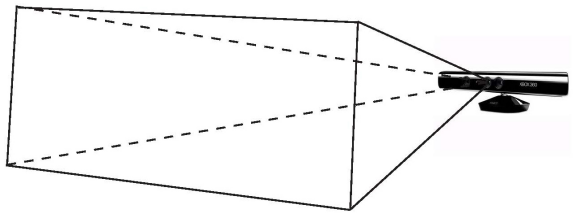

Figure 3.6: Kinect Viewable Area

To demonstrate the constriction on possible movements recorded by the radiodrum, we recorded the captured output of a user moving their hand back and forth while holding the radiodrum stick. As you can see, the Kinect is able to capture a much larger range of gestures.

3.4 Software Tools

A common set of software tools were used in the development of these prototyped systems. With the exception of Max/MSP and Ableton Live, these programs are written in open source c++, and are available online. These applications, toolkits, and software libraries are used to capture and manipulate data from our sensors, and output corresponding audio - visual feedback.

3.4.1 openFrameworks

openFrameworks [2] is a real-time, rapid prototyping toolkit with many similarities to it's precursor, the Processing [3] development environment. This toolkit is

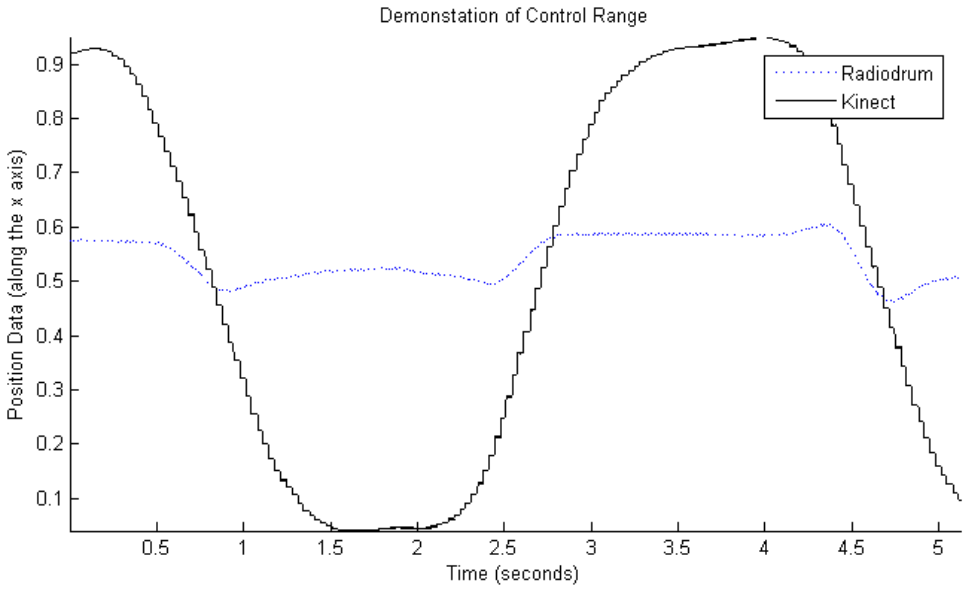

Figure 3.7: Horizontal Range of both controllers

mainly used by artists, musicians, and creative programmers. It's simple and intuitive framework and cross platform capabilities allow for rapid development and can easily incorporate other libraries.

3.4.2 Marsyas

MARSYAS (Music Analysis, Retrieval and SYnthesis for Audio Signals) [4], is an open source audio processing framework with specific emphasis on building MIR systems. It has been under development since 1998 and has been used for a variety of projects both in academia and industry.

3.4.3 OpenCV

OpenCV [5] is a c++ library of programming functions for real-time computer vision. Motion capture is often achieved with vision-based sensors, and so this widely-adopted library is essential to process vision-based sensor data.

3.4.4 OpenNI

OpenNI [6] is an organization that produces a software library for communication with Natural Interaction (NI) devices. This library provides both low level access to audio-visual sensor data, and also high level vision-based tracking algorithms. One of the main members of this organization, PrimeSense [1], is responsible for the development of the technology behind the Xbox Kinect.

3.4.5 Max/MSP

Cycling 74, the company behind Max/MSP [7], has long been the developer of this standard computer music software. The modular nature of the program, as well as the visual nature of the programming, has made it very popular among musicians, artists, and researchers. This program is used to capture audio data, including data streams from the Radiodrum. While the other open source libraries are embedded within a single openFrameworks project, Max/MSP is a standalone program that communicates with the rest of this application with Open Sound Control (OSC) protocol [8].

3.5 Future Work with Motion Capture

Another potential area of exploration involves comparing the three-dimensional coordinate measurements from both the radiodrum and the Kinect with a ground truth and attempting to compensate for the disparity.

We have shown that there is significant latency and temporal jitter from the Kinect data relative to the signals from the radiodrum. This makes direct fusion of the data difficult except for slow movements. One potential way to help resolve this issue is to extend the body model to include more of the physics of motion. The current body model is largely based on just the geometry of segments (a kinematic description) whereas a full biomechanical model would include inertial (kinetic) parameters of limb segments as well as local limb acceleration constraints. Once the biomechanical model

is initialized, it can be used to predict (feed-forward) the short term future motion and then the delayed motion data from the Kinect can be used to make internal corrections (feedback). Also, because internally the biomechanics body model will have estimates of limb segment accelerations, it would be relatively easy to incorporate data from 3D accelerometers placed on important limb segments (such as the wrist) to enhance motion tracking.

The Kinect sees the world from a single view and so although it produces depth information, this can only be for objects closest to it (in the foreground); all objects in the background along the same Z axis are occluded. This results in occlusion shadows where the Kinect sets these areas to zero depth as a marker. These regions appear as shadows due to the area sampling of the coded-light algorithm used to estimate scene depth. The use of a pair of Kinects at +/-45 degrees to the performance capture area has the potential to make the motion capture much more robust to occlusions and should improve spatial accuracy. Practically the IR coded-light projectors on each Kinect would need to be alternated on and off so they don't interfere. If the timing is synchronized and interleaved, the effective frame rate could be potentially doubled for non-occluded areas.

General free motion (motion with minimal physical contact) is usually imprecise in absolute terms as it relies on our proprioception (our internal sense of relative position of neighbouring parts of the body) combined with vision to provide feedback on the motion. The exception to this is highly trained free motion such as gymnastics or acrobatics. Physical contact with a target reduces the spatial degrees of freedom and provides hard boundaries or contact feedback points. In the absence of this hard feedback, gestures performed in free space are going to be difficult to automatically recognize and react too, unless it is highly structured and trained. This requirement could significantly distract from the expressive nature of a performance. Because the proposed future system will include a biomechanical body model of the motion, it

should be possible to predict potential inertial trajectories (gestures) in real-time as the trajectory evolves. This would allow the system to continuously predict the likely candidate match to the gesture and output an appropriate response. The trajectory mapping will need to be robust and relatively invariant to absolute positions and rotations.

Chapter 4

Motion-controlled Spatialization

Audio-scene rendering is an important aspect in the creation of realistic auditory environments. Sound spatialization has many applications, a prominent example being the film and video game industry. As these industries strive towards a fully immersive visual experience, these experiences will also require immersive audio. Ideally, we should be able to control aspects such as the strength, distance, and apparent motion of each sound source in the auditory scene. It is also important to properly render the reaction of the virtual space to a sound with the use of early reflections and reverberation. Much work has been done in the area of high-quality offline sound rendering, as well as lower quality real-time rendering [31]. As the computational power of computers increase, it is possible to create high-quality scenes in real-time. While the quality and efficiency of spatialization have been studied at great length, the control of the system has not been the focus. We present here a user control system for virtually rendered sounds, including a gestural control for the location of sounds, and a graphical user interface (GUI) to control other parameters and receive visual feedback.

In this chapter, we will discuss the background and previous research to do with spatialization and gesture-controlled spatialization. Understanding spatial sound synthesis requires knowledge of psychoacoustics, or in this case specifically how humans

localize sound. Spatialization also requires a physical model that incorporates psychoacoustics. For this system, Moore's room within a room model [46] is used in conjunction with the Image method to produce both direct and indirect ray paths from a sound to the listener. Details of implementation are given, including information on calculating the position of a sound source and its reflections around a room, and on using a tapped delay line to recreate sound in a synthesized location through a set of loudspeakers. Finally, a summary and suggestions for future work are presented.

4.1 Related Work

The first recorded system to control the spatialization of a sound dates back to 1951, when Schaeffer created the potentiometre d'espace [19]. Since then, there has been a great deal of work done on sound spatialization, and a variety of techniques have been developed. Dominant commercial systems often include 5.1 or 8.1 surround, a sure sign that a stereo pair is often not sufficient. Ambisonics [40] creates stable sound images and fewer distinct channels of audio data pairing each speaker with a decoder. Wave field synthesis [60] produces a consistent audio space by using tens or hundreds of speakers. Positioning of a sound source is done in [52] using vector-based amplitude panning, and in [14] using virtual microphones. The perceived direction in [58] is controlled by amplitude panning the direct sound, and perceived distance is controlled by adjusting the energy decay curve of reverberation and gain of the direct sound. Scene description models and rendering engine for interactive virtual acoustics are outlined in [31]. Li et al. [37] use the reverberation tails of measured room impulse responses in addition to the direct path and early reflections obtained by ray tracing.

The development of methods for gesture control of sound spatialization is presented in [43]. Three specific roles for control of spatialization are identified, the first

of which is directly relevant to our work with the radiodrum: a Spatial Performer that performs with sound objects in space by moving sound sources in real-time using gesture.

The Human Interface Devices (HIDs) used to control spatialization systems has also been the subject of much work. The implementations in [43] are done using instrumented datagloves, and in [42] using the Polhemus 3D electromagnetic sensor. Currently, audio engineers use panning controls such as sliders, soft knobs, and joysticks. When using sliders, for example, the level of each speaker must be controlled individually. When using joysticks, only two of three dimensions can be controlled simultaneously. Often, the process requires many iterations in which automations are adjusted to produce the desired effect. Part of the need for these iterations may be due to the fact that the position of these sounds are not controlled by a device that is capable of positioning objects in three dimensions. These systems also often rely only on panning alone and not delay lines. The use of spatial control, with its intuitive direct connection between stick position and sound source position is to the best of our knowledge a novel idea.

