

# **The Sonification of Human Motion for Motor Learning**

A Thesis Presented to

The Faculty of the Computer Science Program

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree of

Master of Science in Computer Science

by

Kevin M. Smith


April 2014

© 2014


Kevin M. Smith

ALL RIGHTS RESERVED


APPROVED FOR THE COMPUTER SCIENCE PROGRAM

 5/15/14  
\_\_\_\_\_  
Advisor: Dr David Claveau      Date

 5/15/14  
\_\_\_\_\_  
Dr. Andrzej Bieszczad      Date

 5/15/14  
\_\_\_\_\_  
Dr Peter Smith      Date

APPROVED FOR THE UNIVERSITY

 5-15-14  
\_\_\_\_\_  
Dr Gary A. Berg      Date

**Non-Exclusive Distribution License**

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author/s retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author/s or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

The CSUCI Institutional Repository will clearly identify your name/s as the author/s or owner/s of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

The Sonification of Human Motion for Motor Learning

Title of Item

MSCS Master's Thesis by Kevin M. Smith

3 to 5 keywords or phrases to describe the item

Kevin M. Smith

Author/s Name (Print)

Author/s Signature

May 20, 2014

Date

# **The Sonification of Human Motion for Motor Learning**

by

Kevin M. Smith

Computer Science Program

California State University Channel Islands

## **Abstract**

This thesis examines how sonification can be used to help a student emulate the complex motion of a teacher with increasing spatial and temporal accuracy. As an example scenario, the system captures a teacher's motion in real time and generates a 3-D motion path, which is recorded along with a reference sound. A student then attempts to perform the motion and thus recreate the teacher's reference sound. The student's synthesized sound will dynamically approach the teacher's sound as the student's movement becomes more accurate. Several types of sound mappings, which simultaneously represent time and space deviations are explored. For the experimental platform, a novel system that uses low-cost camera-based motion capture hardware and open source software has been developed. This work can be applied to diverse areas such as rehabilitation and physiotherapy, performing arts and aiding the visually impaired.

The contributions of this work include the following: 1) A methodology for capturing complex human motion in 3-D and mapping the motion to sound for in-context feedback of position and speed; 2) The development of low-cost software system which is being used as experimental platform for the research; and 3) Initial testing of the system using several different types of motion to demonstrate performance.

## **Acknowledgements**

I would like to thank the faculty, staff and students of California State University, Channel Islands for their support and feedback. I would also like to acknowledge the authors and contributors of the *Processing Language* and the *SuperCollider Real-Time Audio Synthesis and Algorithmic Composition* system, which are the key tools used in the development of the framework. My daughter, Kaleigh Smith, took time out from her busy class schedule and was immensely helpful in testing the system. I wouldn't be in the graduate program without the encouragement of Dr. Andrzej Bieszczad who convinced me to apply to the M.S. Computer Science program after taking several influential courses he taught. A special thanks goes to my thesis advisor, Dr. David Claveau who offered guidance, focus and creative collaboration on many of the big ideas in this work including the concept of using sound as a feedback mechanism for learning. Finally, my aspiration to be a graduate student and to follow the path to completion would not have become a reality without the patience, encouragement and thoughtful advice of my wife and soulmate, Norma Camacho.

## TABLE OF CONTENTS

<b>CHAPTER 1: INTRODUCTION .....</b>	<b>10</b>
1.1 Sonification.....	10
1.2 Sonification for Interaction.....	11
1.3 Main Objective .....	12
1.4 Thesis Overview .....	12
1.5 Thesis Roadmap .....	13
1.6 Key Terms for this Thesis .....	13
<b>CHAPTER 2: BACKGROUND AND RELATED WORK .....</b>	<b>15</b>
2.1 Sonification.....	15
2.2 Sound Synthesis Systems .....	15
2.3 Related Work.....	16
2.4 Recent Developments in Motion Capture Technology .....	18
<b>CHAPTER 3: SYSTEM DESIGN .....</b>	<b>19</b>
3.1 <i>SoundTracer</i> – An Overview .....	19
3.2 Workflow.....	20
3.3 Data Flow .....	22
3.4 Motion Path Representation .....	23
3.5 Sonification Process .....	24
3.5.1 <i>The Attack, Decay, Sustain, Release Model (ADSR)</i> .....	26
3.5.2 <i>Sampled Music Feedback</i> .....	27
<b>CHAPTER 4: SYSTEM IMPLEMENTATION.....</b>	<b>28</b>
4.1 Overview .....	28
4.2 General Structure.....	31
4.3 The <i>SoundTracerApp</i> Class.....	33
4.4 Tools .....	35
4.5 Tracks .....	36
<b>CHAPTER 5: SYSTEM TESTING.....</b>	<b>39</b>
5.1 Test Setup .....	39
5.1.1 <i>Sonification and Visual Aid Combination</i> .....	39
5.1.2 <i>The Canonical Moves</i> .....	39
5.1.3 <i>Example Score Card</i> .....	41
5.1.4 <i>Evaluating the Difference using Root Mean Square</i> .....	42
5.1.5 <i>Automatic Generation of 3-D Motion Plots for Analysis</i> .....	42
5.2 Test Results and Discussion .....	45
5.2.1 <i>Semi-Planar Motion – Starting Simple</i> .....	45
5.2.2 <i>Mixed 3-D Motion – Adding Complexity</i> .....	47
<b>CHAPTER 6: ANALYSIS OF RESULTS — SUMMARY.....</b>	<b>56</b>
6.1 Learning Convergence.....	56
6.2 Comparisons Between Multiple Users .....	59
6.4 Timing Comparisons .....	63
6.5 Motion Path Start/End Points .....	64
6.6 Noise in the System .....	64
6.8 User Feedback/Impressions.....	65

**TABLE OF CONTENTS (continued)**

**CHAPTER 7: CONCLUSION AND FUTURE WORK ..... 66**

7.1 Conclusion ..... 66

7.2 Future Work..... 67

    7.2.1 *Development of Manual Skills and Navigation for the Blind* ..... 67

    7.2.2 *Sports Training* ..... 68

    7.2.3 *Gestural Interfaces*..... 68

    7.2.4 *Sound Mappings*..... 69

    7.2.5 *Constraining Degrees of Freedom*..... 69

    7.2.6 *New Motion Capture Technologies* ..... 69

    7.2.7 *Extending to Full Body Motion*..... 69

    7.2.8 *More Elaborate Motion Study* ..... 70

**LIST OF FIGURES**

Figure 1-1. Thesis Roadmap ..... 13

Figure 2-1. Movement sonification of the rowing athlete for motor learning  
from the work of Effenberg et al..... 16

Figure 2-2. Vicon motion capture system for full body motion. .... 17

Figure 2-3. V Motion Project. Kinect-driven visual effects projected. .... 17

Figure 2-4. Microsoft Kinect sensor ..... 18

Figure 3-1. Student (right) reproducing motion stored in track (left)..... 19

Figure 3-2. Student attempts to replicate track. Left, recorded track.  
Middle, student deviates from curve. Right, student follows curve  
within tolerance..... 20

Figure 3-3. System Workflow. .... 21

Figure 3-4. *SoundTracer* shot sequence for spatial testing. .... 23

Figure 3-5. *SoundTracer* shot sequence for temporal testing..... 23

Figure 3-6. Spatial error: Deviations from teacher's path influence "pitch bend"  
parameter in sound mapping..... 24

Figure 3-7. Student moving too fast, ahead of teacher's target point  
(blue marker)..... 25

Figure 3-8. Student is moving too slow, behind teacher's target point  
(blue marker)..... 25

Figure 3-9. Envelope curve types. .... 26

Figure 4-1. *SoundTracer* system consisting of a Processing application  
communicating with capture device (Kinect), OSC control devices  
and the SuperCollider sound server. .... 28

Figure 4-2. iPhone interface for *SoundTracer*..... 30

Figure 4-3. *SoundTracer* general program structure, commands, and tools. .... 31

Figure 4-4. *SoundTracer* application class structure. .... 34

Figure 4-5. Class hierarchy of *Tool* classes in *SoundTracer* ..... 35



**LIST OF FIGURES (continued)**

Figure 4-6. The persistent Track class with the TrackList container and TrackedPoint (based on Vector) classes. .... 37

Figure 5-1. Horizontal planar movement..... 40

Figure 5-2. Vertical *planar movement*..... 40

Figure 5-3. Mixed 3-D movement..... 41

Figure 5-4. 3-D plot of teacher (red) and student (blue) motion paths..... 43

Figure 5-5. 3-D surface plot showing gap between teacher (red) and student (yellow) motion paths..... 44

Figure 5-6. 3-D plot file script generated by *SoundTracer* ..... 45

Figure 5-7. Horizontal motion captured..... 46

Figure 5-8. Learning convergence of horizontal motion, four(4) scores plotted..... 46

Figure 5-9. Mixed 3-D motion captured..... 47

Figure 5-10. Trial 1. Reference motion (red) and student's attempt (blue)..... 48

Figure 5-11. Trial 1. Surface between two paths. Non-uniform tessellation shows that timing is considerably different from reference motion. .... 49

Figure 5-12. Trial 4. Student's motion (blue) is fairly close to reference (red), but improvement not apparent until the surface plot is reviewed..... 49

Figure 5-13. Trial 4 surface plot. Tessellation is more uniformly distributed..... 50

Figure 5-14. Trial 7. Reasonably good match between student's motion and reference..... 50

Figure 5-15. Trial 7. The thin ribbon-like profile of the surface indicates the paths are similar shape. Tessellation is uniformly distributed. .... 51

Figure 5-16. Trial 3. Wide variation between student (blue) and reference path (red) with correction at approximately (400, 300, 1500)..... 52

Figure 5-17. Surface plot. Potting package is not able to tessellate spike in correction properly however, wide variation in motion path shape and timing are clearly shown with non-inform tessellation..... 53

Figure 5-18. Trial 7. Convergence of shape between student (blue) and reference data (red). .... 53

Figure 5-19. Trial 7. Surface plot shows narrower gap between plots, although tessellation still shows significant timing differences. .... 54

Figure 5-20. Trial 11. Closeness of shape between student data and reference. .... 54

Figure 5-21. Trial 11. Plot shows regular tessellation with fairly close execution of timing..... 55

Figure 6-1. Results with sound and visual aids..... 57

Figure 6-2. Results with sound only..... 57

Figure 6-3. Learning convergence for visual aids only..... 58

Figure 6-4. Combined graph of all three combinations of learning aids..... 58

Figure 6-5. Reference Motion used for multiple users..... 59

Figure 6-6. Plot of learning convergence for three users with sound and visual aids enabled..... 60

Figure 6-7. Plot of learning convergence for three users with sound only enabled (no visual)..... 61

**LIST OF FIGURES (continued)**

Figure 6-8. Plot of learning convergence for three users with visual aids only (target markers).....	62
Figure 6-9. Mismatched Timing.....	63
Figure 6-10. Closely Matched Timing.....	63
Figure 6-11. Erratic data caused by noise.....	64
Figure 7-1. Blind woman navigating through a room with the assistance of a dog. ....	67
Figure 7-2. Conductor's baton pattern for $\frac{3}{4}$ time on left, while the right shows $\frac{1}{2}$ time. ....	68

**LIST OF TABLES**

Table 4-1. Commands supported by the iPhone interface.....	30
Table 4-2. List of Visual Affordances.....	32
Table 5-1. Sample score card for two students using three types of motion. ....	41
Table 5-2. Horizontal motion — four trials converging to an acceptable score.....	47
Table 5-3. Summary of seven trials with visual and sound aids active. ....	48
Table 5-4. Table showing 11 trials with sound only aiding the student's motion. ....	52

# Chapter 1: Introduction

Learning a complex motor skill, whether it is related to sports, performance or everyday living, can be a difficult task. Recreating this precise movement requires much practice, even with the assistance of visual aids and coaching. While performing some tasks, it is not always practical to engage the visual senses. For those who cannot see, it is impossible. Sound provides a unique opportunity to convey complex information by leveraging the auditory skills of a person to either augment or replace visual aids. Imagine if we were able to engage another channel of feedback — the use of sound to convey information about motion and assist us in learning how to move in an exciting new and interactive way.

Just as graphics can be used to visualize complex information, sound can be used to help us understand complex data by leveraging the auditory capability of humans. If we see human motion as a complex data set with its natural qualities of position in space over time, it should be possible to use sound to help us understand the qualities of that motion including position and timing information. Taken one step further, the use of sound as a feedback mechanism to help us *learn* how to move is the focus of this work.

## 1.1 Sonification

*Sonification* is a way to convey information about data using auditory means rather than visual. Sometimes called the audio equivalent of data visualization, sonification is a broad multidisciplinary field that brings together components from many fields such as audio synthesis, music, interface design, data mining and psychology. The roots of sonification are in the area of auditory data representation and the use of sound to communicate information to a user to help them perceive complex data. As an example, Frysinger[1] describes the early history up to the 1980's of auditory display with examples in the perception of differentiating complex seismic data with sound and acoustic chromatography for presenting analytical chemistry data to visually impaired students. A system for generating a complex *soundscape* based on live inputs from a stock trading system was studied by Mauney and Walker[2] in which trading performance was enhanced and the user was allowed mobility not offered with a purely visual system.

As we will see in this thesis, sound can also be used to describe characteristics of human motion including both path and timing elements. In these instances, sonification is used to 1) enhance the user's perception of the data; 2) provide instantaneous feedback on the data; and 3) reduce the requirement of user to be visually engaged. The data may be too complex or streaming too quickly to visualize in real time without additional aids, therefore, sound can be used to provide an auditory method to communicate features of the data. It can also provide instant feedback on motion when the performer/athlete cannot access a visual display or if the data is too complex to represent visually. In some scenarios, the user may not be in a stationary position to watch the data or the user may be visually impaired. Sonification, if implemented effectively, could provide a better interface: one in which the user could be given a clearer picture of what's going on in real time without depending on visual feedback.

## **1.2 Sonification for Interaction**

Motion is related to sound on many levels. A dancer performs to the sound of music, a composite layering of many different sounds, weaving complex melodic and harmonic information conveyed to the user. People may react to sound to avoid a potential hazard such as the sound of the car nearing a pedestrian or the response of the pilot to a warning beep on the control panel. Sound can also be generated from motion itself. Think of the sound of footsteps down a quiet hallway or the vibrant percussive sounds from a drummer. In short, sound and motion are related bi-directionally. Motion can be instigated by sound and motion can generate sound in many different ways.

Many human tasks require learning a motion or acquiring a new motor skill to perform an action. Take for example, the swing of a tennis racket or golf club or the freestyle stroke of swimmer. In addition to sports examples, there are more basic forms of motion that we take for granted such as walking or getting up and down from a chair. These tasks mostly illustrate gross motor skills, but fine motor skills, such as movement of the hands — pointing, reaching and grasping — are equally as important. Is it possible that, given the interrelationship of sound and motion, we can use sound in some way to improve one's learning of motion and the motor skill required to produce it?

The possibilities of conveying sonic information related to even simple motion is almost limitless. For example, if a simple motion path is created using the hand, there is a lot of information we can infer from that motion. We have both the 3-D path of travel and the timing of the path or, in other words, the progress of the body along the path of motion at any point in time. In essence, we can regard a space-time picture of what happens to that hand over time.

Given the ability to record the motion and timing as a reference, computing differences instantaneously between the reference motion and a student's motion in terms of space-time variations is possible. Using this information, sound can be used as a real time auditory feedback mechanism for learning the motion. Inaccuracies in recreating the motion could drive changes in the original sound and provide an altered sound as continuous feedback to the student as the move is performed. When the motion is correct or on track, the original sound is unaltered. As the motion deviates from the correct path or timing, the sound is modified proportionally.

## 1.3 Main Objective

Most techniques for learning movement, whether they are used for dance, sports or other articulated motion, are visual. These can include interactive methods such as live demonstration and/or video recording of the movement for playback and coaching. Typically, these methods are predominantly visual with assisted verbal instruction provided by a teacher or coach. For this work, I focus on a different approach: the augmentation of the learning process with layered *sound* to provide an immediate sonic feedback mechanism for learning precise motion in 3-D with a high degree of accuracy. In this approach, it is examined how layered sound can be used to provide *concurrent* feedback on both *spatial accuracy and timing* of the motion. That is the principal contribution of this work. It is my goal that once I solve the atomic problem of a single motion path, the research can be extended in the future to include more complex hierarchical motion containing potentially many motion paths of multiple joints and multiple bodies.

As a byproduct of this work, a software system called *SoundTracer* is developed to enable recording and sound mapping of 3-D human motion with interactive tools provided for learning the motion. The system provides a framework for motion sonification experimentation and motor learning feedback. The system has been informally tested and demonstrated with several users and detailed results of the learning sessions with several different types of motion are presented in Chapter 5, System Testing.

## 1.4 Thesis Overview

Having introduced the topic of sonification of motion for learning, the body of the thesis will focus on the design and implementation of the system, the testing methodology, and some preliminary results.

The design will focus on the overall workflow of the system: how motion gets recorded, sonified and used as a basis for the reference motion for the student. The implementation will look at the detail around how the system is structured, including examples of sonification mappings used. This will also include the overall architecture of the system and the object-oriented class hierarchies developed. In addition, the various components in the system including the sound synthesis system will be discussed.

After the detailed coverage on the system design and implementation, a discussion will follow on how the system was tested. This will include the types of motion tested and how differences in the data set can be visualized from a spatial and timing perspective. Finally, I will conclude my thesis with some ideas and opportunities for future groundbreaking work.

## 1.5 Thesis Roadmap

*Chapter 2: Background and Related Work* will provide an overview of the field, focusing on prior research work in the area of sonification of motion and current techniques for learning of motion skills.

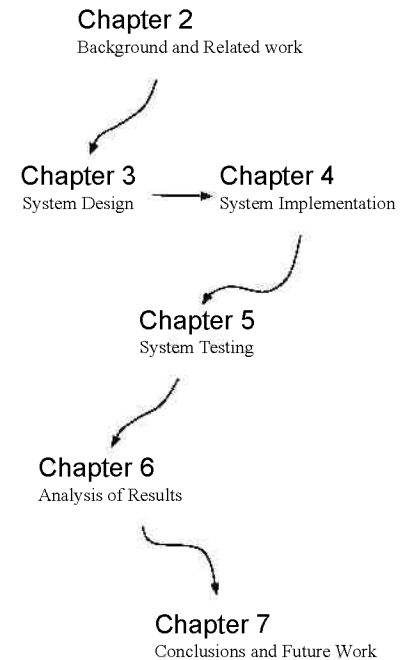
*Chapter 3: System Design* will focus on the design of my experimental platform, which will include the data representation, capture methods and sonification mapping.

*Chapter 4: System Implementation* will cover the details of the implementation of the system, including software components and classes.

*Chapter 5: System Testing* will review the testing method and performance observations to date.

*Chapter 6: Analysis of Results* will analyze the results acquired.

*Chapter 7: Conclusion and Future Work* will summarize observations based on the work so far and propose future directions.



**Figure 1-1.** Thesis Roadmap

## 1.6 Key Terms for this Thesis

**Actor** – A user of the *SoundTracer* system. There are two types of actors, students and teachers.

**Auditory display** – The use of sound to communicate information from a computer to the user (sometimes used interchangeably with *Sonification*).

**Auditory feedback** – Sound used as feedback for the user in order to provide guidance for learning motion. The sound is generated through *sound mappings* described below.

**Auditory graphs** – The auditory equivalent of mapping data to visual plots, graphs and charts.

**Envelope** – A 2-D curve used as input to a function used to process sound. In *SoundTracer*, envelopes are used to control overall energy or volume of the sound over a time interval to emulate the attack, sustain and decay characteristics of a musical instrument.

**Euclidean distance** – The distance between two points in 3-D space.

**Learning convergence** – The number of trials a user must execute before an acceptable level of performance is reached.

- Motion capture – Using a hardware device to capture motion of an actor (human or nonhuman) in real time. The captured data is then usually stored in a computer for further processing.
- Motion path – An internal representation of a parameterized curve in *SoundTracer*, which represents 3-D motion. The curve consists of data points connected by linear segments.
- Motor skill – An intentional movement involving a motor or muscular component that must be learned and voluntarily produced to perform a specific goal or complete task.
- Planar motion – Motion that is confined to a plane. In motion testing for *SoundTracer*, all reference motion is 3-D, but it may have dominant 2-D form such as vertically or horizontally oriented motion.
- Playback – Playback of a pre-recorded track with sound mapping applied.
- Reference motion – Motion recorded by a teacher which is to be copied by the student.
- Reference motion – The motion created by the teacher to be copied or emulated by the student.
- Sonification – The use of sound to convey information about data.
- Sound mapping – A mapping from data to a synthesized sound. The mapping is dynamic in that the sound can change with variation in the data over time.
- Spatial accuracy – Accuracy in reproducing motion with respect to shape or form. The level of accuracy varies with the distance from the reference curve being reproduced.
- Temporal accuracy – Accuracy in reproducing motion with respect to time. Reference motion can be stored with timing information. Variation from the timing in the reference motion affects the timing accuracy.
- Track – An object which represents a motion path with associated timing information at each data point. Tracks can be stored on the disk.
- Visual affordance – A visual aid on the screen, such as a marker or guide, used to assist the user in performing an action. For the purposes of this work, an affordance can be interactive or just a simple indicator.

## Chapter 2: Background and Related Work

### 2.1 Sonification

*Sonification* is the use of sound to communicate information from a machine or computer to a user. It is a very broad and multidisciplinary field that has roots in the area of auditory displays. Machine examples include cockpit systems such as stall warning indicators that alert the pilot when the aircraft is about to stall. Another example from the computer area are *earcons*[4], short sounds that signify some event or action. Computer examples include the many sounds that an operating system makes when different events occur, such as startup or shutdown or when an error occurs. Product design has benefited from the use of simple, effective sonification for providing intuitive interactive feedback. The iPod click wheel patented by Apple Inc. is one example of this. Most electronically controlled home appliances benefit from some sort of auditory display for indicating activity or start/end cycles.

More elaborate methods of sonification have been used to visualize complex data sets such as weather phenomenon or other natural phenomenon[5]. *Auditory graphs*, which are the sonification equivalent of visual plots, graphs and charts has its roots in Mansur et al[6]. The evolution of computing hardware along with the field of audio synthesis has made it possible to map sound to complex data sets and hear it in real time

### 2.2 Sound Synthesis Systems

The science of audio sound systems are well covered in texts as exemplified by Cook[7]. In this work I will focus on the application of sound synthesis systems, which have evolved from the theory of audio synthesis as a basis for implementation. Software sound synthesis systems, which include programming languages for synthesizing audio and sound generation, is a mature area with several high quality contributions that have become standards in the research area.

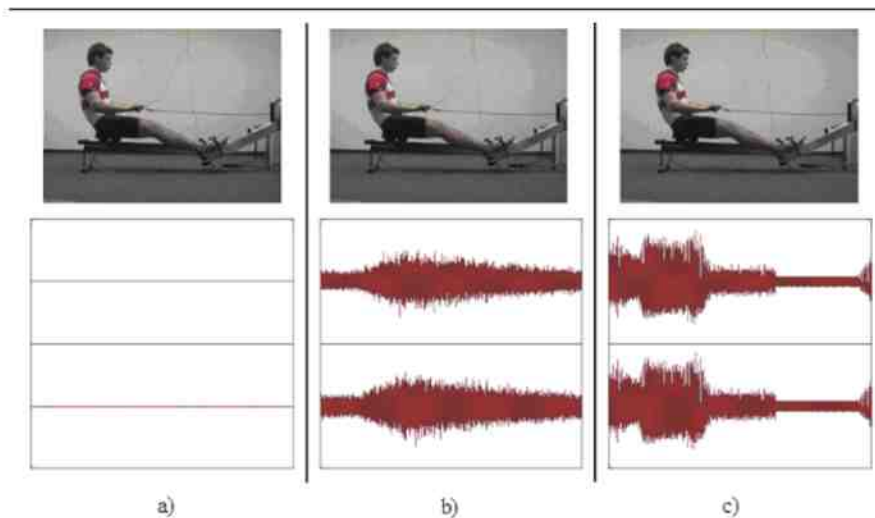
The SuperCollider[8] system, which was chosen for this work is a very powerful system which provides an object-oriented programming language for creating algorithmically-generated sound. The SuperCollider sound server architecture provides a flexible channel for communicating with devices and other programs. The *sclang* programming language for SuperCollider provides virtually unlimited capability for sound generation, layering and envelope control over sound. It will be shown that this is an important capability for the research covered in this work.



Another newer system which has been gaining momentum was evaluated for this project. The ChuckK[9] audio programming language developed by Ge Wang allows on-the-fly programming while the program is running, which can benefit live performance. ChuckK is gaining popularity among artists and sound designers who use *live-coding* as a performance technique where sound is created and modified live during a performance programmatically, and in some cases improvisationally.

## 2.3 Related Work

Existing work in the area of aiding movement by the use of sonification focuses on a number of different topics. In the work of Rober and Masuch[10], the use of interactive auditory environments and 3-D sound rendering to explore virtual auditory environments is the focus in the design of a framework. In the area of motion, Effenberg et al[11][3] used sonification to assist in the reproduction of human movements and acquisition of motor skills, showing that sound can provide additional information in the accurate reproduction of jumps and other athletic movements. In particular, the topic of auditory feedback for athletic performance in rowing, as one example, is investigated in Effenberg, Fehse and Weber[3] as illustrated in Figure 2-1.

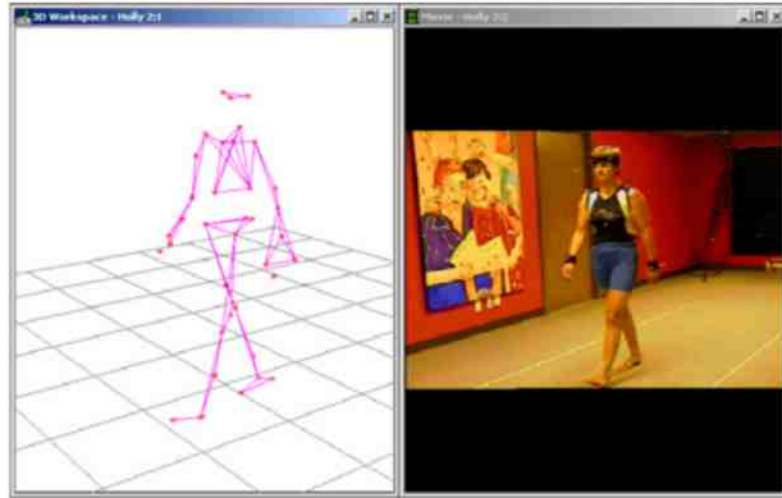


**Figure 2-1.** Movement sonification of the rowing athlete for motor learning from the work of Effenberg et al.

Other work includes the use of sound for physiotherapy. Feedback is an area studied by Pauletto and Hunt[12]. In their work, the sonification of EMG signals gathered in a clinical environment provide auditory display to a therapist in real time, producing sound with muscle movement that is audible in the room when visual displays are not always within view. PhysioSonic[13] was developed as a system to map motion capture data to sound to provide auditory feedback for physiotherapy and training.

In addition to feedback, there are existing projects using real time articulated motion for *generating* sound and music. In these interactive performances, performance gestures are translated to music and motion graphics, allowing the body to generate sound and visual effects. *Synapse*[14] and *The V Motion Project*[15] are two such example projects

which both use the *Kinect* device (see Section 2.4) for capturing the motion. Figure 2-2 shows the Vicon motion capture system used for capturing full-body motion.



**Figure 2-2.** Vicon motion capture system for full body motion.

Kapur et al[16] focused their work on building a framework for the sonification of Vicon[17] motion capture data showing that parameters of motion could be mapped to sound and synthesized to the ChuckK language for various applications including musical instrumentation and motion analysis.

The *V Motion project* utilizes motion capture based visual effects and music projected in an outdoor environment. A Kinect-driven visual effect for V Motion is shown in Figure 2-3.



**Figure 2-3.** V Motion Project. Kinect-driven visual effects projected.

## 2.4 Recent Developments in Motion Capture Technology

There is a limited amount of work related to the sonification of motion in real time. The technology for capturing human motion in real time (i.e., motion capture systems) has become a mature but expensive technology in the entertainment industry for CGI films and games, and the connection between motion capture and sonification has not been explored that deeply.

Traditional optical motion capture systems such as those used in the film *Avatar*[18] provide the best resolution and frame rates for capturing human motion in real time. These systems are typically very expensive with entry-level prices in the 12K USD range[17] with typical installations costing in the hundreds of thousands or more, excluding facility costs. These systems also require a space or *stage* (nomenclature for capture area in the motion capture industry) where the cameras can be set up for a period of time to be calibrated and left untouched. Most optical systems also require markers to be placed directly on the actor or on a special body suit.

With the advent of relatively new and inexpensive technologies such as the Microsoft Kinect (2010)[19] and Leap Motion Leap (2013)[20], it is now possible to capture elements of human motion in 3-D and in real time on relatively low-cost hardware in a semi-portable environment using a laptop. With the fast evolution of this hardware and the consumer demand for games and applications that use them, I expect full-body skeletal-capture capabilities to evolve rapidly in terms of performance and accuracy. The sale of popular video games such as Kinect Sports which sold 5.8M units as of February 2014[21] is a prime example. Figure 2-4 shows the relatively low-cost Microsoft Kinect camera used to capture full-body motion.



**Figure 2-4.** Microsoft Kinect sensor

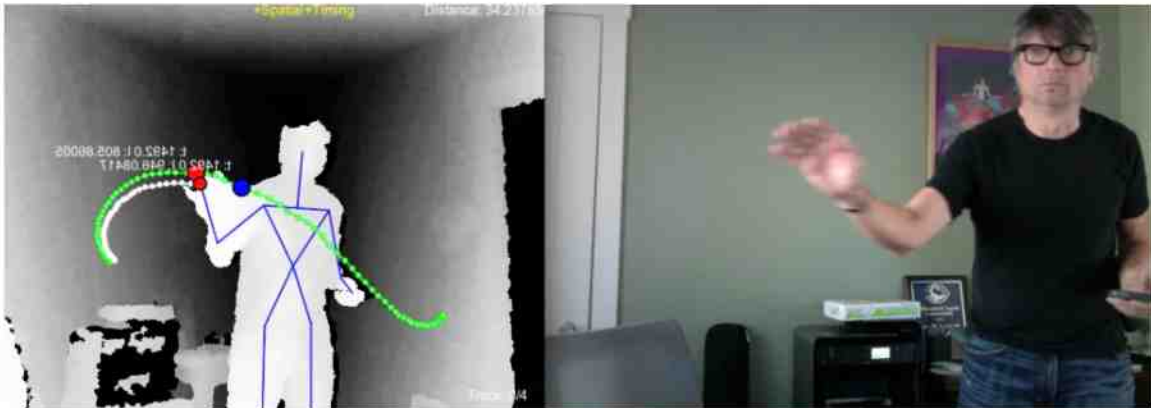
Building on this existing work, the focus of this thesis specifically looks at sonification feedback for learning precise motion along a path with spatial and timing accuracy. The development of a portable laptop system that uses low-cost consumer capture devices will also be a focus of the work. In Chapter 3 of this thesis, the overall system design is described, which includes the internal data representation of the 3-D motion path with timing, sound mappings and synthesis. Chapter 4 follows with a description of the actual implementation of the experimental system, *SoundTracer*. Following that, Chapter 6 will look at some of the initial results of using my system and finally Chapter 7 presents conclusions and opportunities for future work.

## Chapter 3: System Design

### 3.1 SoundTracer – An Overview

As part of this research, I have developed a new system for exploring motion sonification and learning called *SoundTracer*. Using this system, a teacher can create a 3-D motion path in real time along with a sound track. The motion paths are stored persistently as tracks with sound. In this manner, a library of different types of motion can be stored. The student can then attempt to reproduce the motion path using sound as feedback. When the student's motion approaches the motion of the teacher's, the feedback sound will also approach that of the teacher's. Any deviation from the path in timing or spatial error will produce corresponding changes in the sound generated.

Figure 3-1 shows a *SoundTracer* session. The left image shows the Kinect camera view of the depth map image of the scene which is shown in the right using a second video camera. The student (me) is attempting to reproduce the motion path drawn in green with correct accuracy and timing. The motion is being captured and sonified in real time. Spatial differences from the trial curve (white) and timing differences from the student's point (red) and the target point (blue) are mapped to changes made to the sound in order to provide auditory feedback for the student.



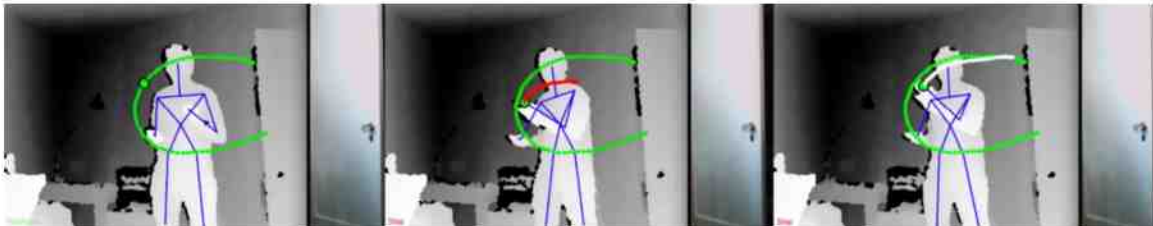
**Figure 3-1.** Recreating a motion path with *SoundTracer*.

## 3.2 Workflow

In the workflow of the system, there are two primary actors<sup>1</sup>, the teacher and the student[22]. The teacher creates a movement that is captured and recorded in 3-D space in real time. The student then has the goal of learning the motion created by the teacher by performing the motion as accurately as possible. A successful performance is determined by the ability of the student to reproduce the motion of the teacher as accurately as possible with the same trajectory and correct timing.

The principal goal of the system is to use sound as a feedback mechanism to assist the student in learning the motion. When the teacher creates the motion, *sound* is automatically generated for the reference motion and stored. When the student attempts to reproduce the motion, sound is also generated. By reproducing the same sound as the teacher, the correct motion is recreated. Any deviation from the correct motion will produce a corresponding change in the sound generated by the student. How the sound is generated or mapped from the 3-D motion path over time is described in Section 3.4.

In Figure 3-2 the student attempts to reproduce the teacher's motion path. The left frame shows the teacher's curve (green trace). As seen in the middle frame, the student's motion (red trace) is too far from the original curve. In the right frame, the student's motion (white trace) is close to the original curve and is within tolerance.