4.2 Sound Localization

When localizing sounds, human perception relies on binaural differences in amplitude and time. At high frequencies, the relative loudness of a sound is the dominant factor in localization. At lower frequencies, the difference in time of arrival between our ears becomes dominant. This shift in dominant cues is due to the wavelength and diffraction of a sound. Lower frequency sounds diffract around barriers, and are therefore more likely to sound with nearly equal intensity at both ears. Sounds above approximately 3kHz have a wavelength smaller than the distance between our ears, and any sensed difference in phase is indeterminate. So, diffraction makes intensity cues ineffective, and sound with a short wavelength makes time cues ineffective. It is

therefore important to use both time and loudness cues to properly model a sound space.

Most of our ability to localize a sound source occurs on a horizontal plane. However, the pinnae in the human outer ear allows us to distinguish sounds coming from above and below, and the listener will also tend to move their head to enhance localization. Depending on the number of speakers used to recreate a sound scene, it is possible for auditory illusions to occur, where one or more sounds can be localized improperly. Sound spatialization systems strive to simulate sound above and below the listener, as well as from side to side accurately.

4.3 Creating a Spatial Model

4.3.1 Image Method

The image method for calculating reflections in a room was originally proposed at Bell Laboratories in 1978 [11]. This time-domain model involves calculating the position of images in a space, rather than calculating all modes of a sound within a given frequency range. The reflected sound paths are created as images outside the boundaries of the room by reflecting the original sound across each wall. This method allows reflections within a room to be quickly simulated for a rectangular room, and can easily be expanded for a 3 dimensional rectangular box.

4.3.2 Moore's Model

Moore proposed a model using the image method to calculate the time and amplitude of each reflection for a set of surround speakers, and is presented as a room within a room [46]. This model treats each speaker as a window into a room that exists beyond the boundaries of the listening space. The distance of the original source and the reflections to each speaker are calculated, treating each speaker as a listening point.

In implementing Moore's model, we expanded the model for rooms as rectangular

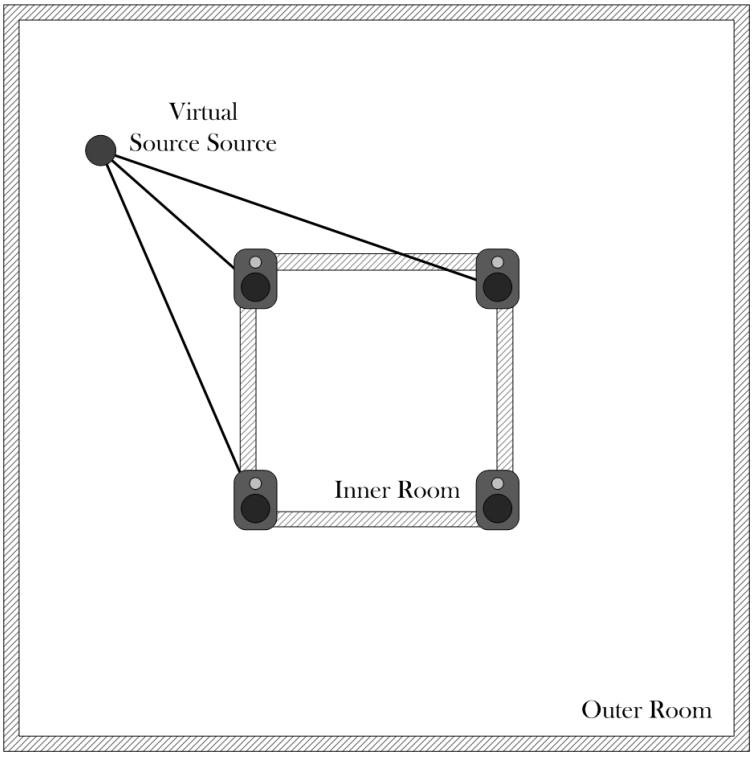

Figure 4.1: Room within a room model

boxes, with speakers at each of the 8 corners of the inner room. Moore's model is improved by treating each wall of the inner room as opaque. As seen in figure 4.1, the sound must be "seen" by a speaker to sound. The result of this is a clear sense of localization perceived by the listener.

4.4 Implementation

The implementation of this model can be separated into four components: gesture capturing, audio processing, virtual source and delay line calculations, and graphics processing. Figure 4.2 shows an implementation flow chart of the system.

4.4.1 Gesture Capturing

Our system captures the intended sound source position in real-time with the radiodrum, and displays the data graphically for additional feedback. Gestures created with with the radiodrum sticks can be monitored visually through the GUI and au-

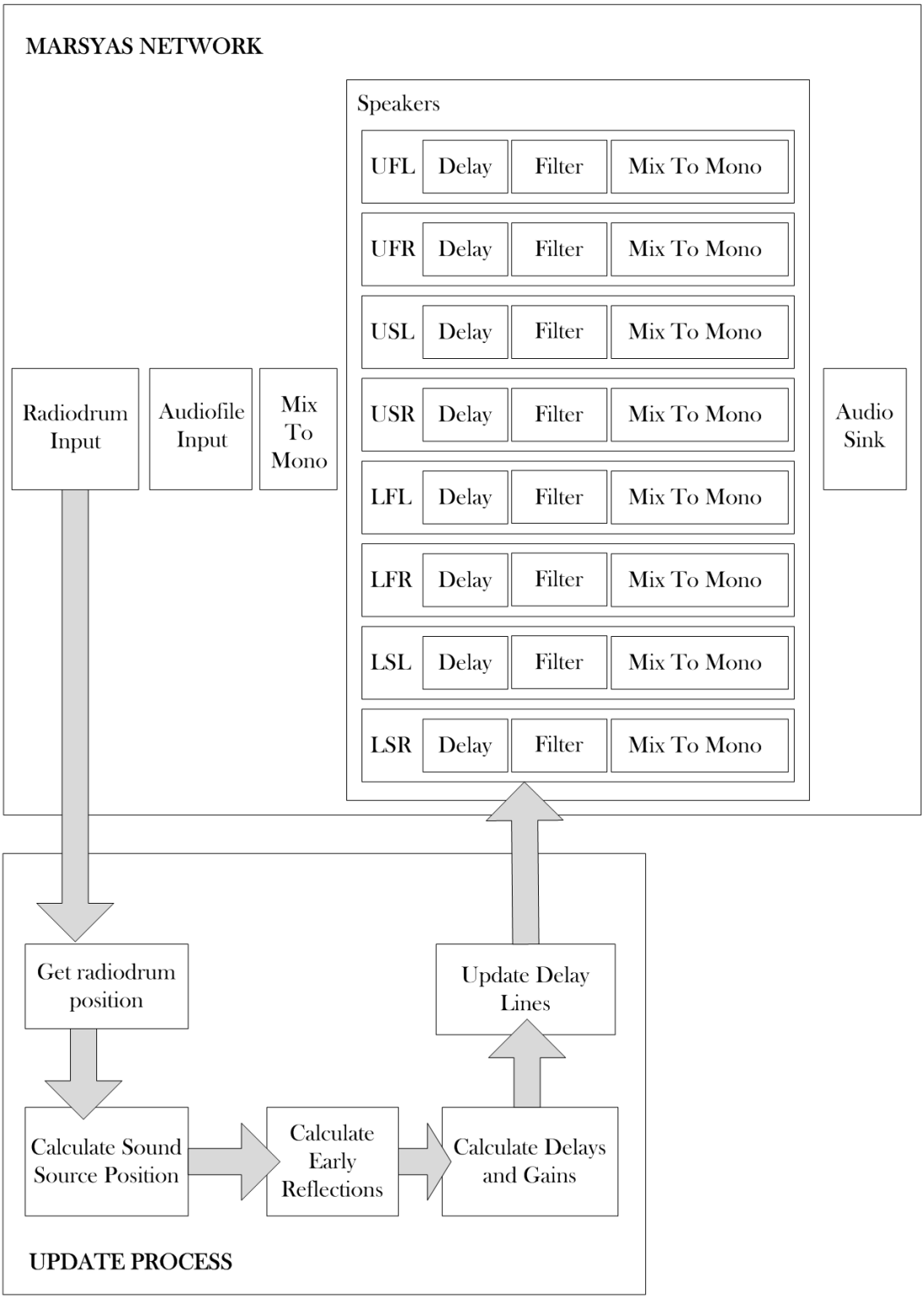

Figure 4.2: Implementation Flow Chart

rally from the surround speaker setup. In contrast with other systems used to control the position of a sound, the use of a 3D spatial sensor is intuitive, because the stick can be moved freely in three-dimensions. While the original design of this software used only the radiodrum, any sensor that returns 3D positions can be used, and the Kinect has also been used to control this spatialization system.

4.4.2 Audio Processing

All audio processing for this system is done using Marsyas. Our system includes modules that involve reading audio from a sound file source, computing delay lines, gains, filtering, and outputting multi-channel audio in real-time.