**Figure 3-2.** Student attempts to replicate track. Left, recorded track. Middle, student deviates from curve. Right, student follows curve within tolerance.

---

<sup>1</sup> The term *actor* is used in the context of a UML (Unified Modeling Language) use-case.

For the following description of the recording and learning process, refer to Figure 3-3.

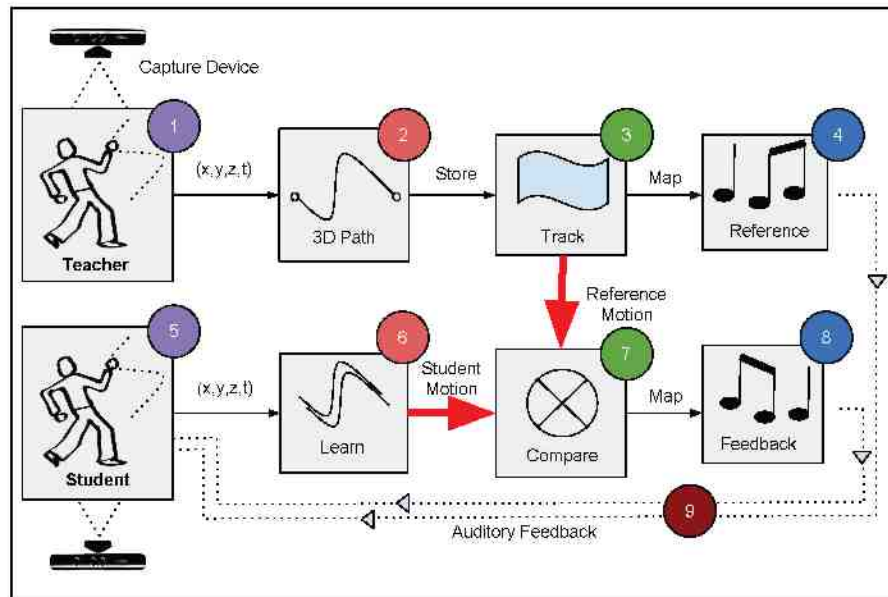


Figure 3-3. System Workflow.

At the recording session, the system captures the teacher's motion in real time using the Kinect device (1). An internal representation in the form of a 3-D motion path is stored (2). Sound is generated to accompany the motion (3). The path and the sound are stored in a track (4), which can be recalled for playback.

At the learning session (which is separate from the recording session and can be done at another time), the student's motion is captured in real time using the Kinect device (5). The student's motion is similarly converted to a motion path in a track (6). The motion is compared in real time with the motion the teacher previously generated in the reference track (7). Trajectory and timing differences between the student's motion and the teacher's are used to modify the sound (8). The new sound is provided as feedback to the student to aid in making corrections to the motion (9).

The *teacher* creates a movement that is captured and recorded in 3-D space in real time. The *student* then has the goal of learning the motion created by the teacher by performing the motion as accurately as possible. What is meant by *performing* is the ability to reproduce the motion of the teacher as accurately as possible to achieve the same trajectory of motion in both space and time.

### 3.3 Data Flow

Motion is captured in real time from a 3-D tracking device. At a capture rate of 30 Hz, a stream of sequential 3-D points for the capture of one joint is obtained in the form of:

$$P = (x, y, z, t) \quad (3.1)$$

Where  $P$  represents a single point in data stream at time  $t$ .

The 3-D motion path is represented by the stream of points which constitutes a piecewise linear approximation of a curve. From this curve, features such as the location on the path at any point in time, the distance to the nearest point on the curve from any point and the length along the curve at any point by can be accessed by simple numerical methods. It is worth mentioning that a more mathematically complex (and computationally more expensive) spline representations of the motion path was considered; however, it was determined that because most devices are capable of returning points at 30 Hz or more, the density of the data was sufficient to approximate the paths at typical move durations.

As shown in Figure 3-3, a 3-D path generated by the teacher is stored in a *track*. These tracks are persistent objects which can be stored on disk for later retrieval by the student. The same mechanism for generating a motion path is used by both the teacher and student. Once the data is stored in a track, it is processed, so that when the track is *played back*, automatically generated sound will accompany the motion. This sonification process will be discussed in more detail in Section 3.5.

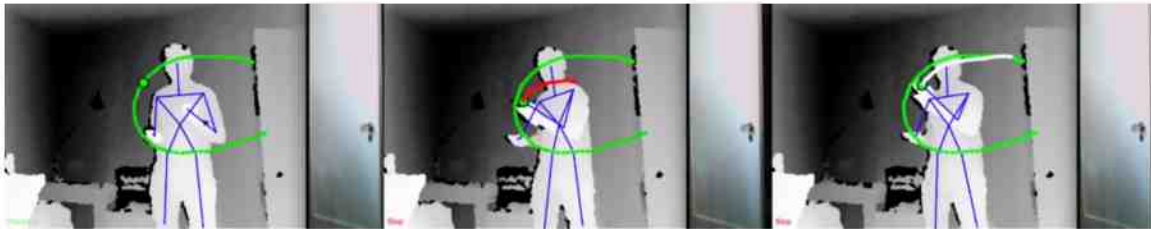
After a teacher records a path to be learned in a track, the student can retrieve this path and initiate a learning session. The learning session enables real time capture and comparison of the student's motion path with the teacher's. A comparison is made both spatially and temporally. The student has the option of obtaining feedback on each component independently or concurrently. For the spatial comparison, the distance between the student's current point  $(x, y, z)$  at any point in time is compared with the nearest point on the teacher's path. The distance between these two points determines the amount of spatial error present at any time. This error can be used as an input to the sonification of the student's motion. If there is no error (within a preset tolerance) the sound is not modified.

For the temporal comparison, the progress along the motion path by the student in terms of curve length is calculated at the current time,  $t$ . If the student is ahead of where the teacher should be (the curve length is longer), then the student is moving too fast and the difference in progress is propagated to the sonification process and the feedback sound can be modified. Conversely if the curve length is shorter, then the student is behind the teacher and moving too slowly and that difference is also propagated. If the progress along the path is the same as the teacher's (within a preset tolerance), the sound is not modified.

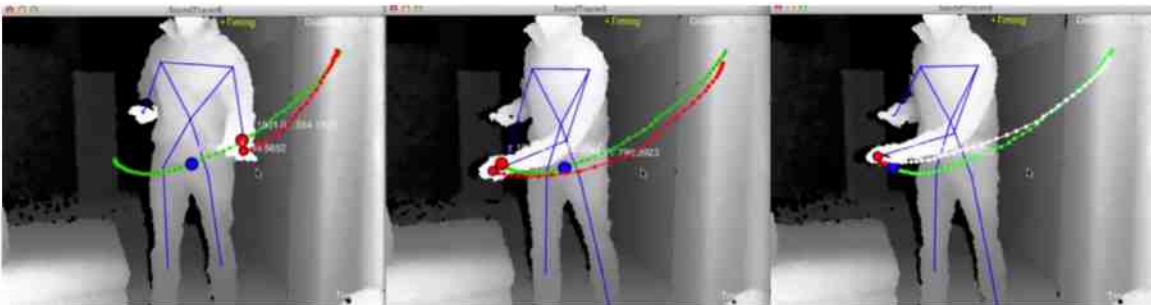


### 3.4 Motion Path Representation

As discussed in Section 3.2, the purpose of *SoundTracer* is to provide features for the real time capture of 3-D motion, the storage of this data in the form of tracks, sonification of the 3-D motion data, and the real time comparison of this data with student attempts to recreate the motion. Figure 3-4 and Figure 3-5 picture sequences show examples of a sample motion path and how spatial and temporal testing can be accomplished. Both can be done individually or concurrently. Visual indicators for the motion path, target point, distance to target, and color coding for the student's path to indicate on-target proximity is provided.



**Figure 3-4.** *SoundTracer* shot sequence for spatial testing. The first frame shows the reference motion (green) created by teacher in green. The middle frame shows the out-of-bounds motion path in red. In the last frame, the motion is shown as white as it correctly approaches the reference motion and stays within tolerance.



**Figure 3-5.** *SoundTracer* shot sequence for temporal testing. In the first frame, the student is early and behind target point. In the middle frame, the student is too fast and arrives at the end of the path ahead of target point. The final frame shows the student is on track to arrive at the same time as the target.

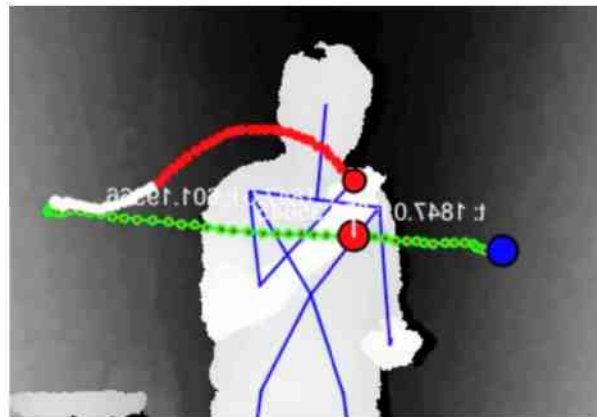


### 3.5 Sonification Process

In designing the auditory feedback for the learning process, two primary goals were established. The first goal was to provide a way to simultaneously allow the student to correct for both spatial accuracy and timing accuracy concurrently. The system should provide feedback that would enable the student to make real time corrections to both the path of travel and the progress along that path with respect to time. For the timing aspect the system should be sensitive and provide feedback to changes in rate of progress (acceleration) and speed (velocity) over the trajectory. This may be challenging for the type of articulated motion present in many applications (performance arts, sports, rehabilitation, or therapy), which can have a very wide range of space and time characteristics. A slow coordinated movement over several seconds would have different sonification requirements than a fast movement, such as a golf swing, which has a short duration. The goal is to design a mapping that will help with both slow and fast motion. A secondary goal is the aesthetic quality of the sound. With an objective of serving artists and performers, the sound should be as pleasing as possible.

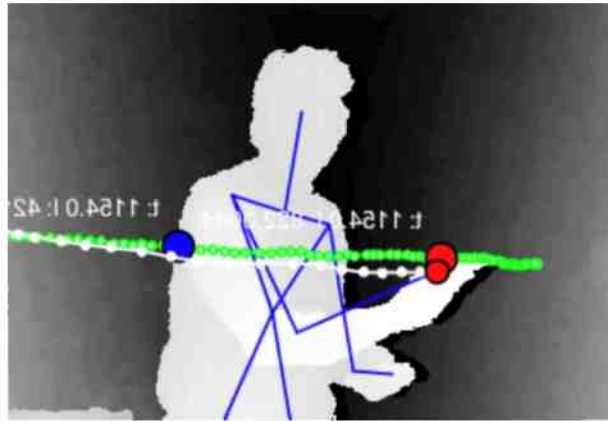
Each  $(x,y,z)$  coordinate is captured in real time at frame rate of 30 Hz and saved in the track along with a time stamp  $t$ . The curve is stored as a piecewise linear representation and approximations of the curve length between two points or times on the curve can be quickly evaluated.

For spatial accuracy, the Euclidean distance is calculated in real time between the student's current point and the closest point on the teacher's path. Any differences in the distance can be used as an input to the sound mapping. In Figure 3-6 (from the prototype), the spatial error (shown in red) is used to generate a "pitch bend" to vary the pitch of the teacher's sound being generated. When the motion is accurate, the pitch bend value is effectively 0 and the student's sound will be the same as the teacher's.



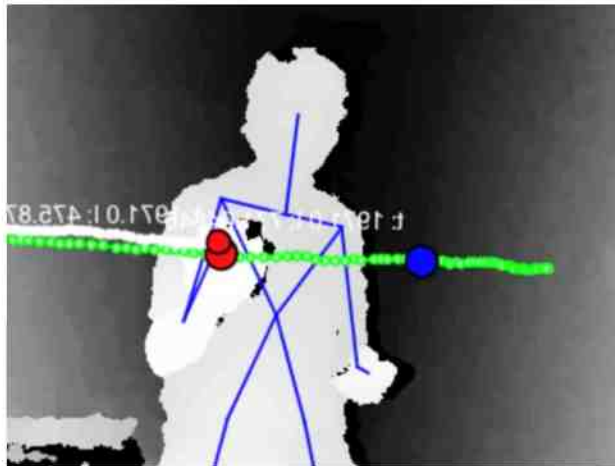
**Figure 3-6.** Spatial error: Deviations from teacher's path influence "pitch bend" parameter in sound mapping.

For temporal accuracy, the student's current time and position is compared with the on-target position of the teacher's at the same time. If the student is ahead of the on-target position (shown in Figure 3-7), the time and distance error is provided as an input to the sound mapping.



**Figure 3-7.** Student moving too fast, ahead of teacher's target point (blue marker).

Similarly, if the student is moving too slow and is tracking behind the teacher's target point, error parameters are provided to the sound mapping (see Figure 3-8).



**Figure 3-8.** Student is moving too slow, behind teacher's target point (blue marker).

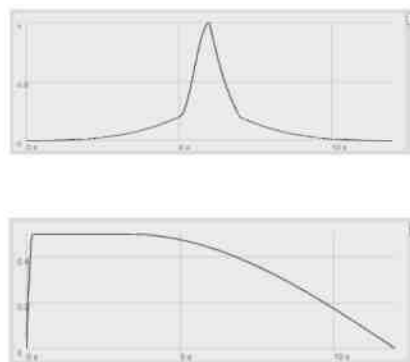
For the sound mapping, a novel approach was developed using layering to add an additional component to the sound to provide feedback on the timing. Taken from the sound synthesis field[23], an attack, decay, sustain, release (ADSR) envelope is used to control the volume of the sound output as the motion progresses along the curve. For the initial tests it was found that a bell curve worked well although any envelope curve could be used.

### 3.5.1 The Attack, Decay, Sustain, Release Model (ADSR)

To incorporate timing and spatial information in auditory feedback for fast motion the concept of an *envelope* is used. By modeling the *attack, decay, sustain, release envelope (ADSR)* used in electronic instrument synthesis, a control envelope is applied as a multiplier to the overall output or volume of the sound being generated. In a real instrument such as a guitar, time is a linear quantity and the ADSR progresses from the time the instrument is plucked to the time the vibrating string stops moving. In this model, an ADSR envelope is generated to match the teacher's duration of the recorded motion. When the student attempts to reproduce the motion, their progress along the path with respect to time actuates the ADSR. As an example, using a simpler ADSR similar to a stringed instrument, if the student is initially too fast, the attack component of the sound will come sooner. If the student is late in the second part of the move, the sound will sustain longer.

ADSR incorporates timing information into the model, but there also needs to be a way to provide corrective feedback for spatial variances. For this approach a simple pitch shifting method is used. When the student deviates from the prescribed path, the pitch will change proportionally to the distance from the location on the path where the student should be at that point in time. The combined layering of the ADSR as envelope for volume and the shifting of the pitch provide two concurrent degrees of freedom for auditory feedback. The effect of the ADSR will depend on the type of envelope chosen and the source sound that it will operate on. In the test cases, the sound of a synthesized flute was used.

Figure 3-9 shows two envelope examples that were used in experiments. In initial testing, it was found that a bell-shaped curve worked the best. Since the rise in the energy of the sound comes roughly half way through the motion, it was easier to learn the timing of the motion based on mental correlation on where the midpoint of the sound should be with respect to the midpoint of the motion.



**Figure 3-9.** Envelope curve types. *Top:* An ADSR with a bell-shaped curve. *Bottom:* An ADSR with early attack and longer sustain.

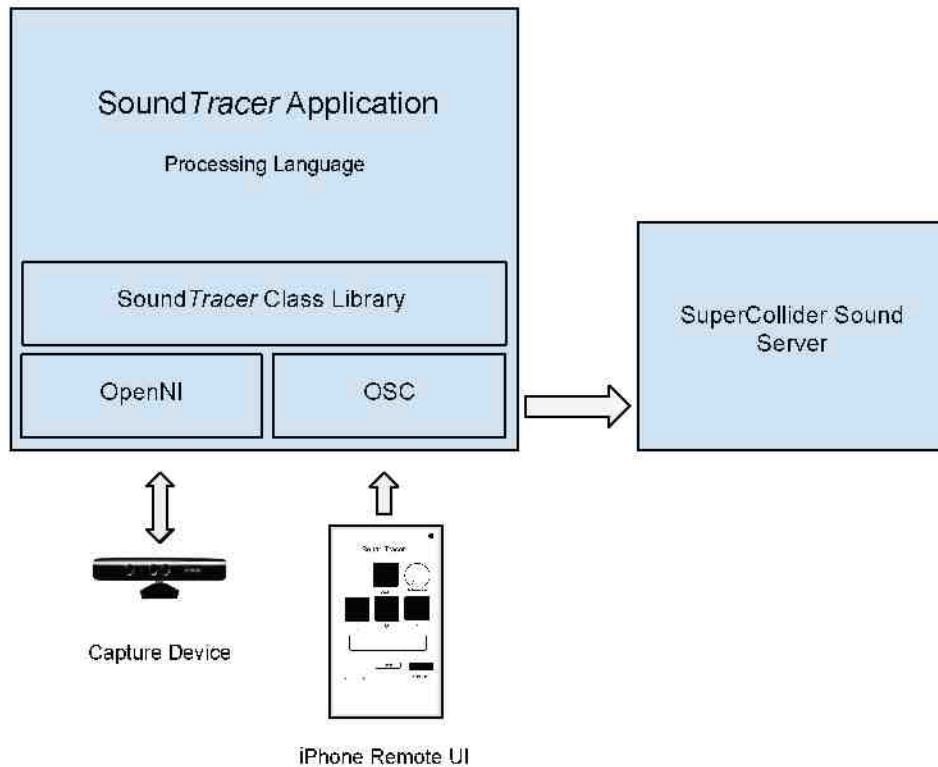
### 3.5.2 Sampled Music Feedback

With ADSR envelopes, an approach was employed that would enable layering of spatial and temporal information to provide auditory feedback for fast movement. To explore the approach of using a pre-recorded musical *sound track* for providing auditory feedback, a *sampled music* approach was also researched; this could be particularly helpful for slower, more performance-oriented movement such as ballet dancing. In this approach the teacher performs the move to sampled music, rather than using a synthesized single instrument with an ADSR. The goal of the student is to reproduce the move and the music at the same time. Any deviations to the motion path will cause a pitch shift in the music. Any deviation to the timing along the path will cause the tempo of the music to change. This method, again, provides us with two degrees of freedom simultaneously for auditory feedback on spatial and temporal accuracy.