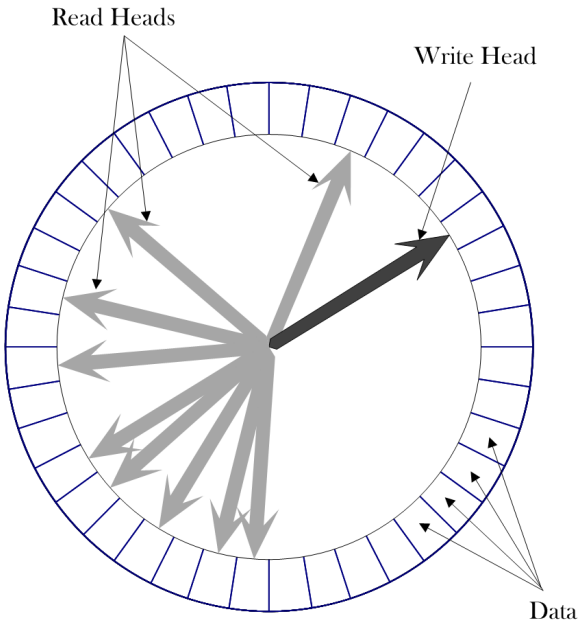

Figure 4.3: Delay Line Implementation

A set of delay lines are used to simulate early reflections, and each delay line has a corresponding gain. In Marsyas, a module implementing delay lines was created specifically for this application. The delay lines are implemented as a rotating read-heads on a circular buffer. The first pointer writes to the buffer with updated data. The following pointers read from the buffer at set delay points. Linear interpolation is used to smooth between frames of data as the delay lines vary, which occurs every

time the sound source placement changes.

4.4.3 Virtual source and delay line calculations

The calculations required to create proper delay lines involve determining the position of the sound source and each virtual source, and calculating the corresponding delays and gains for each source. The virtual position of early reflections were calculated using the image method, and we calculate the distance to each reflection, and the corresponding time delay and gain.

The early reflections were calculated using the image method. For simplicity, the implementation d description begins with only two dimensions. A set of imaginary rooms surrounding the center room are given indices $[i, j]$, $-N < i < N$ and $-N < j < N$, where N is the order of reflections. The order in each square is given by

$$N = abs(i) + abs(j) \tag{4.1}$$

The images then fall inside each imaginary room at a position of $[i*p_x, j*p_y]$. The final step is to account for the displacement seen in every second block. For example, along the x-axis, the positions of each image are either equal to $i * p_x$, or they fall within that block with a constant displacement. We define these displacements as:

$$disp_x = 2(w/2 - p_x) \tag{4.2}$$

$$disp_y = 2(h/2 - p_y) \tag{4.3}$$

Expanding this algorithm into 3 dimensions, the calculation of each reflection is done by iterating through the N^3 rooms, and calculating the order and position of each reflection. Once we have the position of each reflection, we determine if each reflection is "seen" by the speaker. We then calculate the distance from each virtual

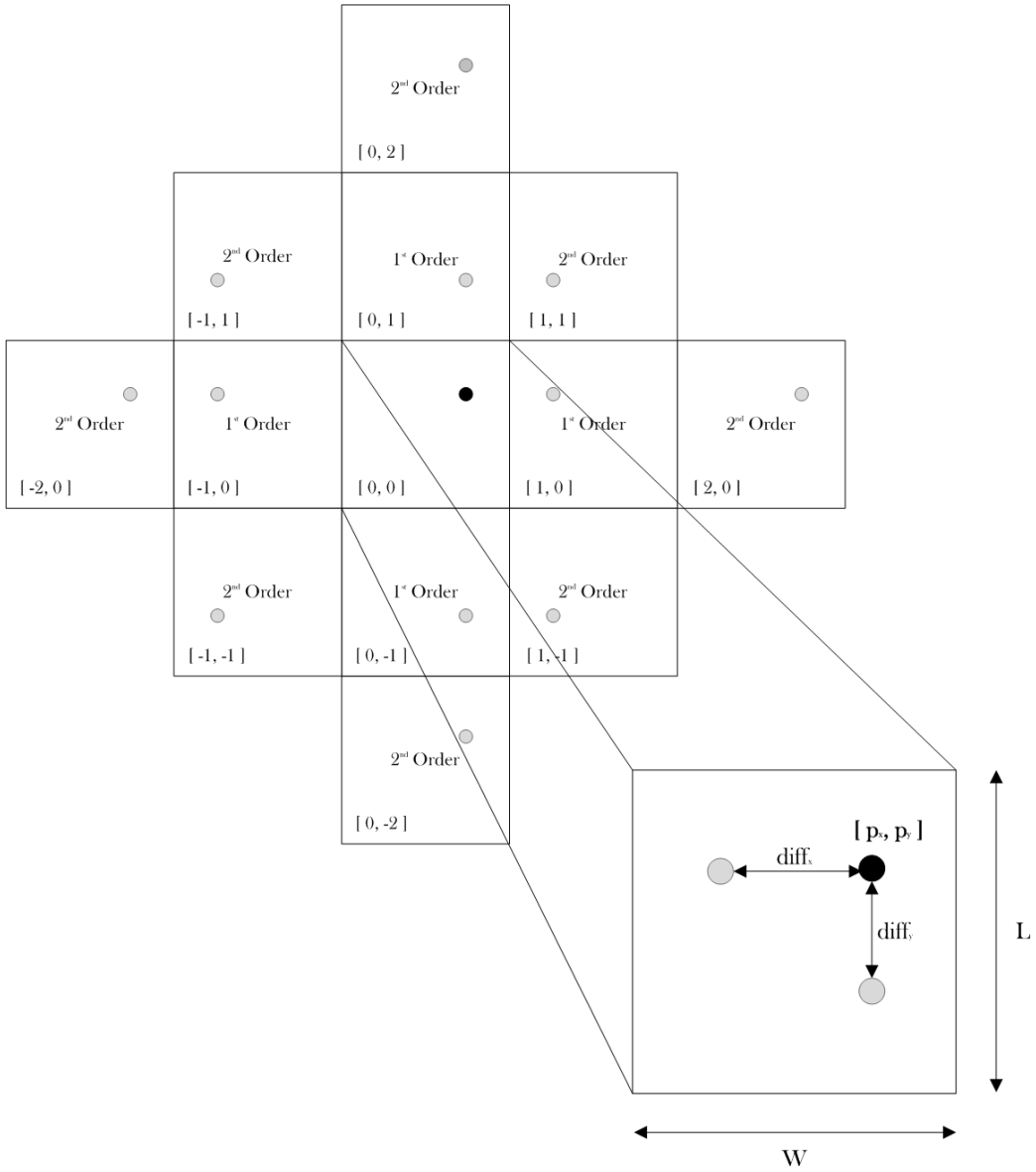

Figure 4.4: Image Source Model

image to each speaker using the distance formula.

$$d = \sqrt{(p_x - s_x)^2 + (p_y - s_y)^2 + (p_z - s_z)^2} \tag{4.4}$$

The delays and gains are relative to the calculated distances.

$$g = 1/(4 * pi * r^2) \tag{4.5}$$

$$t = d/c * fs \tag{4.6}$$

The calculated delays and gains are used to update the MARSYAS delay lines.

4.4.4 Visualization

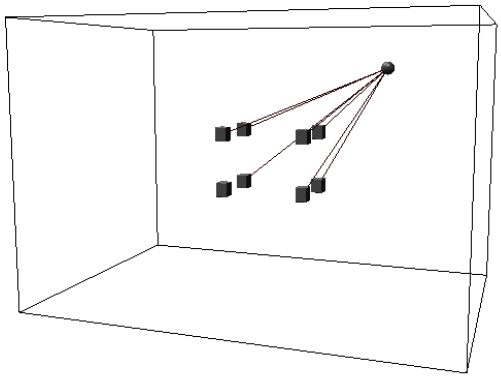

Figure 4.5: OpenGL Screenshot

OpenGL was used to create 3D graphics, a display showing the inner and outer rooms, as well as the position of the sound within the space. The visualization shows the size of the outer room, the position of the speakers in the inner room, and the position of our sound source. When controlling the position of our sound source, the visualization allows the user to see where the sound in moving with respect to the size of the inner and outer room. Without the visualization as additional feedback, it is difficult to conceptualize the scale of the inner and outer rooms.

For testing and debugging, we can add the images and reflected rays to the visualization, and separately display an impulse responses for each speaker showing the delay and amplitude of the reflections from source to speaker.

4.5 Summary and Future Work

In this chapter we presented a novel sound spatialization system using the radiodrum for gestural control of the movement of sounds within a space. The intuitive control and graphical interface distinguish this system from similar spatialization models.

Future work will includes thorough evaluation of this system. A user study in which users were asked to compare different controllers to complete tasks related to moving sounds in a 3D space. This study would give qualitative results as to the usefulness of free-space gestural control with spatialized sound. A separate or complimentary study comparing Moore's room-within-a-room method to other spatial synthesis models like ambisonics and wave field synthesis would also be interesting and beneficial research.

Chapter 5

Gesturally-controlled Music Browsing

Advances in technology have drastically changed how we interact with music. The increasing capabilities of personal computers have allowed listeners access to digital music collections of significant size. As the number of available songs increases, searching and browsing through this music becomes difficult. The conventional hierarchy of "Artist-Album-Track" and the spreadsheet interface of music software such as iTunes are still the dominant ways of organizing and navigating digital music collections. While this method is effective for finding a specific song when one knows exactly what they are looking for, it does not allow for effective browsing through music collections when there is no specific target song. To address this issue, browsing interfaces that are based on organizing music tracks spatially based on their automatically computed similarity have been proposed.

Content-based browsing has some advantages over traditional systems, many of which stem from the fact that users can browse music aurally, and no longer require a pictorial or textual representation. By removing the need for a keyword representation, we can possibly access music that has no associated text, or text available only in a different language. This type of audio browsing can also be useful for music creators or video game audio designers who need to sort through large collections of sound clips or sound effects. Accessing music information aurally makes sense intu-

itively, and even allows people with vision or motion disabilities improved access to the world of music [66]. We describe a novel interface for browsing music and sound collections based on automatically computed similarity, spatially arranging the audio files in 3D using self-organizing maps (SOMs), and browsing the sonified space using 3D gestural controllers.

The next section will deal with related work with self-organizing music browsers. Following that, details are given on how to map sound files into a 3D space, clustered based on similarity. The use of 3D gestural controls is important to this work, and thus navigating through the space is also discussed. Finally, details on implementation and some future work are presented.

5.1 Related Work

Novel interfaces for browsing music began to appear about ten years ago with SOMs being one of the first algorithms to be used for music clustering and visualization [24]. The early development of applications demonstrating these concepts such as the Sonic Browser [39], Marsyas 3D [65] and Musescape [63] was fuelled by advances in the field of Music Information Retrieval (MIR). Each system uses direct sonification rather than button triggered playback as a means of music browsing to create a continuous stream of sound while navigating the music space. In Pampalk, Dixon and Widmer [50] and Knees et al [33], a visualization of the organized music collection is proposed in which the clustered songs are represented as islands, where the height of each island is relative to the number of songs in each cluster, and the terrain itself is based on a 2D SOM. In each of these applications, navigation is achieved using a mouse or joystick. In Ness et al. [48], the authors explored the use of various controllers for interfacing with self-organized music collections. These interfaces include multi-touch smartphones, motion trackers like the wiimote, and web-based applications. While advances in self-organized browsing progressed, the use of augmented reality

in musical applications was being developed [51].

Often, augmented reality (AR) is understood to be related to display technologies. However, AR can be applied to any sense, including hearing. In Azuma et al. [13], a mixed-reality continuum is presented, with Augmented reality defined as virtual objects added to a real space. Another good example of early combinations of self-organized music collections and augmented virtual spaces is the "Search Inside the Music" program [35]. This application allows users to browse through a virtual 3D space of songs and also showed the songs on each album visualized with the cover art. The key contribution of the system described in this chapter is the utilization of gestural 3D control for interacting with a 3D self-organized map of music.

5.2 Organizing Music in a 3D space

One of the main goals of Music Information Retrieval is to approximately model the concept of "similarity" in music. Similarity can be determined by using manually assigned metadata, however MIR often also focuses on extracting features directly from the audio signal. A variety of methods have been proposed in self-organized music browsers to project high-dimensional feature data onto a lower dimensionality for visualization, such as Principle Component Analysis. Although there are many dimensionality reduction methods, the most common approach to organizing music collections is that of the Self Organizing Map [34]. In this case, a set of features is extracted from an audio file, producing a single high-dimensional feature vector representing each song. The feature vector corresponding to a piece of music or a sound is then then mapped to a corresponding set of coordinates in a discrete grid. Feature vectors from similar audio files will be mapped either to the same grid location or neighbouring ones. The resulting map reflects both an organization of the data into clusters as well as a mapping that preserves the topology of the original feature space.

The goal of feature extraction is to produce a vector of numbers known as features that represent a piece of audio. By choosing how the vectors are computed, we are able to come up with numbers that are similar when they correspond to perceptually similar sounds or music tracks. As described in [64], we extract features such as Flux, Rolloff, MFCCs (Mel-Frequency Cepstral Coefficients), pitch histograms and rhythm-based features. These audio features are extracted for very short periods of audio (usually under 25ms). An entire song would therefore have an array of numbers for each feature, depicting how these features change over time. To model large collections of songs, this sequence of feature vectors representing each song needs to be summarized into a single feature vector characterizing the music at the song level. To shorten the length of our feature vectors and simplify the calculations each sequence of a particular feature is summarized down to two single values: the mean and standard deviation. That way both the central tendency of the feature and the deviation from it are modelled. Finally the features are normalized to have values between 0 and 1 across the dataset.

$$V_k = [v_0, v_1, ..., v_N] \tag{5.1}$$

The resulting feature vector V is calculated for each audio file in our collection, and is given in Equation 5.1 where k is the song index, n is the number of features, and v_n is a normalized feature.

Most of the previous work in the area of self-organized music browsing involves SOMs that lie on a 2D grid. This has a nice correspondence with the majority of human-computer interfaces, like the mouse or touch screen tablets, which allow the user to navigate a 2D space. With the recent popularity of 3D Gestural controllers like the Kinect, exploring a 3D SOM is a natural extension of the current models. Luckily, the algorithm used to create 2D self-organizing maps is easily modified for

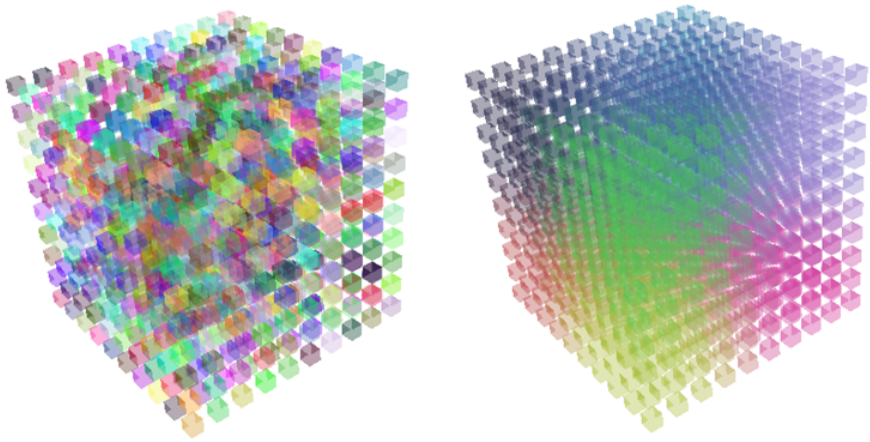

Figure 5.1: A 3D self organizing map before (a) and after (b) training with an 8-color dataset

any number of dimensions.

The self-organizing map is a type of artificial neural network, meaning that it is inspired by interactions between biological neurons. Our neural network begins with a set of objects referred to as nodes. Each node has an associated weight vector, W, as shown in equation 5.2, and spatial placement $P = [x, y, z]$. Although the nodes in figure 5.1 have been spaced evenly within a cube, these nodes could hypothetically be placed in other, more arbitrary formations. Initially, the weights of each node are set randomly. As the organization process progresses, the weights of each node will begin to align more closely with their neighbours and also more closely with our song features. This process is depicted in Figure 5.1, where each node has weight vector visualized as a colour. Initially, the weights shown in this figure are random (Figure 1a). As the SOM is trained with 8 distinct colours, the weights of each node become organized (Figure 1b).

$$W_k = [w_0, w_1, ..., w_N] \tag{5.2}$$

The training process involves selecting a song to train the map with and determining which node represents that song the best. Similarity between songs and nodes

is calculated as the euclidian distance between the song features and node weights, as shown in Equation 5.3.

$$d = \sqrt{\sum_{k=0}^{N} (V_k - W_k)^2} \tag{5.3}$$

The smallest distance corresponds to best-matching node or best-matching unit (BMU). Now each node in the vicinity of the BMU is updated with a new set of weights, adjusted to become more like our BMU. Equation 5.4 described how this adjustment is made. V(t) is the feature vector, W(t) is the weights vector, and L(t) is a learning function, which decays over time, takes into account the distance between the nodes, and allows the organizing algorithm to converge.

$$W_k\,(t+1) = W_k\,(t) + L\,(t)\,(V_k\,(t) - W_k\,(t)) \tag{5.4}$$

The learning function is made up of two components, and it controls how much change is allowed to occur in a given iteration. The first is the component that allows the distance between each node and the BMU to make closer nodes more similar to the BMU than nodes that are farther away. The second component allows the system to converge over time, so that L(t) slowly approaches zero at a rate determined by the iteration t and a time rate τ.

$$L\,(t) = exp(-d^2) * exp(-t/\tau) \tag{5.5}$$

By iteratively training our SOM, our resulting nodes reside in a space where nearby nodes have similar weight vectors. Each song is mapped to the most similar node, resulting in a set of songs residing in a space where nearby songs have similar feature vectors. In Figure 5.2, you can see that songs from similar genres will tend to be near one another. Note that the self-organizing map algorithm has no knowledge

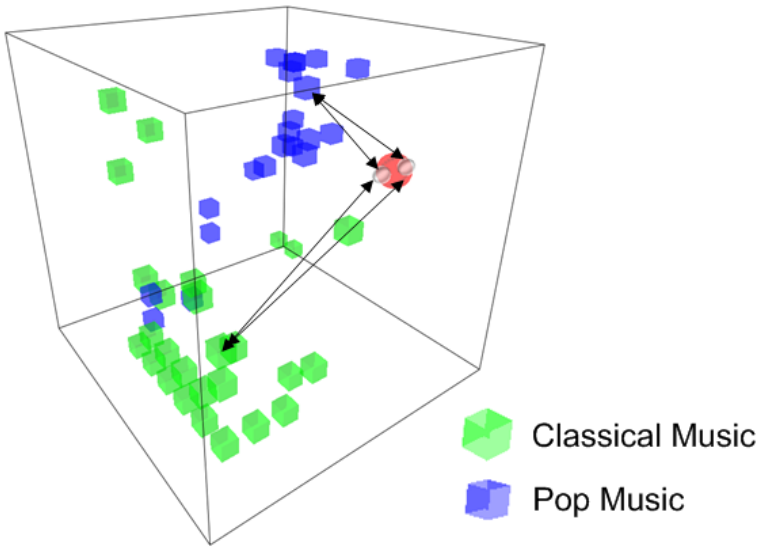

Figure 5.2: 3D SOM with two genres and user-controlled cursor

of the genre labels and their spatial organization is an emergent property of the mapping and the underlying audio features.

5.3 Navigating through the collection

Once our songs have been organized into a virtual 3D space, user interaction becomes a significant consideration. Since the use of 3D sensors was one of the primary motivations behind this work, our focus has been on using sensors capable of reporting gesturally-produced position data for two or more points. How we go about using that captured motion is another point of discussion, and we present here only the beginnings of this interaction with continuous play-back of music for continuous gestures. Previous work has been done into user interaction with 2D visualizations for music browsing [38], and similar concepts can be applied to the 3D scenario. We utilize two controllers: the radiodrum and the Kinect.

Using the 3D spatial sensors described in Chapter 3 as a set of 3D cursors, we want to sonify the organized sounds as we move our 3D cursors about. The simplest way to do this is to simply play back songs from whichever node is currently closest

to one cursor, and only one song plays back at a time. The other hand could then be free to perform other types of control gestures. Another content-aware browser [57] presented a different method for playing back songs. In this case, the user can manipulate the centre point and radius of an encompassing circle, and any songs within the circle will play simultaneously. To modify this method for our purposes, the two cursors were made to act as the bounding points for a variable-size sphere. Nodes with positions within the user-controlled sphere are sonified, with a gain relative to their nearness to the centre of the sphere.