# Chapter 4: System Implementation

## 4.1 Overview

The *SoundTracer* system enables a student to practice and reproduce real time motion paths created by a teacher. The overall design concept was presented in Chapter 3. In this section, I will go into more detail on the actual design and implementation of the code. A block diagram of describing how the system devices communicate is shown in Figure 4-1.



**Figure 4-1.** *SoundTracer* system consisting of a Processing application communicating with capture device (Kinect), OSC control devices and the SuperCollider sound server.

*SoundTracer* is based purely on open-source software components. Processing[24] was chosen as the implementation language because of its strength as a rapid prototyping language and the out-of-the-box experience, enabling it to be installed and set up quickly with good library support for communicating with other devices and servers.

Although the code is generic enough to accept data from any 3-D tracking device, the first generation *Microsoft Kinect* device was used along with public domain libraries for the *OpenNI*[25] driver. The *Kinect* is limited in resolution and tracking stability in comparison to a professional optical motion capture system, however, the cost and convenience of this device made it practical for testing in most environments. In addition to the *Kinect*, some limited testing was done with the *Leap Motion* device[20], which provides smaller scale (but more precise) tracking for finger motion.

The generation of synthesized sound for audio is a very mature field. Rather than implement a software synthesizer from scratch, the *SuperCollider*[23] language and sound server was used. The flute sound, used for the motion tests, was generated using a waveguide flute example[26] *SynthDef* in the *SuperCollider* language. The ADSR envelope was implemented in the same language using an envelope generator, which can poll values by index (see *IEnvGen* in the *SuperCollider Reference*)[27].

In order for *SoundTracer* to communicate with *SuperCollider*, an open-source Processing library, *SuperCollider Client*[28], which supports the Open Sound Control Protocol (OSC)[29], was used. The dual benefit of this is: 1) the *SuperCollider* server uses OSC as its native protocol to receive messages from its language module and 2) OSC can serve as a channel for communicating with *SoundTracer* using other mobile devices such as the iPhone, which has a number of customizable OSC-based apps available.

During the course of initial testing, I determined that it was necessary to have a remote capability for controlling the user-interface. Although gaining some additional level of fitness from the exercise of running between testing motion in front of the camera and operating controls on the computer, the inefficiencies required a better solution. To address this problem, I built an iPhone custom user-interface on *TouchOSC*[30] to control *SoundTracer* remotely creating a very convenient interface while motion testing.

Figure 4-2 shows a photo of this interface with transport controls shown. Most features of the application, including record, playback and learning modes can be controlled directly from the touch screen of the iPhone display.



Figure 4-2. iPhone interface for SoundTracer.

Table 4-1 lists the available iPhone commands and operations. Because the iPhone app is based on OSC, a protocol that can operate over a network, the iPhone can control the application running on the computer as long as it is within range of the same wireless local area network the computer is connected to. All the commands available on the iPhone app are also available as keyboard shortcuts on the SoundTracer application running on the computer. In the remainder of this chapter, I will discuss the components and classes structure of SoundTracer in more detail.

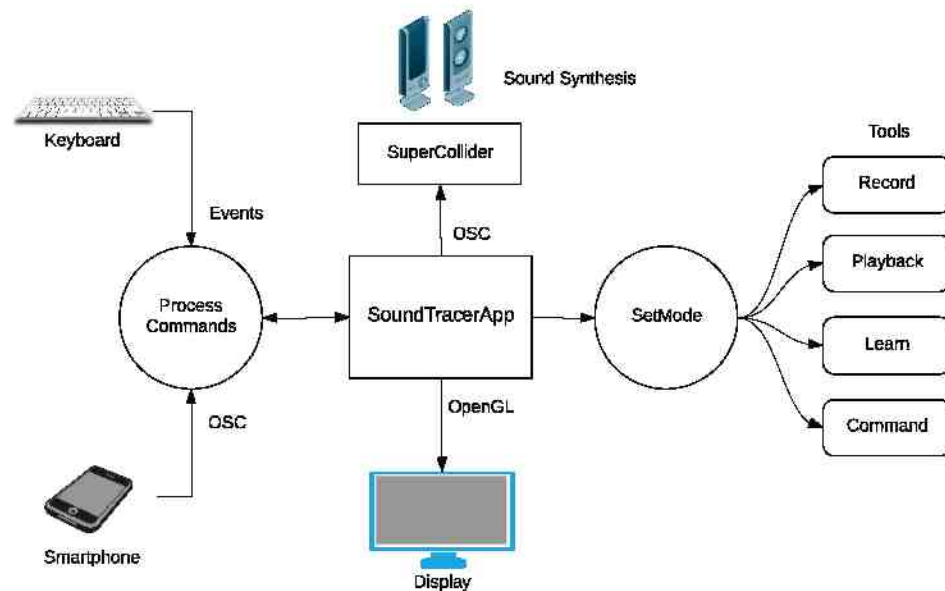
Table 4-1. Commands supported by the iPhone interface

iPhone Command	iPhone Operation
Record	Start recording of a new track
Play	Playback recording in real time with sound
Learn	Start learning mode with sound
Next track	Go to next track
Previous track	Go to previous track
Clear	Clear tracks

## 4.2 General Structure

The general structure of the application, which is written in *Processing* is shown in Figure 4-3. The primary functions of the application class, *SoundTracerApp* are: (a) process incoming commands from the user from either a remote device, such as an iPhone, or directly from the keyboard console; (b) generate sound by sending requests to *SuperCollider*; (c) provide a vocabulary of interactive tools for the user; and (d) provide visual display in an OpenGL window on the primary computer, which can be either a laptop or workstation. Because of the performance requirements for real time video and depth-map processing and the associated cost of processing 3-D input from the user, a workstation-class machine provides a more stable platform and a guaranteed frame rate; however, a laptop will work for testing smaller data sets and demonstrations.

The tools are implemented using a class interface (described in Section 4.3) and are designed to be extensible, so potentially more tools could be added beyond the basic set which has already been implemented. Once a tool is activated (by a command which calls *SetMode*), the selected tool controls the main event loop of the program and all computation is then processed through that tool. The base tool, *Command*, which does nothing but accept commands, can be considered to be the *home* tool from which other tools can be activated. In many drawing or paint programs, this can be compared to the base selection mode, which would typically just show an arrow cursor. Figure 4 3 below diagrams the program structure, commands, tools, and flow.



**Figure 4-3.** *SoundTracer* general program structure, commands, and tools.



Incoming commands are processed by the *SoundTracerApp* in the main event loop of the program. These commands can come from two possible sources: the main computer or a remote device such as an iPhone. The latter uses OSC (Open Sound Control) to communicate with *SoundTracer*. The main event loop has the ability to process incoming commands from OSC using the OSC API library, *oscP5*[31]. The main advantage of using OSC is that many other applications and hardware devices on the market can “speak” OSC protocol. This standardization and availability of third-party devices makes it easy to extend *SoundTracer* to potentially use other input devices such as an iPad or other control surfaces that can generate OSC either on their own or through a custom application based on TouchOSC similar to the one I have provided for the iPhone. Commands initiated directly from the main computer can be in the form of keyboard shortcuts or command strings, which can potentially come from a command or scripting language. As of first publication, only keyboard shortcuts are supported.

Sound synthesis is provided by the open-source *SuperCollider* system. *SoundTracer* needs only to determine which synthesizer to use, what the parameters are (e.g., frequency, duration, envelope), and to send a command to the SuperCollider server to initiate the sound. All SuperCollider commands are issued via OSC. Since *SoundTracer* already uses the OSC API library for the remote control interface, the additional use of OSC for controlling sound incurs no additional code overhead.

The graphics requirements for SuperCollider are straightforward. A video reference of the camera image must be displayed in a window so that the teacher or student can see their motion while performing. In addition, there are various visual affordances, which are displayed using OpenGL for the motion path itself, including target markers and numerical displays for parameters such as depth and program mode. The following table lists the visual affordances, which are implemented.

**Table 4-2.** List of Visual Affordances

Visual Affordance	Implementation
Motion Path	Curve representing target motion path.
Learning Path	Curve representing student's progress.
Closest Point Marker	Marker showing closest point on target path from student's hand.
Target Marker	Marker on the Motion Path where the student should be at any point in time.
Numeric Output	For distance along path for the markers.
End-effector Marker	Marker for the particular end-effector being tracked (e.g., hand).
Curve Start/ End Indicator	Indicator circles flash over the curve start/end point when student's motion comes within a threshold distance.

### 4.3 The SoundTracerApp Class

The overall class structure of *SoundTracer* is shown in Figure 4-4. The *SoundTracerApp* (henceforth “the App”) class is the main application class and holds all of the state data for the running program. In addition to this function, this class is responsible for creating and initializing the Tools (described in Section 4.4), the *TrackList* (described in Section 4.5); and establishing connections to the sound server, *SuperCollider* using OSC. This protocol is also used to connect with external remote devices, such as the iPhone previously described. The App class also includes methods for processing and executing commands and setting the active tool.

In addition to the data management function, the App class interfaces with the motion capture device being used, which in this implementation is the Kinect. Through an OpenNI interface library[32] for the *Processing* language, the App class can initialize the Kinect interface, calibrate the skeleton, and retrieve 3-D joint positions from the subject in real time. The API also supports real time video display of the depth map and RGB images taken by the IR (infrared) depth sensor and video camera, respectively. For more information on the Kinect camera components and related hardware, please see the Kinect Sensor Component Specification.[19]

The architecture of *SoundTracer* does not depend on the Kinect device, solely. It only needs to be able to capture 3-D body/joint positions over time and any device that is capable of doing that could, theoretically, interface with *SoundTracer*. As a preliminary test, the Leap Motion device[20] which can focus on precise hand and figure motion for gestural interfaces, was interfaced. Although more work needs to be done to solidify, a working prototype was developed in less than an hour with the existing system.

The top level *SoundTracer* application class structure is diagrammed in Figure 4-4. The App class maintains the global state of the program, the interactive tool list and the track list. In addition to this data, the App class provides an interface to the Kinect device, the sound synthesis server (*SuperCollider*) and the OSC interface. These components are described in detail in the following sections.

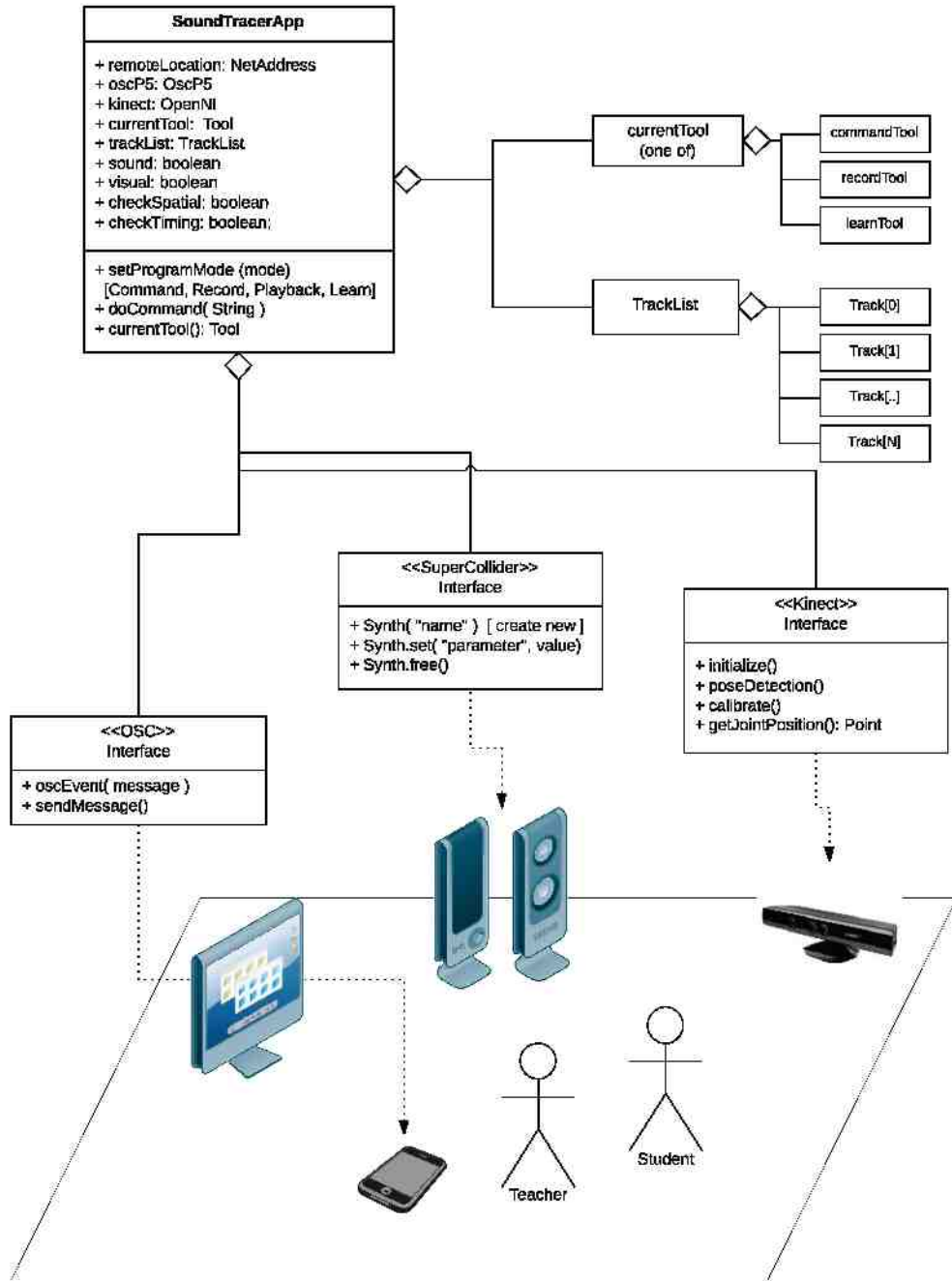
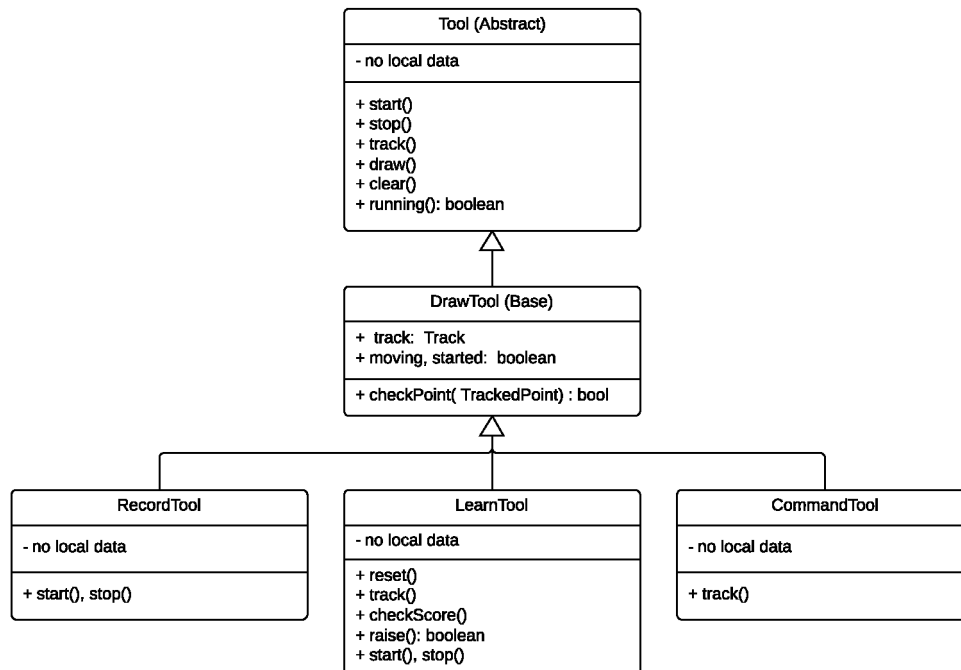


Figure 4-4. SoundTracer application class structure.