Once the cursor data from the sensors is mapped to playback in the auditory representation of our sound collection we need a richer gestural language to enhance the user control. For example, once music exploration is complete and the user has found a song they would like to listen to, they will want to listen to that song and stop searching. Our simple way of implementing this functionality is to use timers, so that if we preview a song for longer than a set duration it will trigger song playback. Each node is sonified with a loudness based on its position relative to the cursor. By creating listening points that surround our cursor, we are able to perform multi-channel panning. As shown in figure 5.2, two smaller points are situated on either side of the user's current position, representing the the two listening points required for a stereo reproduction. This spatialization gives an aural sense of space and direction for navigation of our music collection.

5.4 Implementation

The hardware required for this music browser is simple: a controller, a computer, and a sound system. Figure 5.3 demonstrates the application design and interactions between the devices and software libraries. The software libraries used for this system are described in Chapter 3. The SOM data file is a small text file containing a list of songs with their accompanying metadata and SOM position.

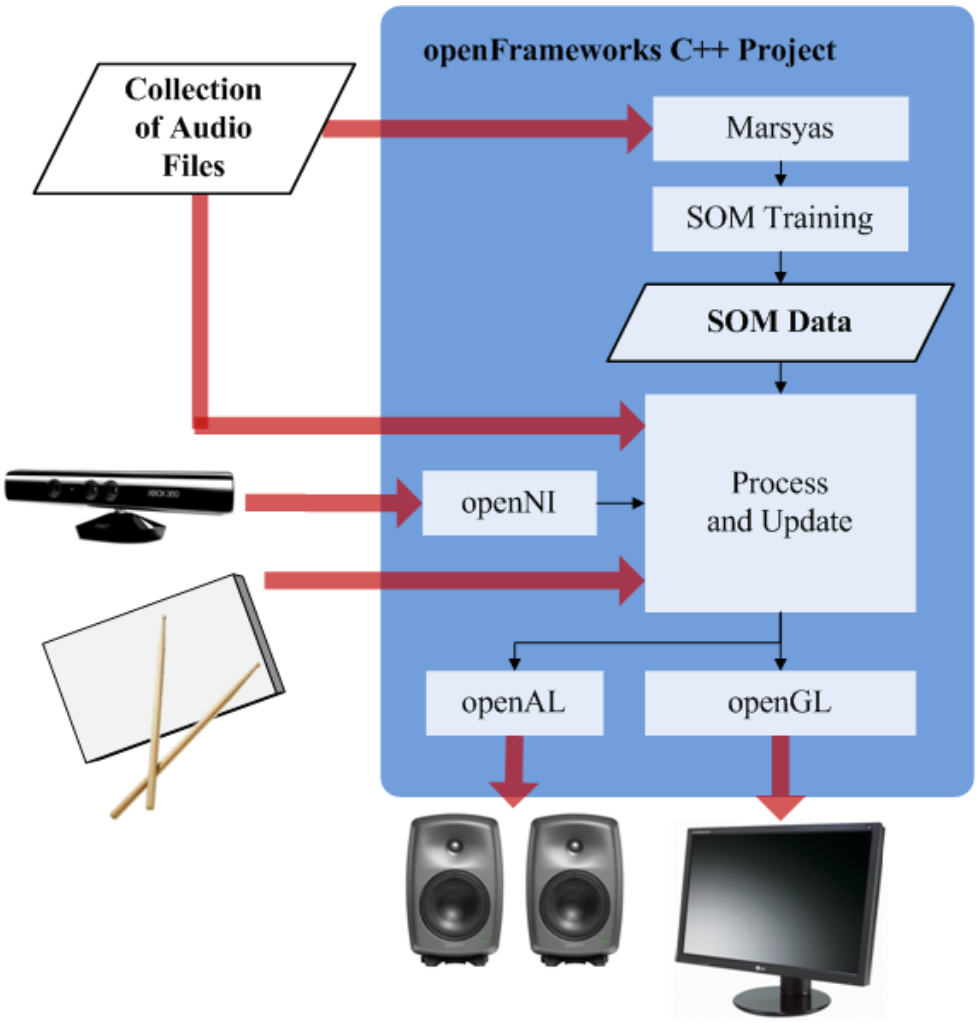

Figure 5.3: Implementation Diagram

5.5 Summary and Future Work

The self-organized map has become a popular method for organizing songs based on similarity. This type of music browser not only reflects the way that how we interact with music is changing, it also reflects how our interaction with technology and computers is changing. By expanding previous work with self-organized music collections and adding a third dimension, it is possible to convey additional information and browse extra songs. Additionally, navigating this type of map is a good example of the advantages 3D gestural sensors like the radiodrum and the Kinect have in specific control contexts and the more natural interaction they enable.

Future work will involve performing user evaluations that could help to answer questions about browsing music with this system. Three-dimensional SOMs have the possibility to represent richer topological spaces, reflecting more accurately the relationship between songs in our music collection. Furthermore, using 3D gesture-based controllers to navigate a 3D space probably offer advantages over using a joystick or other 2D controllers. However, without the proper evaluation provided by a user study any claims we can make are purely speculative. Further evaluation of this system is required, in which the time it takes to complete tasks of browsing for certain music will be measured. Quantitative comparisons between 3D and 2D SOMs can also be performed, where the distance between similar songs are compared for the same set of songs.

Chapter 6

Hyper-Vibraphone

Although instruments of the modern symphony orchestra have reached maturity, musical instruments will continue to evolve. A significant area of development is in electro-acoustic instruments, combining natural acoustics with electronic sound and/or electronic control means, also called hyperinstruments [32]. The evolution of new musical instruments can be described in terms of both the sound generator and the controller. Traditionally, these separate functions are aspects of one physical system; for example, the violin makes sound via vibrations of the violin body, transmitted via the bridge from the strings, which have been excited by the bow or the finger. The artistry of the violin consists of controlling all aspects of the strings vibrations. The piano is a more complex machine in which the player does not directly touch the sound-generating aspect (a hammer hitting the string), but still the piano has a unified construction in which the controller is the keyboard, and is directly linked to the sound-generation. For hyperinstruments these two aspects are decoupled, allowing for the controller to have an effect that is either tightly linked to the sound produced (as any conventional acoustic instrument has to be) or can be mapped to arbitrary sounds.

Modern advancements in consumer based digital technologies are allowing for unprecedented control over sound. What the traditional model provides in terms

of performance is a context for music that has been mastered over centuries of care and practice. Traditional musical models have proven to be socially binding, bringing people together through music. We believe that retaining some tradition techniques is critical with evolving methods of music production, dissemination and performance.

This chapter outlines our progress towards the development of a system for the design of digitally extended musical instruments and their utilization in multi-media performance contexts. We examine the performance of the Kinect when adapted for use with a vibraphone as an example of effectively embedding new technologies onto traditional instruments with two examples of gestural enhancement.

The addition of gestures to a pre-existing instrument usually is done with one of two objectives [47]:

- Gestural Range

  To take advantage of intentional or unintentional gestures that would not conventionally result in sounds, increasing the range possible adjustments and adaptations.

- Adaptability

  To find specific ways of making a musical instrument as adaptable as possible, to allow the performer or instrument itself to implement changes easily and intuitively.

Two methods for gesturally-enhancing a vibraphone performance have been developed. The first aims to increase the *gestural range* of the vibraphone with an application called "Magic Eyes". This software capture gestures and uses them to modify filter parameters on sounds from a live performer. Conversely, the second enhancement aims to increase the *adaptability* of the vibraphone. This application, called "Fantom Faders", tracks the tips of the performer mallets relative to the bars of the vibraphone, and turns each bar into a virtual slider.

The related work for this chapter will include gesturally enhanced hyper-instruments and also hyper-vibraphones specifically. Both enhancements will then de described, and finally, directions for future work is discussed.

6.1 Related Work

Previous areas of research include Randy Jone's Soundplane (a force-sensitive surface for intimate control of electronic music that transmits x, y and pressure data using audio signals generated by capacitive input captured at a high audio sampling rate) [30], Adam Tindale's acoustically excited physical models using his E-drumset (a piezo based drum surface that uses audio signals, created by actual drumming, to drive physical models) [61], and the radiodrum [49].

Commercial mallet-percussion based MIDI control solutions are limited. The Simmons Silicon Mallet and MalletKat both incorporate rubber pads with internal FSR's laid out in a chromatic keyboard fashion and vary in octave range. This Simmons offered very little configurability as it was basically an autonomous instrument and had a 3 octave mallet keyboard controller played with either vibraphone mallets or drum sticks. The Malletkat offers the same typical internal MIDI configurability as high-end piano style keyboard controllers and can connect to any MIDI sound module. K&K Sound[9] produced a (now discontinued) piece of hardware that extracts control data from a vibraphone using an array of sensors. The Xylosynth, by Wernick[10], is a digital instrument that aims at providing the same haptic response of a wooden xylophone while yielding MIDI instead of acoustic data. The Marimba Lumina [26] is another mallet style controller, and is able to gather velocity, position and contact. Additionally, it allows one to relate different controls to be triggered when playing different bars. These instruments are, in general, technologically robust, but unable to offer the same haptic feedback as an acoustic instrument and typically require the instrumentalist to modify playing technique in order to achieve successful results.

Our work has been influenced by several new music instrument ideas beyond the pitched percussion family. The Theremin is unique and relevant in that it is played without physical contact controlled by 3D hand gestures. The modern Moog Ethervox, while functionally still a theremin, can also be used as a MIDI controller, and as such allows the artist to control any synthesizer with it.

With the development of new gaming interfaces based on motion controls, it became easier to generate musical interfaces controlled by movements. That allows one to use natural motion, which, as observed by [22], is an inherent part of the performance of a pitched-percussion player. Motion-based game controllers were used as musical tools in [27], taking as basis the WiiMote device.

6.2 Gestural Range (Magic Eyes)

In this section, a system is presented that captures the performer's gestures, and converts them to MIDI [1] data for composers, music producers, and musicians to incorporate into their music. This system filters live audio from the vibraphone through a set of filters that morph and distort the sound.

In order to take into account the specifics of musical interaction, one needs to consider the various existing contexts- sometimes called metaphors for musical control [67] where gestural control can be applied to computer music. In order to devise strategies concerning the design of new hyperinstruments for gestural control of sound production/modulation, it is essential to analyze the characteristics of actions produced by expert instrumentalists during performance [22]. The importance of this analysis can be justified by the need to better understand physical actions and reactions that take place during expert performance. The xylophone makes for good coupling of technology because of the very constant visceral movement required to produce sound. This is good for two reasons: the sound produced is well-suited for

[1] Musical Instrument Digital Interface

conversion into control data and the motion involved in the process is ideal for computer vision recognition. Thus, along with furthering our 3D control practice, we subsequently seek to advance the role of the acoustic xylophone tradition to incorporate advanced HCI capabilities.

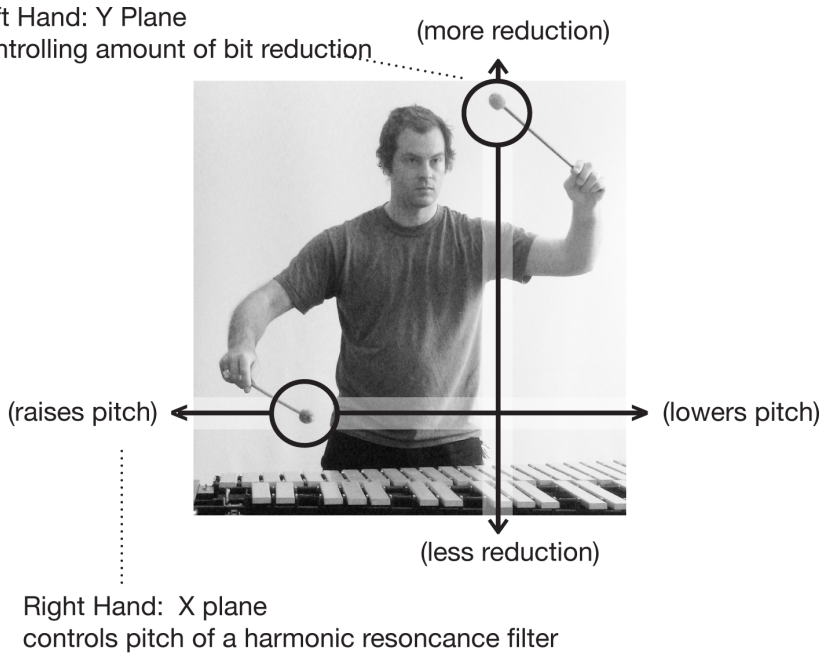

Figure 6.1: Music Control Design

The specific mappings and filter parameters chosen were not arbitrary, but rather specific to the musician's artistic practice. Being both a sound designer and computer musician, the researcher is also a vibraphonist and thus chose intuitive mappings based on specific vibraphone techniques within a given original composition, and subsequently chose the extended digital sound design parameters based on the familiarity of both the music and instrument's natural characteristics, having a background with the devices, their workings and characteristics as well.

The audio from the acoustic vibraphone was captured with an Akai 414 microphone and input into an RME Fireface 800 connected to a Macbook Pro hosting Ableton Live. The Vibraphone audio channel went to the Master bus with a slight

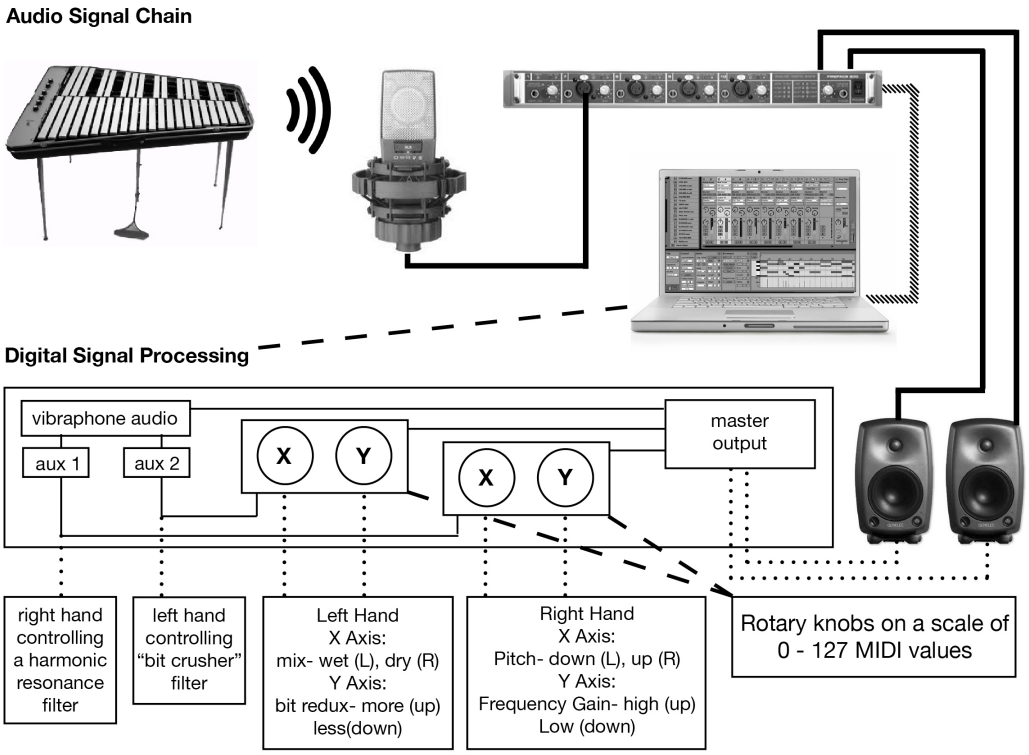

Figure 6.2: Audio Signal Chain

bit of equalization and compression, it was also routed to Auxiliary sends 1 and 2. On Aux Return 1 was a harmonic resonance filter plug-in. The X axis in the right hand controlled the tonic of the 7 voice harmonic scale pre-programmed into the filter. Moving the mallet to the left would lower the tonic, as would the inverse; moving the mallet to the right, would raise the pitch. The Y axis controlled the global frequency gain of the filter, allowing a performer to over-accentuate the high frequencies of the filtered audio by raising the mallet and boosting low frequencies as the mallet is lowered. Aux Return 2 hosted a bit reduction plug-in that was used as a type of digital distortion, much the way a guitarist would use an analog fuzz

pedal [2]. The X axis in the left hand controlled the the ratio of the filter applied to the source signal. Moving the mallet to the left would reduce the effect resulting in a dry source signal, moving the mallet to the right, would increase the 'wetness' of the filter effect. The Y axis controlled the rate of reduction. The bit rate decreases when the mallet is raised resulting in more distortion, lowering the mallet results in less distortion. These effects are calibrated so that the default playing position of the vibraphone results in no processing so they react only to extended gestures.

6.3 Adaptive Control (Fantom Faders)

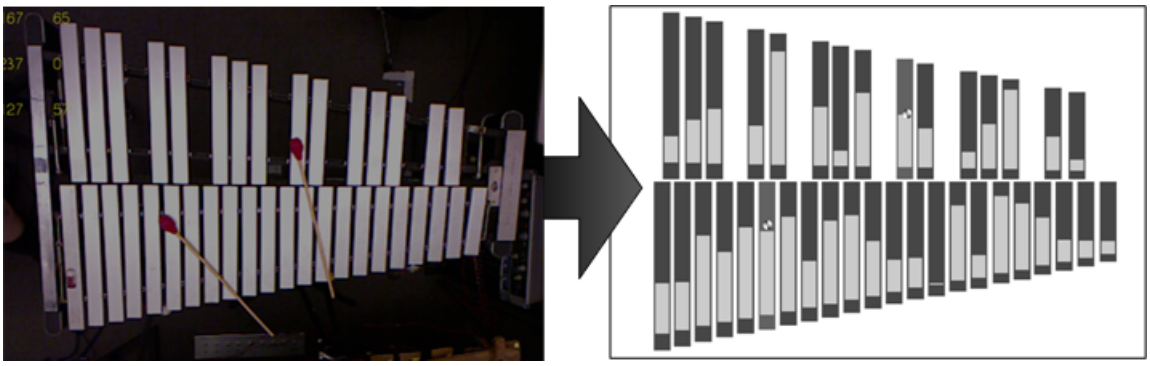

Figure 6.3: Virtual Vibraphone Faders

The Marimba Lumina is a good example of adding adaptability to an instrument [26]. In this case, the instrument created synthesized sounds that could be controlled with the gestures created by moving the mallets on a surface, where each key of the mallet becomes a control surface. We draw upon this metaphor, and extend it for use on an acoustic percussive instrument.

A new extension of our work in non-invasive sensing introduces a form of augmented reality to our hyper-vibraphone. Using the Kinect sensor and computer vision libraries, we are able to detect the position of the percussionist's mallet tips. This tracking used to augment each bar of the vibraphone with the functionality of a fader.

[2]Except, in this case, the Ghanaian Gyil was a direct reference in that in a traditional context the Gyil's resonators (made of gourds) would have holes drilled into them with spider egg casing stretched over them resulting in an intentional distorting buzz of the acoustic sound

Using this technique on a 37 bar vibraphone, it is possible to provide the functionality of 37 virtual faders that may be used as traditional controllers. This augmentation, illustrated in Figure 6.3, provides an intuitive platform that allows the performer to control a large number of sliders without turning away from the instrument.