## 4.4 Tools

The *Tool* base class provides an interactive context in *SoundTracer* to provide the interface between the user and rest of the system. All interaction classes are derived from *Tool*. The *RecordTool*, *LearnTool* and *CommandTool* respectively facilitate the three primary modes of the application: 1) recording and playback of new tracks; 2) learning the motion stored in a track; and 3) command processing and general feedback in the viewport.

Please refer to Figure 4-5 for the following discussion on the *Tool* class hierarchy. The *Tool* base class is an abstract class only. It is never instantiated by itself; it is used primarily as a base class to inherit from when a new tool is built. The *Tool* class provides all of the common methods for providing the most basic interaction with the system. These include starting and stopping of the tool, any special drawing operations the tool may need to do, and test functions to indicate if a tool is running or not. There is a special method called *track()*, that enables the tool to track state at each iteration in the main event loop. For tools that draw or learn, there may be checks that are done at each call made to *track()* that perform some operation, comparison or computation on the data. For example, a recording tool might check to see if a new incoming 3-D point is valid and store it in the current track. A learning tool might perform some comparisons on the student's data at each time step with the teacher's data stored in the *Track* that is being learned.



**Figure 4-5.** Class hierarchy of *Tool* classes in *SoundTracer*

The tool used for creating a new motion path and storing it in a track is the *RecordTool*. The recording tool simply records the user's data into a track. Typically, this will be a new track that is recorded by the teacher to be stored for later retrieval by a student to learn.

The *LearnTool* provides the mechanism for the student to learn a pre-recorded track. This tool provide all of the functionality to compare the student's motion in real time with the teacher and convert this difference into parameters that can be used by the sonification process to alter the sound in term of pitch or timing effects (such as the ADRS envelope described in Chapter 3:). This tool also has the ability to draw visual affordances that can be shown to the user to aid in the accurate reproduction of the movement as described in Section 4.2.

The final interaction class is the *CommandTool*. This is simply the default tool that supports the command mode when no other tools are selected. When this tool is active, the user can enter command or move the cursor around in the viewport and select objects.

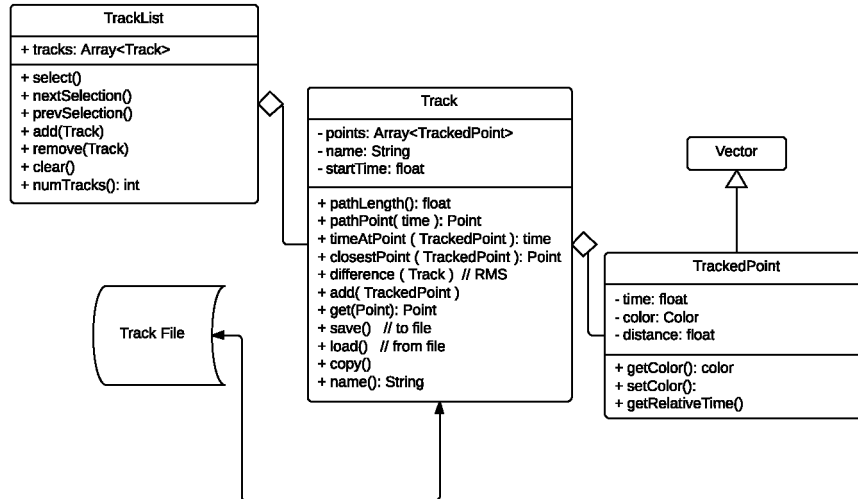
There is an additional base class, called the *DrawTool*. This tool provides all the common functionality used by tools, which are required to draw in the viewport. Even though the *CommandTool* is not primarily a drawing tool, it still is required to draw objects, such as the motion path start/end indicators in the viewport, when the cursor is moved over them.

## 4.5 Tracks

The Track object is the fundamental data structure for storing the underlying motion path representation. The track stores the data as an array of TrackPoints. Each TrackPoint contains the 3-D coordinate representing a position in 3-D space along with the time<sup>2</sup> it was recorded. Tracks are persistent in that their data can be cached to a file on the disk and retrieved for later use. The track classes are illustrated in Figure 4-6.

---

<sup>2</sup> In *Processing* language, time is retrieved using *millis()* call which returns the number of milliseconds since the program was started.



**Figure 4-6.** The persistent Track class with the TrackList container and TrackedPoint (based on Vector) classes.

The Track class supports the mathematical operations required to evaluate a curve along its path. Since the path is based on a fairly dense array of data points, the curve is stored as a *piecewise linear approximation of a curve*. From this data, the path length along any point in the curve and the distance between any two points along the curve is easily computed.

This is useful for sonification; for example, by calculating the relative distance traveled along the curve at a given time,  $t$ , and comparing that distance traveled to the teacher's distance at time  $t$ , it can be determined how much faster or slower the student is going with respect to the teacher. This difference can be used as a parameter, and fed into the sonification mapping. A sound track could then be sped up or slowed down by this difference. Another sonification possibility might be to use the student's progress along the curve (in terms of percentage of path length traveled) as an index into an envelope, which can control the overall energy of the sound when played back in real time. This is the approach used by the ADSR sonification described in Section 3.5.1, The Attack, Decay, Sustain, Release Model (ADSR).

As part of the system, the *TrackList* object provides a container object for storing multiple tracks. This class provides an interface for selecting, adding, and removing tracks. The *SoundTracerApp* class maintains the *TrackList* as the central storage area for tracks that are created and saved.

The architecture of *SoundTracer* proved to be flexible framework during both the system development phase and testing phase. The use of an object-oriented approach, which *Processing* supports, provided an extensible framework for experimental programming. With good design sensibility used in the base classes, new tools could be added as needed. The interpretive nature of *Processing*, while not as powerful as other languages such as Lisp, provided fast iterative development and modern support for libraries and devices — a big win for a rapid prototyping solution.

## Chapter 5: System Testing

Preliminary tests have been conducted on the system with several users testing the ADSR sonification method as described in Section 3.5.1. The goal of the testing is to provide us with an experimental methodology for analyzing 3-D motion, to provide initial feedback on the workflow process in order to increase usability of the system. This has been accomplished with focused testing on a few users. Once this milestone has been accomplished, a more rigorous study can be done.

The following section describes the initial test configuration of the system in terms of the various combinations of sound and visual aids used along with the three canonical types of motion tested: horizontal planar, vertical planar, and mixed 3-D.

### 5.1 Test Setup

#### 5.1.1 Sonification and Visual Aid Combination

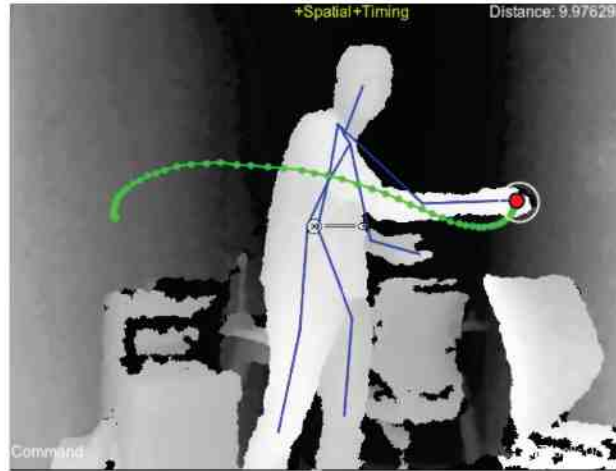
As discussed in Sections 3.4 and 4.2, there are visual affordances provided in the 3-D viewport to assist the student in locating the motion path to learn, to show progress along the path, and evaluate how well the student is tracking along the trajectory. A feature is provided in the application to enable the visual aids to be turned off to allow the student to use purely sound as an aid for tracing the path. In this fashion, three canonical movements are tested (described next) using different combinations of visual aids and sound. It is expected the case that with both visual and sound *engaged* would produce better results than the case with sound alone, however, I believe there is value in getting good results with sound alone to support activities where visual cues or feedback might not be available to the student while they are training.

#### 5.1.2 The Canonical Moves

Three movement types were standardized for these initial tests: 1) *Horizontal Planar* and 2) *Vertical Planar* and 3) *Mixed 3-D*. These are illustrated in the following figures.

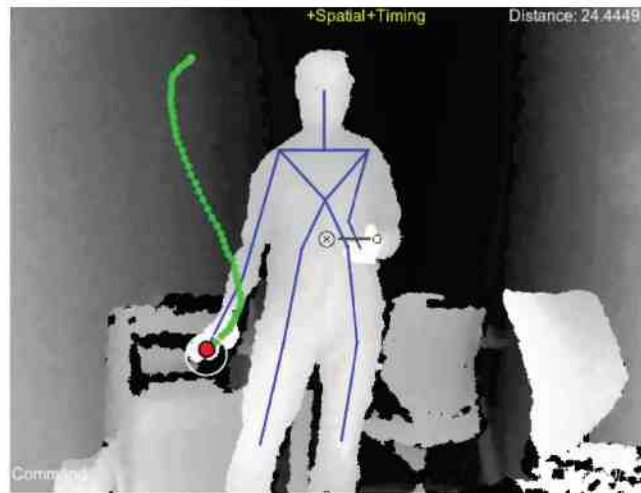


Figure 5-1 shows an example of a *horizontal planar* move. In this case, the user is moving mostly in a horizontal direction. Some displacement in the y-direction (up and down) is used to avoid straight paths that are too predictable for the student. In addition, there may be some movement in the z-direction (depth) to account for the fact that the arm is rotating.



**Figure 5-1.** Horizontal planar movement.

Figure 5-2 shows an example of the *vertical planar* move. As in the *horizontal planar* move, some displacement is allowed in the off-plane directions (in this case, x and z) to avoid using straight paths.



**Figure 5-2.** Vertical planar movement.

In order to test motion that has all three directional components, a third and final move type was created, which is called *mixed 3-D*. For this move, I created a complex composite move to test movement in all directions of *x, y, z*. This move combines some qualities of both the horizontal and vertical moves. Figure 5-3 shows an example of a more complex mixed 3-D move.

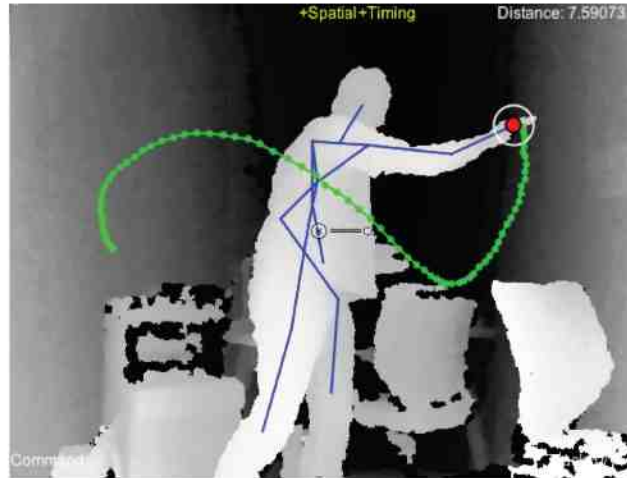


Figure 5-3. Mixed 3-D movement.

### 5.1.3 Example Score Card

Table 5-1 shows an example of a *take* with two users with five trials each. They are scored for the three movements (planar horizontal, planar vertical, mixed 3-D) with level VIS+SON on, indicating that both visual aids and sonification aids are active.

User (n)	Trial (n)	Mapping (ADSR, SAMPLED)	Level (VIS, VIS+SON, SON)	Horiz. Score (RMS)	Vert. Score (RMS)	Mixed 3D Score (RMS)
1	1	ADSR	VIS+SON	112	194	407
	2	ADSR	VIS+SON	68.7	55	227
	3	ADSR	VIS+SON	98.8	85	153
	4	ADSR	VIS+SON	82.7	47	109
	5	ADSR	VIS+SON	59.1	77	148
2	1	ADSR	VIS+SON	147	74.6	398
	2	ADSR	VIS+SON	81.2	48.9	294
	3	ADSR	VIS+SON	55.1	54.36	188
	4	ADSR	VIS+SON	77	76.46	146
	5	ADSR	VIS+SON	52.1	51.47	127

Table 5-1. Sample score card for two students using three types of motion.

### 5.1.4 Evaluating the Difference using *Root Mean Square*

I used a simple *root mean square* calculation (RMS) to score the difference between the student's curve and the teacher's curve (the reference data) at each time interval. This is shown in equation 5.1. A lower number indicates a better score. The RMS value calculated is the square root of the total of the squares of the distance between the student's point at time,  $t$  and the teacher's point divided by the number of points. The RMS value is calculated using this equation:

$$S_{rms} = \sqrt{\frac{(d_1^2 + d_2^2 + \dots + d_n^2)}{n}} \quad (5.1)$$

where  $d$  is the Euclidean distance between each of the student's points,  $P(n)$  and the corresponding point,  $Q(n)$  in the reference  $N$  data points.

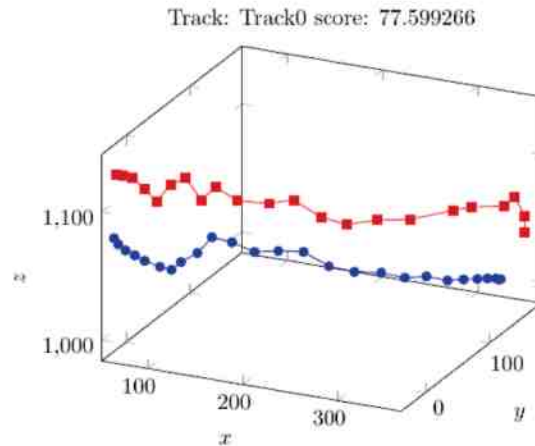
To compute each  $P(n)$  and  $Q(n)$  pair, the algorithm sequentially iterates through the student's data points in the curve from  $n = 0$  to  $N$ . For each student point  $P(n)$ , the time stamp  $t$  which was recorded along that point is found. Since the generalized curve model for the motion path can parameterize the curve by time, the value is obtained using  $Q(n) = R(t)$ , where  $R(t)$  is the function returning point on the reference curve at time  $t$ .

Note that  $t$  in  $R(t)$  may not correspond to a time stamp in the reference data since the times in student data and reference data may be different.  $R(t)$  interpolates the value of the point returned when  $t$  corresponds to a point captured between two data points.

It is possible to have widely varying scores for different movements. For example, a mixed 3-D move is more complex than a simple planar movement (horizontal or vertical), so it is expected that it would be more difficult to get a lower (or better score) with a complex move than a simpler move.

### 5.1.5 Automatic Generation of 3-D Motion Plots for Analysis

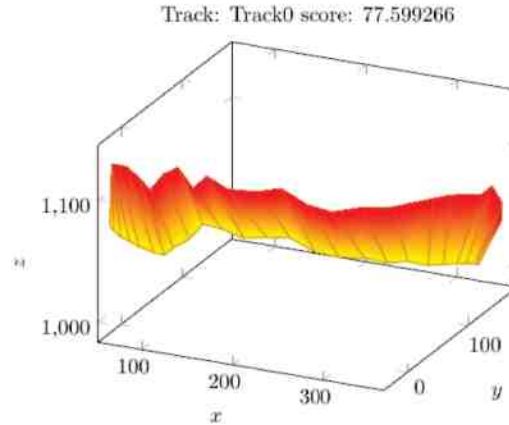
A novel technique was used for visualizing surface plots for the teacher's motion and corresponding student's in the same graph. This was developed using a simple translator to generate a script for the LaTeX-based[33] package, *pgfplots*[34]. In this fashion the motion paths can be compared as individual curves or as a surface mesh with a colormap to visualize the gap between the curves. Because the curves are represented as data points in time, the faceting in the surface provides an interesting way to visualize timing differences in timing. Please refer to Figure 5-4 below for a description of this approach in more detail:



**Figure 5-4.** 3-D plot of teacher (red) and student (blue) motion paths.

The teacher's data (or the reference motion the student is trying to learn) is shown in *red*. The student's attempt is shown below it in *blue*. The two paths are displaced from each other, indicating that the student was consistently traveling off the reference path. Also, note the spacing in the data points. They are different, which is indicating timing differences along the motion path between the student and teacher. If their speed and acceleration were identical the points would have the same spacing. In addition, if the path trajectory is the same, both curves would be identical and occupy the same space. The RMS value of 77.599 used for the score reflects the differences in timing and displacement between the curves.

This can be further visualized by the connected surface as seen in Figure 5-5, where the top boundary of the surface (blending to red) is defined by the teacher's motion path and the bottom boundary (blending to yellow) is defined by the student's motion path. The data points for each of these paths acquired at the same relative time are connected one-to-one. The facets help visualize this and the differences in timing between the two paths.



**Figure 5-5.** 3-D surface plot showing gap between teacher (red) and student (yellow) motion paths.

The algorithm used to generate the student's curve is simply the loop to generate all the stored data points in the track:

```
for each point in student-track, do
  plot point;
```

The algorithm used to generate the teacher's curve is a bit more complex. If *path-point*(*t*) is defined at time, *t*, any 3-D point on a motion path parameterized by time *t* and *time*(point) as the time that the point was acquired (this is stored in relative time to the when the acquisition started in milliseconds), the algorithm can be stated as:

```
for each point in student-track, do
  plot(teacher-track.path-point( time(point) ) );
```

It was mentioned that in *SoundTracer*, an exporter was implemented which generates a LaTeX script to generate a high quality 3-D plot. In the implementation, the **plot** function in the pseudocode above is output commands to export the points and generate the necessary LaTeX syntax to generate the plot.