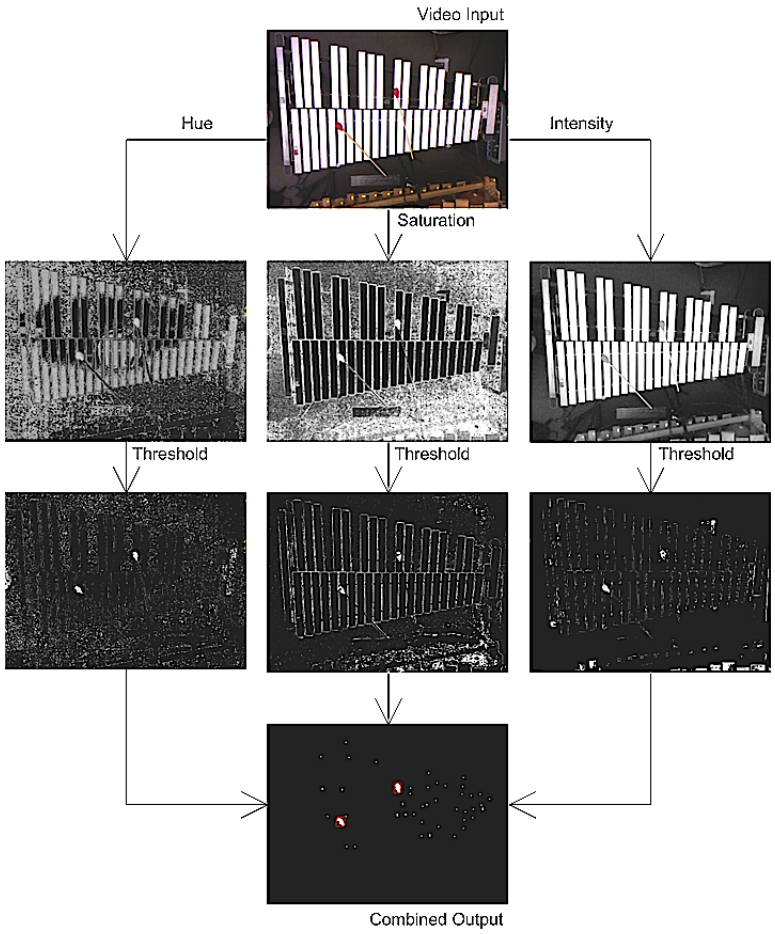

Figure 6.4: Computer Vision Diagram

First, we detect the positions of the mallet tips. Then, the position data is transformed to the same space as a virtual, scaled representation of the vibraphone. Finally, boolean logic structures are used to turn these positions into control data ready to be mapped onto sonic and musical events. In order to detect the positions of the mallet tips, the color image from the video camera is transformed into Hue, Saturation, and intensity Values (HSV). Each of these signals undergo thresholding

to filter out unwanted colors. The resulting signals are combined, and a contour algorithm [21] is executed. This process, shown in Figure 6.4, yields bounding rectangles that identify the mallet tips. The centroid of the bounding rectangle is assumed to be the position of the mallets in the virtual representation.

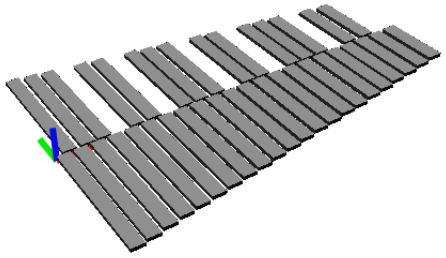

Figure 6.5: Virtual recreation of the vibraphone

Determining the position of the mallet tips in terms of the vibraphone bars requires the creation of a virtual representation of the instrument. This representation was created using explicit measurements of the vibraphone's bars. These measurements were used to create a model consisting of a the set of bars, each with a corresponding placement, size, pitch, and an associated control value. Our computer vision algorithms supply the position of these mallets relative to the camera, but we actually want our position data in a virtual space containing our instrument. For the virtual faders, this is realized by means of the linear transformation shown in Equation 6.1. Effectively mapping the tracked mallets involved several transformations, and requires a calibration phase in which the position of the vibraphone with respect to the camera is also recorded.

$$p_{norm} = \frac{p_o - p_{min}}{p_{max} - p_{min}} \tag{6.1}$$

Our Kinect instrument will be compatible with virtually all MIDI hardware/software platforms so that any user could develop their own custom hardware/software in-

terface with relative ease. The externals we will develop in Max/MSP for use in Max4Live will enable anyone to plug in a Kinect and adapt our language of 3D sound control to their own practice with minimal configuration. This has many implications, putting technology that previously was restricted to research purposes into the common computer musician's hands. Our shared belief is that these new instruments have a legitimate place [32] with potential to become part of an embedded conventional musical practice, not just a research curiosity. While hyperinstruments might seem niche or esoteric at this point [15], historic innovations such as the leap from monophonic to polyphonic music, electrical amplification of the guitar, and computers in the recording studio all brought skepticism, eventually becoming mainstay practices.

Once we have obtained the position of the mallets in the same space as the vibraphone, the system yields information on what bar is currently covered by the mallet, and a fader value associated with that bar. A delay-based activation time was added to the algorithm, so that the performer must pause on each bar for a pre-defined time before the sliders will start to change.

6.4 Summary and Future Work

One of our goals is to go beyond simple static one-to-one mappings and incorporate Machine Learning for gesture recognition. While one-to-one mappings are by far the most commonly used, other mapping strategies can be implemented. It has been shown that for the same gestural controller and synthesis algorithm, the choice of mapping strategy became the determinant factor concerning the expressivity of the instrument [53]. We have been actively developing a systemized mapping strategy for many years and will use this as an apparatus to inform and evaluate our advancements with the Kinect.

The 640x480 resolution of the camera used in our prototype is sufficient to perform

accurate detection of the mallet position. The main restriction regarding resolution is that the camera should point at the vibraphone from a distance that makes each bar to be present in a reasonable number of pixels. While this may suggest using a higher resolution camera, it is important to note that the algorithms must be executed in real-time, therefore less data is desirable. Future work will involve fusing the video camera data with IR sensors data, as well as using motion tracking algorithms in addition to contour detection. This fusion will improve the robustness of the tracking system, and address the current sensitivity to changes in ambient lighting. The addition of more sensor information will also allow us to track the mallets in 3D and create a more sophisticated gestural relationship with the hyper-instrument.

Chapter 7

Conclusions

In the previous chapters, several systems have been proposed and developed in an attempt to use free-space gestural control to solve problems in audio and music technology. Three distinct systems were proposed, each with related and unique potentials for heightened expression and interaction.

- Gesture-controlled spatialization can improve the creation immersive auditory scenes

  The addition of gestural control to the creation of 3D spatial audio has the potential to improve the process of the creating 3D audio. Not only could this lead to a more enjoyable experience for the sound technicians and composers who create the multi-channel surround sounds, but the end result would be a more immersive environment for all concert or movie goers to enjoy.

- Gesture-controlled music browsing provides an effective way to access aural information

  This music browser is like an augmented reality, with the virtual layer of the songs lying on top of reality. The current method of browsing for music is very effective if the listener knows exactly what they want to listen to,

but is less helpful when browsing through unknown music or music with no associated text. The three-dimensions of free-space have the ability to present richer topologies and more accurately reflect the relations between different audio files, and browsing this space with 3D gesture controllers is natural.

- Gesture-extended hyper-instruments allows for a new set of expressive musical tools

  The interaction between gesture and sound has possibilities to reach a wide audience through performance art in a multi-media context. By drawing on the previous long standing techniques and rituals that come with a traditional instrument, the use of new technology has the ability to reach a wide audience. With the satisfaction of the audio-visual feedback created from this type of control, there is no doubt that research into gesturally-controlled hyper instruments will continue to be a source of great interest to researchers and musicians alike.

As the sensors for gesture-based control develop and become more mainstream, and virtual or augmented reality become more common, this development of these types of system may seem quite ordinary. However free-space gestural control is a field interesting not only for controlling sounds, but for many types of interactions with information and computers.

7.1 Recommendations for Future Work

The scope of this work is broad, and so possible areas of future work are also quite vast. Recommendations can be made about many aspects of this work. Improving the speed and accuracy of motion capture is an obvious ongoing effort and important to create expressive musical interfaces. Fusing streams from multiple sensors to take

advantage of the strengths and limitations of different sensors is an interesting future work with many possibilities.

Each gesture mapping described in this paper requires further evaluation and comparison with similar methods and similar controllers. Likely, several user studies could be performed to evaluate many aspects of the gestural-control paradigm. The development of user studies would aim to address some questions, like the following:

- How can free-space gestural control improve current systems, like spatialization systems or music browsers?
- When is 3D free-space control better than a 2D controller and vice-versa?
- How important is the scale of the gesture to the size of an auditory change, and how small precise can these gestures be?

An evaluation of these systems is the next step towards creating improved versions of each of these systems.

Bibliography

[1] [Online]. Available: http://www.primesense.com/

[2] [Online]. Available: http://www.openframeworks.cc

[3] [Online]. Available: http://processing.org/

[4] [Online]. Available: http://marsyas.info

[5] [Online]. Available: http://opencv.willowgarage.com/wiki/

[6] [Online]. Available: http://www.openni.org/

[7] [Online]. Available: http://www.cycling74.com/

[8] [Online]. Available: http://opensoundcontrol.org/

[9] [Online]. Available: http://www.kksound.com/vibraphone.html

[10] [Online]. Available: http://www.wernick.net/

[11] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[12] D. Arfib, J. M. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces," *Org. Sound*, vol. 7, no. 2, pp. 127–144, Aug. 2002.

[13] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *Computer Graphics and Applications, IEEE*, vol. 21, no. 6, pp. 34 –47, 2001.

[14] J. Braasch, "A loudspeaker-based 3d sound projection using virtual microphone control (vimic)," in *Audio Engineering Society Convention 118*, 5 2005.