An example of a LaTeX plot file that was generated is shown in Figure 5-6. Note that the point list is abbreviated for the purposes of being able to show the figure in a reasonable amount of page space.

```
*** This file is generated by SoundTracer***
%
% Learning Track Name  Track0
% Score  77.599266
\documentclass{article}
\usepackage{pgfplots}
\pgfplotsset{compat=newest}
\pagestyle{empty}
\usepgfplotslibrary{patchplots}
\begin{document}
\begin{tikzpicture}
\begin{axis}[mesh/ordering=x varies,
title=Track  Track0 score: 77.599266,
xlabel=$x$,
ylabel=$y$,
zlabel=$z$ ]
```

```
\addplot3[surf, shader=faceted interp, mesh/color input=explicit] coordinates {  
% student track  
(324.08862 , 180.68599 , 997.12415) [color=yellow]  
(322.50317 , 179.3502 , 997.15405) [color=yellow]  
(322.85056 , 175.01372 , 999.6975) [color=yellow]  
(316.47406 , 171.35287 , 1000.3496) [color=yellow]  
...  
% teacher track  
(366.5238 , 154.91379 , 1049.5256) [color=red]  
(363.19025 , 160.2291 , 1059.1744) [color=red]  
(362.41022 , 161.67119 , 1058.8422) [color=red]  
(350.20352 , 163.31898 , 1071.1053) [color=red]  
...  
};  
\end{axis}  
\end{tikzpicture}  
\end{document}
```

Figure 5-6. 3-D plot file script generated by *SoundTracer*.

## 5.2 Test Results and Discussion

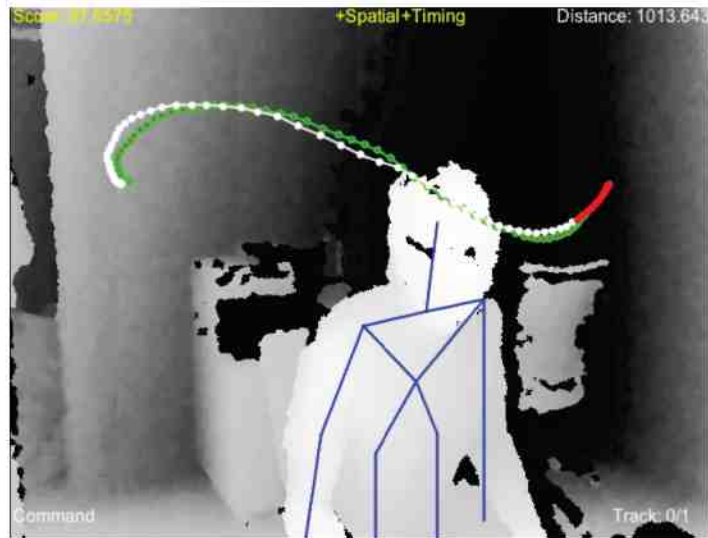
In the remainder of this chapter, I will summarize my initial experiences testing *SoundTracer*. The system is complex enough that some initial familiarity training is required for the test subjects before they can begin to use the system for actual motion training. The initial introduction includes: 1) becoming familiar with how to visualize yourself moving in the monitor/display, as it behaves much like a mirror<sup>3</sup>; and 2) understanding the basic visual aids in the system which have been described in previous sections, with the start marker being most important as it is the first point on the curve where the motion is evaluated.

### 5.2.1 Semi-Planar Motion – Starting Simple

As a first step, a horizontal motion example was used, which is the simplest of the three types studied (horizontal, vertical and mixed 3-D) from a performance perspective. Horizontal motion testing was performed as described in Section 5.1.2 (The Canonical Moves). In this case, both visual and sonification aids were activated. The horizontal s-shaped move that was recorded is shown in Figure 5-7. Although I am classifying this as semi-planer horizontal motion, it does have some curvature. For the initial horizontal/vertical motion types, the motion should be simple enough to reproduce so that the user can gain some experience using the system. Conversely, it should not be too easy. So I start with a curve that has a horizontal orientation, but it may have some curvature.

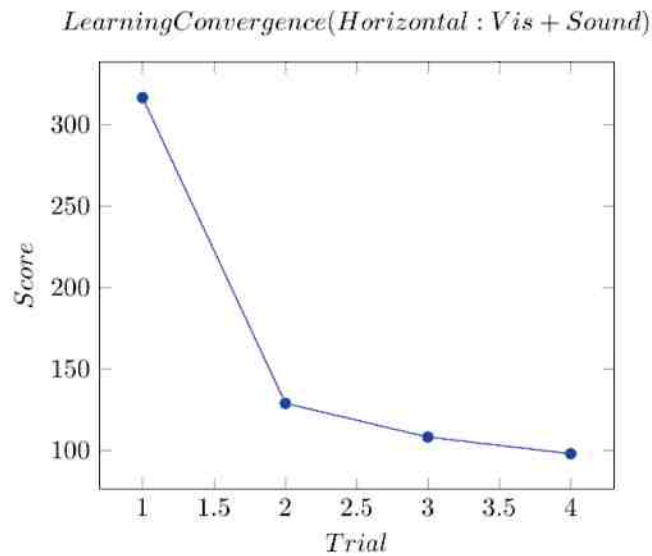
---

<sup>3</sup> The image of the user in the window is generated from the perspective of the Kinect camera, therefore it was required to add functionality in *SoundTracer* to *reverse* the image so that it would appear as a mirror image of the user. In this fashion, the hand moving to the left, for example, would move to the left on the screen.



**Figure 5-7.** Horizontal motion captured.

The reference curve captured by the teacher is shown in green. The student's attempted curve is shown in white. With the exception of the red area where the student went off the curve (in z-direction), the student's performance mimicked the teacher's fairly accurately with timing and spatial aspects within tolerance. The score of 97.66 reflects a good score for this data reference curve based on testing experience. The scores for each trial are plotted in the graph in Figure 5-8.



**Figure 5-8.** Learning convergence of horizontal motion, four(4) scores plotted.

The scores for each trial are shown in Table 5-2. As shown there is a reduction in the score with each successive trial. A reduction correlates to an improvement in spatial and timing accuracy. The initial trial shows the user's first attempt, which was relatively inaccurate. Further attempts show improvement.

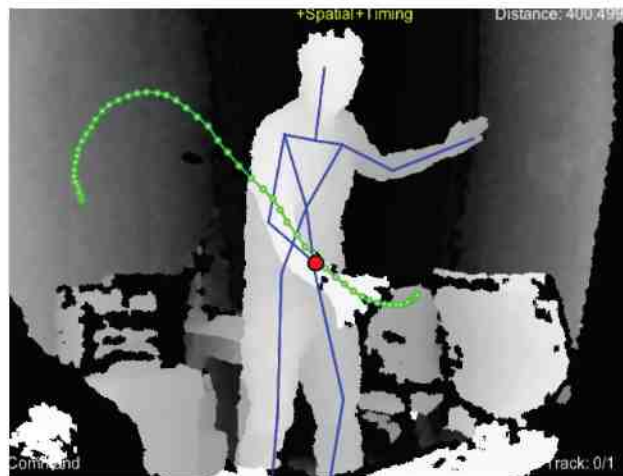
User (n)	Trial (n)	Sonification Type		Motion
		Mapping (ADSR, SAMPLED)	Level (VIS, VIS+SON, SON)	Horiz. Score (RMS)
1	1	ADSR	VIS+SON	316.4
	2	ADSR	VIS+SON	128.85
	3	ADSR	VIS+SON	108.21
	4	ADSR	VIS+SON	97.96

**Table 5-2.** Horizontal motion — four trials converging to an acceptable score.

In the next section, more complex motion is explored using visual aids and sound in various combinations.

### 5.2.2 Mixed 3-D Motion – Adding Complexity

In this section, more complex data for mixed 3-D motion (as described in Section 5.1) is tested. This motion will have free movement in all three axes ( $x$ ,  $y$ ,  $z$ ) and could be commensurate with a move associated with sports (e.g., a tennis swing with a topspin) or a more complex arm/hand motion in dance. Due to limitations in the hardware performance and latency, the move will not be executed as quickly as the swing of a tennis racket or golf club, but it will be executed fast enough to test the general form. The aim is to obtain both spatial and timing accuracy when the move is reproduced. Figure 5-9 shows the mixed 3-D motion path recorded by the teacher.



**Figure 5-9.** Mixed 3-D motion captured.



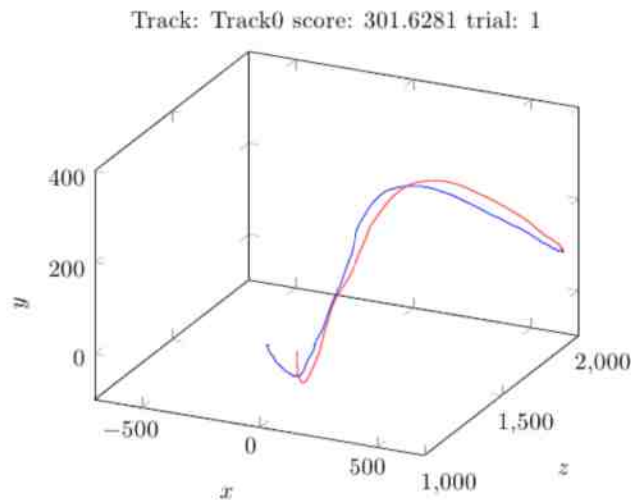
Table 5-3 shows the scores (RMS values) calculated by *SoundTracer* for each trial of a single user attempting to reproduce the motion using sound and visual aids together. As mentioned, a lower score or lower RMS value can be interpreted as a better score with less spatial and timing differences between the student's motion path and the teacher's.

**Table 5-3.** Summary of seven trials with visual and sound aids active.

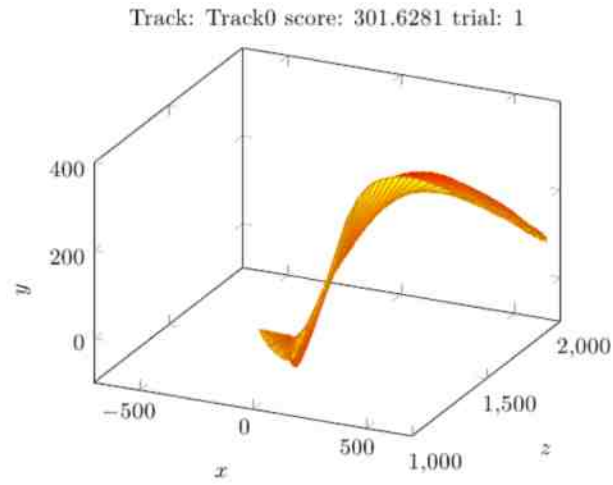
		Sonification Type		Motion
User (n)	Trial (n)	Mapping (ADSR, SAMPLED)	Level (VIS, VIS+SON, SON)	Mixed 3D Score (RMS)
1	1	ADSR	VIS+SON	301.63
	2	ADSR	VIS+SON	149.55
	3	ADSR	VIS+SON	107.28
	4	ADSR	VIS+SON	134.04
	5	ADSR	VIS+SON	145.0
	6	ADSR	VIS+SON	115.59
	7	ADSR	VIS+SON	114.03

The initial high value of 301.63 for trial 1 indicates a substantial amount of difference between the student's motion and the teacher's, but with both visual and sound aids activated, the score rapidly improves in the second trial and remains below 150 for trials 2–7. Trial 5 shows a regression (although not as bad as trial 1) to 145.0, but there is some improvement after that.

Figure 5-10 and Figure 5-11 show 3-D surface plots of the data for trial 1. The plots were generated using the data export system implemented in *SoundTracer* that was described in Section 5.1.5.

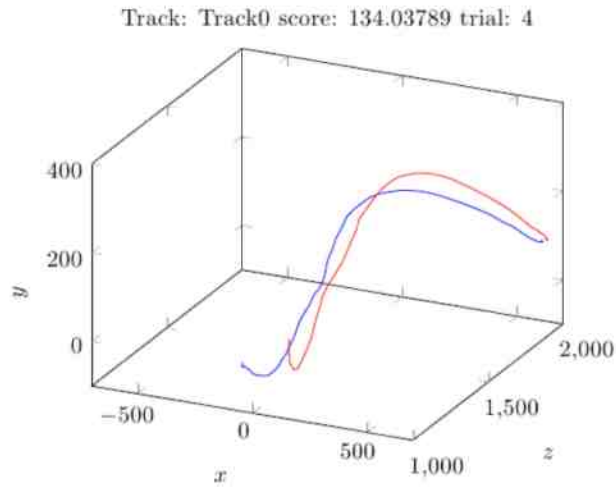


**Figure 5-10.** Trial 1. Reference motion (red) and student's attempt (blue).

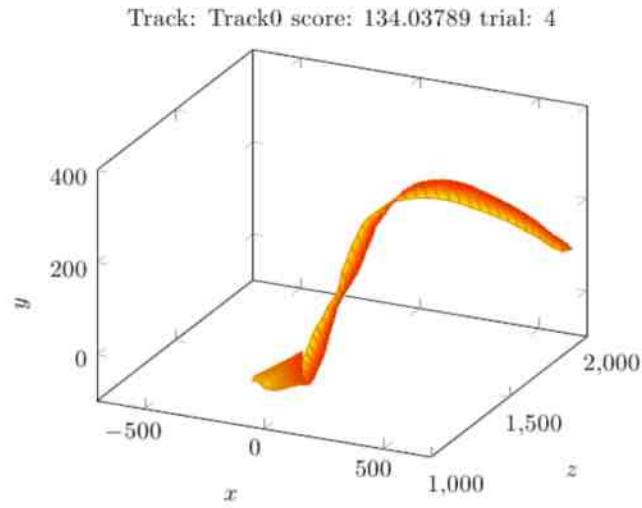


**Figure 5-11.** Trial 1. Surface between two paths. Non-uniform tessellation shows that timing is considerably different from reference motion.

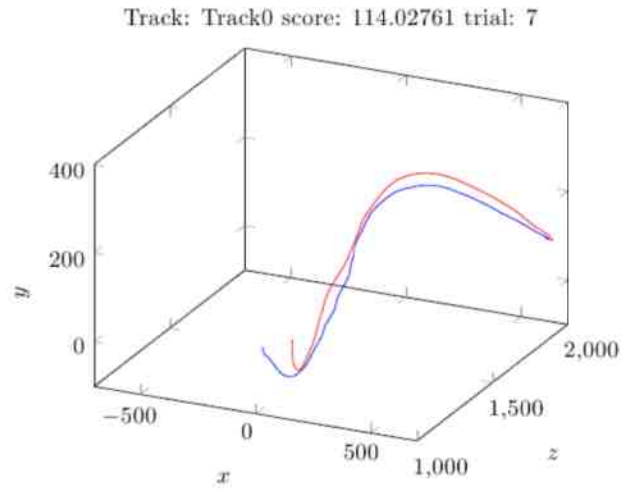
Test results for Trial 4 are shown in Figure 5-12 and Figure 5-13.



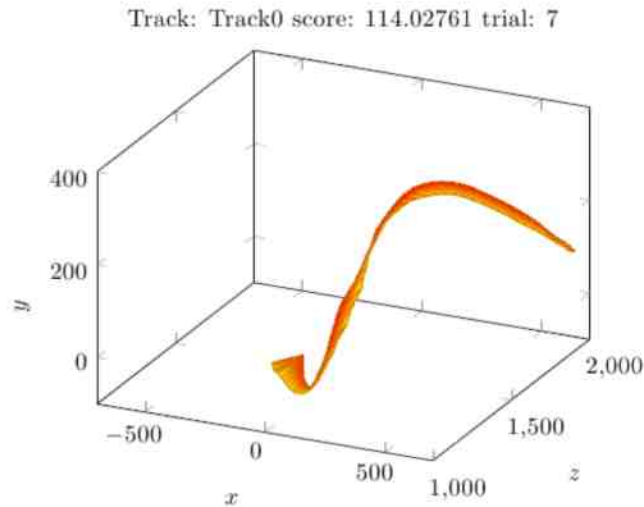
**Figure 5-12.** Trial 4. Student's motion (blue) is fairly close to reference (red), but improvement not apparent until the surface plot is reviewed.



**Figure 5-13.** Trial 4 surface plot. Tessellation is more uniformly distributed. Figure 5-14 and Figure 5-15 show the results from Trial 7.



**Figure 5-14.** Trial 7. Reasonably good match between student's motion and reference.



**Figure 5-15.** Trial 7. The thin ribbon-like profile of the surface indicates the paths are similar shape. Tessellation is uniformly distributed.

In Figure 5-11, Figure 5-13, and Figure 5-15, the 3-D data is shown for a sampling of three trials. Note that in those cases both sound and visual aids were used, therefore, more rapid convergence would be expected because the user has visual aids in addition to sound. Another observation is that in first trial, the initial shape is not too far from the reference motion, indicating that the spatial accuracy starts off relatively high level in the first trial. Even though the shape is fairly accurately reduced, the tessellation shown in the trial 1 surface plot (Figure 5-11) is not even, indicating the timing is inaccurate. As shown in Figure 5-13 and Figure 5-15 surface plots for trials 4 and 7, the tessellation is significantly improved which indicates the student is reproducing the move with better timing.

The motion has thus far been reproduced using both visual and sound as aids, but what happens when the user has a minimum of visual aids and relies mostly on sound? In the data set, trials are analyzed for a user using *sound only*.

In this particular sequence of tests, the user starts with a large variation from the reference data and gets worse in trials 3-5 before there is some convergence to a good result. Note that the best score shown of 141.62 is actually a very good score for this particular data set. In Figure 5-16 through Figure 5-21, a subset of the trials (3, 7, 11) are plotted.

Table 5-4 shows the scores for eleven trials using a mixed 3-D data set with sound as the only navigation aid for the student<sup>4</sup>.

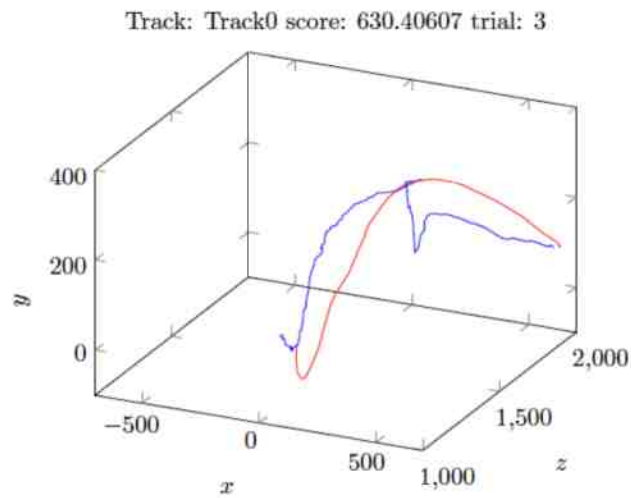
---

<sup>4</sup> It was found that the user needed at a minimum a path start marker to know where the motion starts, otherwise, they wandered aimlessly in space at the start of the test. The other visual affordances were hidden with exception the user's own hand marker.

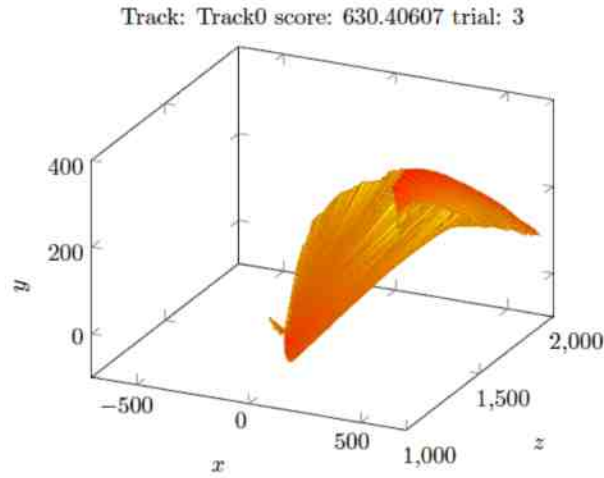
		Sonification Type		Motion
User (n)	Trial (n)	Mapping (ADSR, SAMPLED)	Level (VIS, VIS+SON, SON)	Mixed 3D Score (RMS)
1	1	ADSR	SON	343.44
	2	ADSR	SON	338.2
	3	ADSR	SON	630.4
	4	ADSR	SON	650.99
	5	ADSR	SON	673.9
	6	ADSR	SON	262.78
	7	ADSR	SON	289.38
	8	ADSR	SON	147.2
	9	ADSR	SON	172.54
	10	ADSR	SON	187.24
	11	ADSR	SON	141.62

**Table 5-4.** Table showing 11 trials with sound only aiding the student's motion.

Below, Figure 5-16 and Figure 5-17 show the results of Trial 3 shows wide.

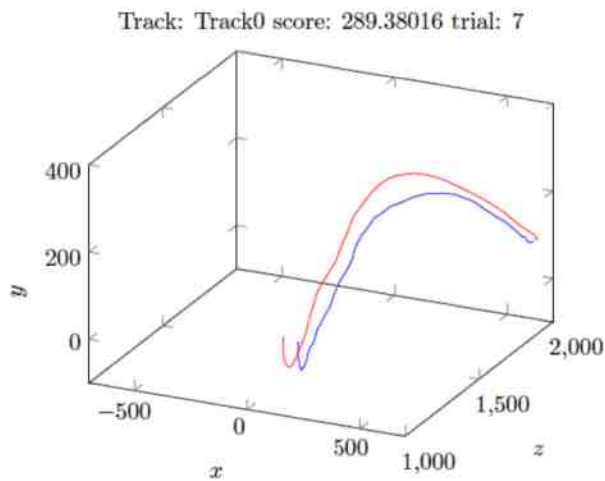


**Figure 5-16.** Trial 3. Wide variation between student (blue) and reference path (red) with correction at approximately (400, 300, 1500).

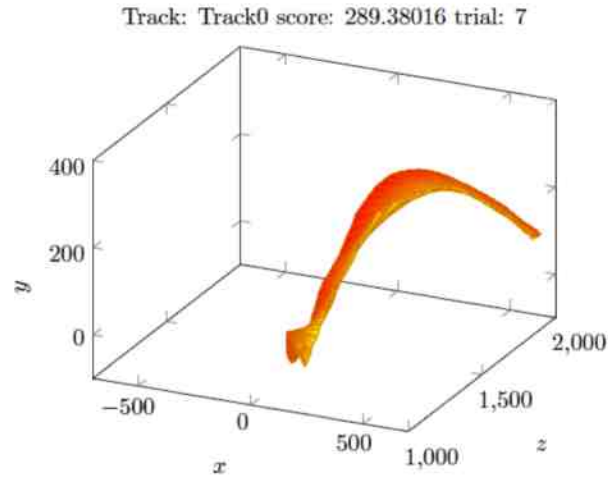


**Figure 5-17.** Surface plot. Potting package is not able to tessellate spike in correction properly however, wide variation in motion path shape and timing are clearly shown with non-inform tessellation.

The results of Trial 7 are shown in Figure 5-18 and Figure 5-19.

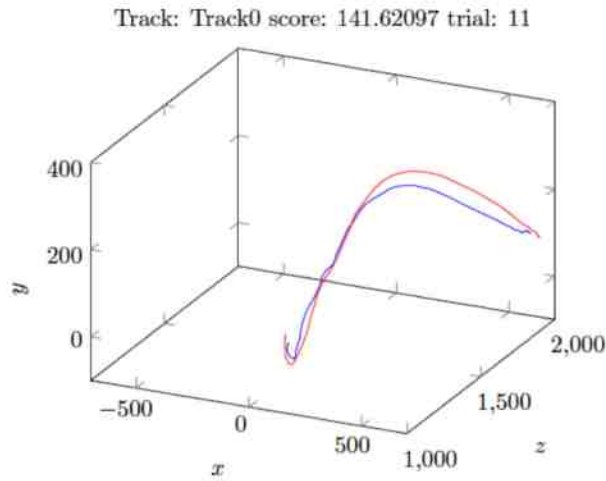


**Figure 5-18.** Trial 7. Convergence of shape between student (blue) and reference data (red).

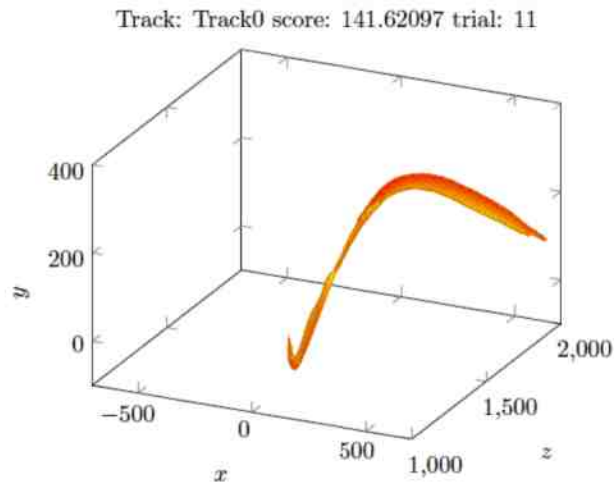


**Figure 5-19.** Trial 7. Surface plot shows narrower gap between plots, although tessellation still shows significant timing differences.

Figure 5-20 and Figure 5-21 show the results of Trial 11.



**Figure 5-20.** Trial 11. Closeness of shape between student data and reference.



**Figure 5-21.** Trial 11. Plot shows regular tessellation with fairly close execution of timing.

For the *sound only* learning session, more trials were required to converge to a reasonable score than with visual and sound aids used together. A similar test was performed with *visual only* in order to test the case where there is no sound at all. This case will be detailed in Chapter 6 along with a summary analysis of the data.



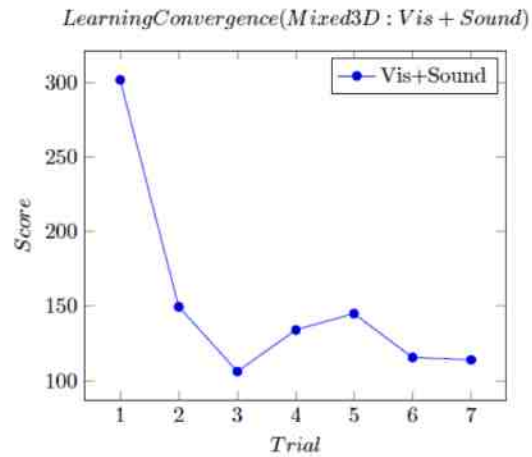
## Chapter 6: Analysis of Results — Summary

In Chapter 5 a testing setup and a method for representing and quantifying differences in the results of the 3-D data generated was presented. I also navigated through a detailed analysis of the results of two different types of motion including one simple and one complex using both sound with visual aids and sound alone. In this section, results will be summarized with some additional analysis and also provide some subjective feedback from users who have been trained on the system.

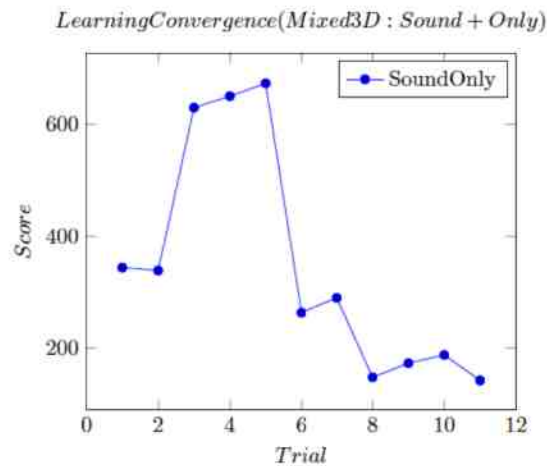
### 6.1 Learning Convergence

For the purposes of this research, *learning convergence* is defined as the number of trials required for a user to obtain an acceptable result. In other words, when the score reaches within a range of an acceptable value for the reference data being used as the learning set. This has to be determined experimentally because included in this variation are the inaccuracies in the hardware motion capture system which is accurate to approximately 2–3 centimeters in absence of noise. For the reference data generated, *good* student scores, which reproduce the data well in terms of shape definition and timing, are in the range of 80–140. This is using the scoring method discussed previously in Section 5.1.4 which calculates the RMS value of the data for each point in student's data against each point in the reference data at time  $t$ .

For the complex mixed 3-D reference data, a 2-D graphical representation of the scores over the number over trials with various sonification options can be shown. Figure 6-1 plots the results of learning convergence for the test session with sound and visual aids. In Figure 6-2, we see the results with sound only while Figure 6-3 shows the results with visual aids only.



**Figure 6-1.** Results with sound and visual aids.

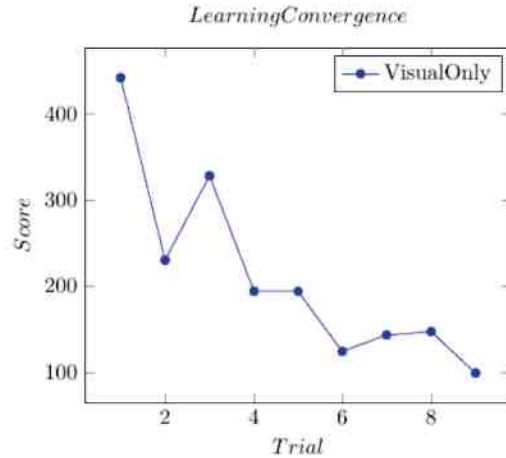


**Figure 6-2.** Results with sound only.

In all cases the user was able to learn the motion to an acceptable degree within twelve trials. Note that there is more regression<sup>4</sup> in the *sound only* case. The user tends to find a solution, then “wander” from it until the scores move towards convergence. With the sound and visual aids enabled, the convergence is more uniform. You can see this graphically when comparing the larger regression in trials 2, 4, 5 in the second figure to the smaller regression in trials 4, 5, 6 in the first figure. With visual only (no sound), the number of trials converges more quickly than sound alone but not as quick as using sound in combination with visual.

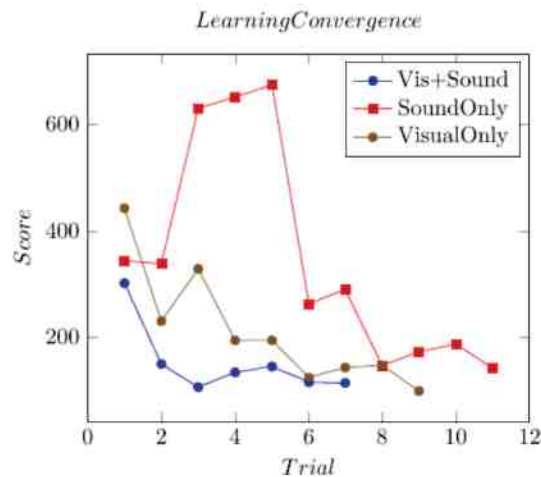
---

<sup>4</sup> In this case, *regression* is defined as when the score actually gets *worse* rather than better.



**Figure 6-3.** Learning convergence for visual aids only.

It is important to note that in all these cases, it is better to use “new” reference data for each set of trials trial for any given user. This requires the user to relearn the motion so the trial will not be biased for the next combination tested. Figure 6-4 provides an overlay graph of all three cases: 1) visual + sound; 2) sound only; 3) visual only.

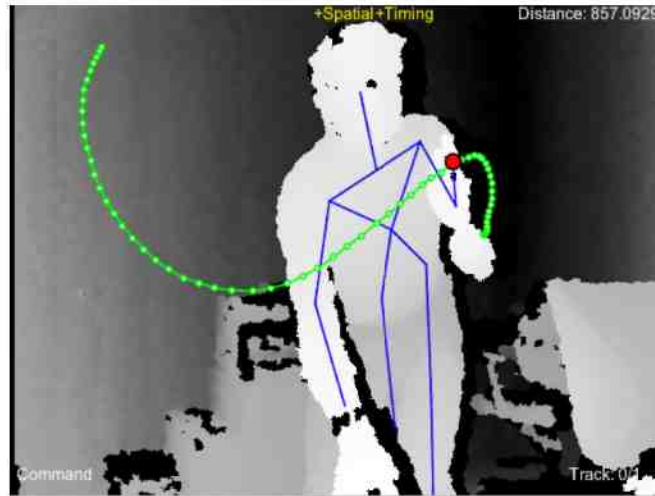


**Figure 6-4.** Combined graph of all three combinations of learning aids.

This graph shows that (in this particular set of trials) the highest convergence was obtained for the case where *visual aids are combined with sound*, followed next by *visual only* and then *sound only*. These results are what I expect given that without visual aids, the student is navigating blind with only sound. The fact that sound alone does converge has value — particularly when visual aids are not available or cannot be seen by the user.

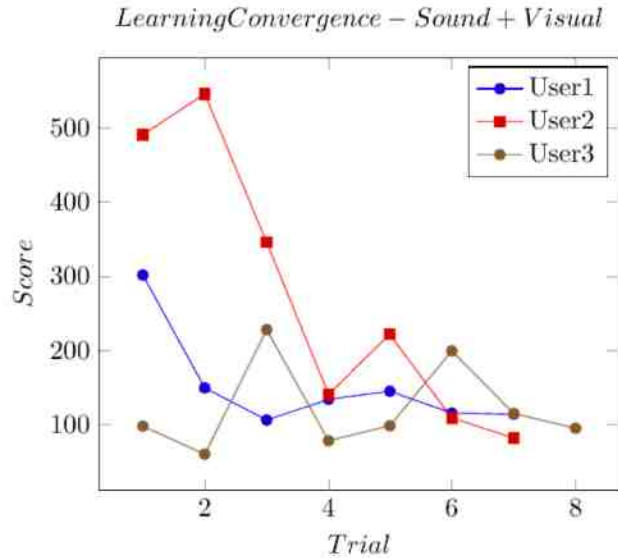
## 6.2 Comparisons Between Multiple Users

The test setup was used for evaluating three users using the reference motion types (horizontal, vertical and mixed 3-D and the sonification combinations (*Visual + Sound*, *Sound Only* and *Visual Only*). Once the users completed simple trials for horizontal and vertical motion to become familiar with the system, they were given a more complex 3-D motion path with horizontal and vertical displacement with a rotation around the body. A screenshot of the motion path is shown in Figure 6-5.



**Figure 6-5.** Reference Motion used for multiple users.

The learning convergence is plotted for all three users tested with both sound and visual aids turned on (*Sound + Visual*) is shown in Figure 6-6. The visual aids are limited to target markers on motion path as discussed in Section 4.2. The reference motion of the teacher was only observed once in the beginning of the test when the motion was recorded.