[15] M. Burtner, "A theory of modulated objects for new shamanic controller design," in *Proceedings of the 2004 conference on New interfaces for musical expression*, ser. NIME '04. Singapore, Singapore: National University of Singapore, 2004, pp. 193–196.

[16] C. Cadoz, M. M. Wanderley, and I. C. Pompidou, "Gesture: music," in *In: M.M. Wanderley and M. Battier (Eds.), Trends in gestural control of music, Paris, IRCAM/Centre Pompidou*, 2000.

[17] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe, "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems," *Comput. Music J.*, 2000.

[18] A. Camurri, B. Mazzarino, M. Ricchetti, R. Timmers, and G. Volpe, "Multimodal analysis of expressive gesture in music and dance performances," in *Gesture-Based Communication in Human-Computer Interaction*. Springer Berlin / Heidelberg, 2004.

[19] J. Chadabe, *Electric Sound: The Past and Promise of Electronic Music*. Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[20] A. Cont, T. Coduys, and C. Henry, "Real-time gesture mapping in pd environment using neural networks," in *Proceedings of the 2004 conference on New interfaces for musical expression*, 2004.

[21] I. Corporation, *Open Source Computer Vision Library.* USA: http://developer.intel.com, 1999.

[22] S. Dahl and A. Friberg, "Expressiveness of musician's body movements in performances on marimba," in *Gesture-Based Communication in Human-Computer Interaction.* Springer Berlin / Heidelberg, 2004.

[23] V. Frati and D. Prattichizzo, "Using kinect for hand tracking and rendering in wearable haptics," in *World Haptics Conference (WHC), 2011 IEEE*, june 2011, pp. 317 –321.

[24] M. Frühwirth and A. Rauber, "Self-organizing maps for content-based music clustering," in *In Proceedings of the 12th Italien Workshop on Neural Nets (WIRN01), Vietri sul Mare.* Springer, 2001.

[25] C. Geiger, H. Reckter, D. Paschke, F. Shultz, and C. Poepel, "Towards participatory design and evaluation of theremin-based musical instruments," in *In Proc. International Conference on New Interfaces for Musical Expression*, 2008.

[26] M. Goldstein, "Playing electronics with mallets extending the gestural possibilities," *Trends in Gestural Control of Music*, 2000.

[27] S. Heise and J. Loviscach, "A versatile expressive percussion instrument with game technology," in *Multimedia and Expo, 2008 IEEE International Conference on*, 23 2008-april 26 2008, pp. 393 –396.

[28] A. Hunt, M. M. Wanderley, and R. Kirk, "Towards a model for instrumental mapping in expert musical interaction," 2000.

[29] A. Hunt and M. M. Wanderley, "Mapping performer parameters to synthesis engines," *Org. Sound*, vol. 7, no. 2, 2002.

[30] R. Jones, P. Driessen, A. Schloss, and G. Tzanetakis, "A force-sensitive surface for intimate control," *NIME*, pp. 236–241, 2009.

[31] J.-M. Jot and O. Warusfel, "A Real-Time Spatial Sound Processor for Music and Virtual Reality Applications," in *ICMC: International Computer Music Conference*, Banff, Canada, Septembre 1995, pp. 294–295.

[32] A. Kapur, E. S. M. S. Benning, and G. T. Trimpin, "Integrating hyperinstruments, musical robots & machine musicianship for north indian classical music."

[33] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web," in *In MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*. ACM Press, 2006, pp. 17–24.

[34] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464 –1480, sep 1990.

[35] P. Lamere and D. Eck, "Using 3d visualizations to explore and discover music," in *ISMIR*, 2007.

[36] M. Lee, A. Freed, and D. Wessel, "Real time neural network processing of gestural and acoustic signals," in *Proc. Intl. Computer Music Conference*, Montreal, Canada, 1991.

[37] Y. Li, P. F. Driessen, G. Tzanetakis, and S. Bellamy, "Spatial sound rendering using measured room impulse responses," 2006 IEEE International Symposium on Signal Processing and Information Technology, pp. 432–437, 2006.

[38] A. S. Lillie, "Musicbox: Navigating the space of your music," Master's thesis, Massachussets Institute of Technology, September 2008.

[39] D. O. Maidin and M.Fernstrom, "The best of two worlds: Retrieving and browsing," *Proceedings of the Conference on Digital Audio Effects*, 2000.

[40] D. G. Malham and A. Myatt, "3-d sound spatialization using ambisonic techniques," *Computer Music Journal*, vol. 19, no. 4, pp. pp. 58–70, 1995.

[41] T. Marrin, "Toward an Understanding of Musical Gesture: Mapping Expressive Intention with the Digital Baton," Master's thesis, Massachusetts Institute of Technology, Jun. 1996.

[42] M. T. Marshall, N. Peters, A. R. Jensenius, J. Boissinot, M. M. Wanderley, and J. Braasch, "On the development of a system for gesture control of spatialization," 2006.

[43] M. Marshall, J. Malloch, and M. Wanderley, "Gesture control of sound spatialization for live musical performance," in *Gesture-Based Human-Computer Interaction and Simulation*. Springer Berlin / Heidelberg, 2009, vol. 5085, pp. 227–238.

[44] M. Mathews and A. Schloss, "The radio drum as a synthesizer controller," in *ICMC*, 1989.

[45] T. Mizumoto, H. Tsujino, T. Takahashi, T. Ogata, and H. Okuno, "Thereminist robot: Development of a robot theremin player with feedforward and feedback arm control based on a theremin's pitch model," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 2297 –2302.

[46] F. R. Moore, "A general model for spatial processing of sounds," *Computer Music Journal*, vol. 7, no. 3, pp. pp. 6–15, 1983.

[47] A. G. E. Mulder, "Design of virtual three-dimensional instruments for sound control," 1998.

[48] S. Ness and G. Tzanetakis, "Audioscapes: Exploring surface iinterfaces for music exploration," 2009.

[49] B. Nevile, P. Driessen, and W. A. Schloss, "Radio drum gesture detection system using only sticks, antenna and computer with audio interface," 2006.

[50] E. Pampalk, S. Dixon, and G. Widmer, "Exploring music collections by browsing different views," 2003.

[51] I. Poupyrev, R. Berry, J. Kurumisawa, K. Nakao, M. Billinghurst, C. Airola, H. Kato, T. Yonezawa, and L. Baldwin, "Augmented groove: Collaborative jamming in augmented reality," in *SIGGRAPH Conference Abstracts and Applications*, vol. ACM: pp. 77, 2000.

[52] V. Pulkki and M. Karjalainen, "Multichannel audio rendering using amplitude panning [dsp applications]," *Signal Processing Magazine, IEEE*, vol. 25, no. 3, pp. 118 –122, May 2008.

[53] J. B. Rovan, M. M. Wanderley, S. Dubnov, and P. Depalle, "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance," *KANSEI-The Technology of Emotion*, 1997.

[54] J. Salas and C. Tomasi, "People detection using color and depth images," in *Proceedings of the Third Mexican conference on Pattern recognition*, 2011.

[55] M. Schedel, N. Fox-Gieg, and K. G. Yager, "A modern instantiation of schillinger's dance notation: Choreographing with mouse, ipad, kbow, and kinect," *Contemporary Music Review*, 2011.

[56] W. A. Schloss, "Using Contemporary Technology in Live Performance: The Dilemma of the Performer," *Journal of New Music Research*, pp. 239–242, Sep. 2003.

[57] J. L. Sebastian Heise, Michael Hlatky, "Soundtorch: Quick browsing in large audio collections," in *Audio Engineering Society Convention 125*, 10 2008.

[58] J.-H. Seo, H. Shim, and K.-M. Sung, "Artificial reverberator with location control in multi-channel recording," in *Audio Engineering Society Convention 122*, 5 2007.

[59] A. Smirnov, "Music and gesture: Sensor technologies in interactive music and the theremin based space control systems," 2000.

[60] S. Spors, H. Teutsch, and R. Rabenstein, "High-quality acoustic rendering with wave field synthesis," 2002.

[61] A. Tindale, A. Kapur, and G. Tzanetakis, "Training surrogate sensors in musical gesture acquisition systems," *Multimedia, IEEE Transactions on*, vol. 13, no. 1, pp. 50 –59, feb. 2011.

[62] T. Todoroff, J. Leroy, and C. Picard-Limpens, "Orchestra: Wireless sensor system for augmented performances & fusion with kinect," in *QPSR of the numediart research program*, 2011.

[63] G. Tzanetakis, "Musescape: An interactive content-aware music browser," in *In Proc. International Conference on Digital Audio Effects*, 2003.

[64] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, no. 5, pp. 293 – 302, jul 2002.

[65] G. Tzanetakis, "Marsyas3d: a prototype audio browser-editor using a large scale immersive visual and audio display," in *In Proc. International Conference on Auditory Display*, 2001.

[66] G. Tzanetakis, M. S. Benning, S. R. Ness, D. Minifie, and N. Livingston, "Assistive music browsing using self-organizing maps," in *Proc. Int. Conference on PErvasive Technologies Related to Assistive Environments (PETRAE)*, 2009.

[67] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," in *Computer Music Journal*, 2001, pp. 11–22.

[68] D. Wessel, M. Wright, and J. Schott, "Intimate musical control of computers with a variety of controllers and gesture mapping metaphors," in *International Conference on New Interfaces for Musical Expression (NIME)*, Dublin, Ireland, 2002, pp. 171–173.

[69] T. Winkler, "Making motion musical: Gesture mapping strategies for interactive computer music," in *In Proc. Intl. Computer Music Conference*, 1995.

[70] M. Wright, R. Cassidy, and M. F. Zbyszynski, "Audio and gesture latency measurements on linux and osx," 2004.

[71] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, june 2011, pp. 15 –22.

[72] M. J. Yoo, J. W. Beak, and I. K. Lee, "Creating musical expression using kinect," *visualcomputingyonseiackr*, vol. 50, no. June, pp. 324–325, 2011. [Online]. Available: http://visualcomputing.yonsei.ac.kr/papers/2011/nime2011.pdf