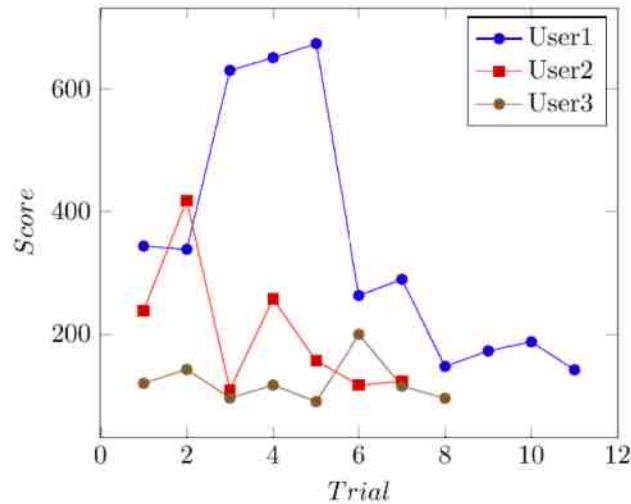


**Figure 6-6.** Plot of learning convergence for three users with sound and visual aids enabled.

For the reference motion used, any score below 150 reflected a reasonable reproduction of the motion and timing within a tolerance of 1-2 cm. Note the difference in learning patterns between the three users. *User1* showed a fairly predictable start with a higher score; he had less accurate reproduction of motion but converged fairly quickly and consistently produced good scores. *User2* had the most trouble learning the system but ultimately produced some of the best results by trial 7; he used the tactic of learning from mistakes to improve his accuracy. *User3* was the quickest learner of the system; he started immediately with a good score, but then struggled a bit to maintain consistency. All users were able to eventually use the system to reproduce the motion path accurately. An interesting conclusion to this test is that it demonstrates how people learn motor skills differently. It would be interesting to study this further with a larger sample of users.

Figure 6-7 shows the same test but with no visual aids turned enabled; only sound is used as an aid. The results are different from the previous tests. *User1* started out with a reasonable start but then diverged in trials 3, 4 and 5 before converging to consistency in trials 9, 10 and 11. Similar results for *User2* were obtained, but the scores were less erratic. Finally, (our champion) *User3* produced a great score from the start with only one smaller regression in trial 6.

*LearningConvergence – SoundOnly*

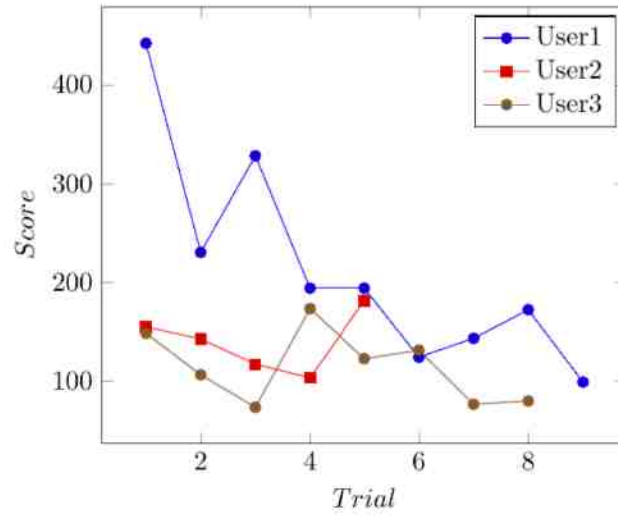


**Figure 6-7.** Plot of learning convergence for three users with sound only enabled (no visual).

Using sound only is obviously the most difficult test for testing because the user is navigating blindly. It is, however, somewhat difficult to draw strict conclusions other than the fact that the three users were able to use sound to reproduce the reference motion and timing with improving ability. Similar reference data set was used previously in the Sound + Visual trial, so it is very likely that the users had become more familiar with the data and the sound primarily provided reinforcement feedback. Further studies using a variety of different reference tracks per test and possibly reversing order of the trials could be done. (See Chapter 7, “Conclusion and Future Work”.)

Finally, I show the results with visual aids only (no sonification) in Figure 6-8. The graph shows that the three users are able to reproduce the motion with only visual aids, however as mentioned, the user had training on the reference motion from the previous two tests, so it is likely that the training experience biased more favorable scores. It is also difficult to conclude how well sound enhances the visual when both are used concurrently. I can conclude, however, that users were able to learn the motion and replicate the motion with improvement using sound only.

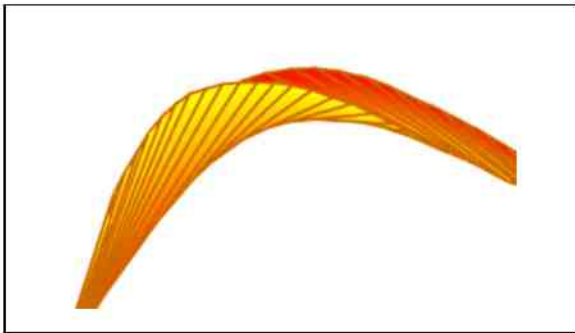
*LearningConvergence – VisualOnly*



**Figure 6-8.** Plot of learning convergence for three users with visual aids only (target markers).

## 6.4 Timing Comparisons

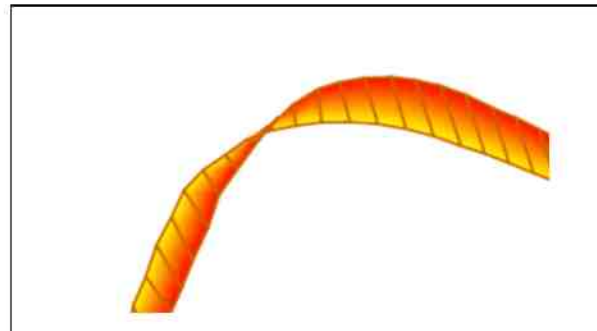
The surface plotting method introduced in Section 5.1.5 proved to be a useful tool for visually evaluating timing comparisons between the student's data and the reference data. This was an unexpected outcome of experimenting with the *pgfplots* package with the data. It was often the case that good reproduction of the shape of the curve still produced a weak score because the user's timing was off. For example, if I zoom in on the data plots for Trial 1 and Trial 4 for the mixed 3-D example with sound and visual aids as shown in the previous section, we see the images in the figures below:



**Figure 6-9. Mismatched Timing**

Figure 6-9 on left shows similar shape motion path but timing mismatch

Figure 6-10 on right shows good shape rendition with closely matched timing.



**Figure 6-10. Closely Matched Timing**

The two curved rails of the plots are each defined by the reference path and the student's path, with the red side highlighting the reference (or target) path. In the example in Figure 6-9, the lines which tessellate the surface interconnect between points along the student path and the target point on the reference data where that point *should be* at each time step. It can be seen that points on the student data all connect to points somewhere *ahead* of the curve in the reference data as shown by the fanning pattern.

With the data shown in Figure 6-10 on the right, not only is the shape rendition good but the points in the student data all line up well with points in the reference data as shown by the more quadric shape to the polygons in the tessellation. Although the polygons are trapezoidal, the difference or offset is fairly minute in terms of time interval. With the mismatched data (Figure 6-9), we can conclude that the timing is *late* with respect to the reference data – points in the student's data corresponds to points that are ahead in the reference data.

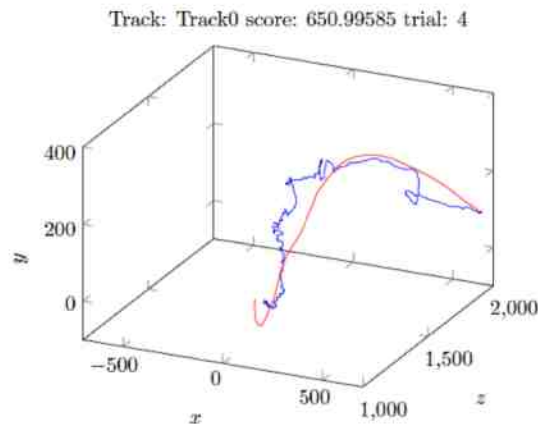


## 6.5 Motion Path Start/End Points

In most of the data sets studied so far, the accuracy at the curve starting location is usually better than at the end points. A typical example is trial 7 in both data sets in the previous example. The starting point is actually on the right and the ending point is towards the left.<sup>6</sup> This data shows a wider gap between reference and trial/student data at the end point than at the start point. This occurs because the trial always starts at the start indicator on the screen. There is a visual aid for this and the student is given time to move their hand/body to the start position of the move before it is executed.

## 6.6 Noise in the System

Sometimes noise is present in the Kinect motion capture system and creates a non-recoverable situation. The cause of this noise is unknown. Once the noise starts, the data being recorded becomes very erratic (see Figure 6-11 below). The noise would be difficult to filter because it is not just isolated to a few stray data points. This problem happens frequently enough that there must be many restarts in the testing process.



**Figure 6-11.** Erratic data caused by noise.

---

<sup>6</sup> The 3-D data is recorded from the perspective of the Kinect camera in which motion drawn from the user's left to right will appear from right to left from the point of view of the camera.

## **6.8 User Feedback/Impressions**

Following each test, feedback was collected from users who tested the system and the comments are summarized below:

**Ambidexterity** – The initial system focuses on motion of a single part (joint) in the body. Most users were not equally as good at reproducing the teacher's path with both hands, particularly the spatial aspect.

**Differences in visual vs. sonic perception** – While visual affordances provided an effective aid for some testers, one tester was distracted by them and produced better scores with sound only.

**Learning the Data Set** – It was found that during testing that once the data set is “learned”, it becomes “stale” for next the experiment. In other words, it will produce biased results when tested against other combinations of sound and visual aids because the user has basically learned the move and can reproduce it more easily in a new test.

**Sonification (Spatial)** – Users all agree that the pitch change sonification helped them to know when and where they went off track, but it was not always clear in which direction to move to correct.

**Sonification (Temporal)** – Initial testing impressions of the ADSR envelope method were favorable. In particular, playback of the teacher's envelope prior to each exercise enabled them to develop a mental image of the sound to make when the correct timing is achieved.

**Tolerance** – Given the accuracy of the hardware and the user, reproducing a path required having a configurable tolerance, effectively converting the path from a line to a “tube.” For broad full-body movements, several centimeters might be acceptable, but for finer hand motion, smaller tolerances may be used.

**Visual representation** – The reference video on the screen is rendered from the perspective of the camera, so the image is reversed from a “mirror”. Most users preferred to see a mirror image of their body. For this reason, I implemented a mirroring function to reverse the camera image on the screen.

# Chapter 7: Conclusion and Future Work

## 7.1 Conclusion

This work provides a new interactive way to learn motor skills leveraging the latest developments in low-cost consumer hardware technology. As a key part of this work, layered sound as a feedback mechanism has been employed in a novel way to provide real time feedback during the learning process. The use of a sound envelope and pitch provides concurrent real time information to the student on both position and timing accuracy. The support for a full 3-D solution enables the study of more complex motion which is not restricted to the plane, benefiting motor skills associated with sports, performance, rehabilitation and the visually impaired.

*SoundTracer* as an experimental system platform for motion sonification and learning has yielded some initial results that are very encouraging. A system has been created that can map real time 3-D motion tracks to layers of sound, which can be used as a feedback loop for learning complex path-based motion. Methods for comparing the motion paths were developed, mapping differences to layered sound envelopes and observing how a student could use this feedback for learning the motion in real time. In addition to the motion sonification system developed, graphical methods for examining motion plots and the gap surfaces between multiple trials of motion were developed. This provides a way to visualize not only spatial differences, but differences in timing between motion paths.

From an implementation perspective, the power of an interpreted language and a flexible sound synthesis system provides an invaluable tool for rapid prototyping. The object-oriented architecture of *SoundTracer* is easy to extend by adding new interactive tools and sound mappings. The sound server architecture of *SuperCollider* can be interfaced with other programs and mobile devices, enabling the architecture to be extended with new programming and provide a convenient remote control while testing. While somewhat limited in performance, the first generation Kinect device provides a low-cost portable capture system. The system developed can easily be extended as new generation devices become available.

The initial testing of the system has yielded very positive results. After training with the system, users can learn complex 3-D moves with both sound-assisted learning and with sound only. The fact that users can navigate a complex path with sound alone is an interesting outcome, which affords new possibilities for the visually impaired or tasks where visual aids are not available.

In an era where gaming is driving consumer technology in many ways, it is important to look at other areas where 3-D technology can help improve the human experience. Motion is a core part of this experience and how we can enhance the learning of it using the concepts from this work offers some new ideas for exploration. There is no limit to what can be done with sound. I would hope this work encourages others to further explore these new possibilities and to bring sound into the foreground of interactive learning.

## **7.2 Future Work**

The experience gathered from this work inspires thought on how some of this knowledge can be used in new applications. There are a number of compelling areas where the sonification of motion as a learning aid could be of interest, especially if solutions could be developed in a cost-effective manner. These ideas would all need to be developed and tested but for the sake of thinking futuristically, let me outline a few:

### **7.2.1 Development of Manual Skills and Navigation for the Blind**

An empirical study done by Brambring[35] suggests that in blind children there is an extremely high degree of development delay in the acquisition of manual skills. Different fine motor skills such as eating with a spoon or drinking from a cup were measured and sharp differences were measured between a sighted group and a blind group. Perhaps my system could be utilized as a training aid to help development in this area. Although my system is focused on precise motor skills in a small capture volume, perhaps it could be expanded to a larger capture volume as shown in the figure below. For example, sonification could assist in teaching a blind person to navigate around obstructions or to learn preset motion paths to navigate an interior space.



**Figure 7-1.** Blind woman navigating through a room with the assistance of a dog.

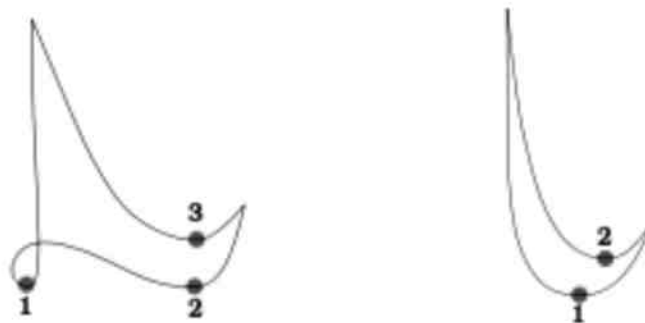
### 7.2.2 Sports Training

The use of video for sports training is prevalent — whether it is swimming or perfecting the golf swing. Wilson covers this in great detail in his article[36]. Most of the technology used records the motion and provides analysis and feedback to the athlete in a coaching session later on. Since my work is focused *instantaneous* feedback, the foundation of it could be used as a new tool for learning sports requiring precise movement.

### 7.2.3 Gestural Interfaces

The system enables us to record precise motion as a reference and make comparisons against that motion. Given this ability, it is conceivable that a number of moves could be recorded in a database of gestures. When motion is compared against those gestures, we can identify an action to be taken. A unique aspect of the system is that gestures can contain timing information, so the movement of the gesture made quickly can mean something different from the same gesture made more slowly, thus the shape and the timing of the motion creates a vocabulary for communication using gestures that can have a compound signature.

As an example of a gestural interface in performing arts, I can use the analogy of a symphonic conductor. In Figure 7-2 below, note the movement of the conductor's baton. The pattern on the left signifies  $\frac{3}{4}$  time; the movement on the right signifies  $\frac{1}{2}$  time. Given the ability to capture movements like this, it might be possible to use the system for either learning of the movement or automatic recognition of the movement. The latter could be used as an interface to the computer that would model movement after a conductor for the purposes of performance (as in electronic music performance) or as a different human-computer interface.



**Figure 7-2.** Conductor's baton pattern for  $\frac{3}{4}$  time on left, while the right shows  $\frac{1}{2}$  time.

#### **7.2.4 Sound Mappings**

The sound mappings used for this work included a novel approach which used an attack-decay envelope to control output of synthesized instrument (e.g., a flute). The goal was to be able to layer both spatial (pitch) and timing information (envelope) in one sound. The strength of this approach was in the real time feedback for the timing. For spatial correction, the pitch change feedback was not always enough feedback for the student to instantaneously understand which direction to make a correction with just a bend or increase in pitch. Additional research on other possible sound mappings that could improve the performance of the system in this regard would be an opportunity for more work in the future. For example, it would be interesting to look at different pitches in different axial directions, or at multi-voice components (harmony) where multiple pitches change concurrently.

There are also many opportunities to alter the type of sounds being used. I am using a full procedural sound synthesis system (SuperCollider). The types of sounds that can be generated are limitless. It is possible that different types of sounds could influence the learning process.

#### **7.2.5 Constraining Degrees of Freedom**

Training a user to correctly track any path precisely in 3-D is difficult, whether sonification and/or visual aids are used. It may be practical to examine ways the user could constrain motion to only two dimensions (say  $x$  and  $y$  first, which corresponds to up/down, left/right). Once the motion is mastered in two dimensions, the third dimension (which is depth or  $z$ ) could then be added.

#### **7.2.6 New Motion Capture Technologies**

Experiments were focused on using the Microsoft Kinect device. I briefly experimented with the Leap Motion device, which is designed for fine motor control (like fingers and hands) and the initial port to this device showed that most of the work done could be applied to fine motor skills as well. As a follow-up to this project, it would be interesting to experiment with the Leap Motion device.

There is also a new version of the Kinect that will be available soon. With greater resolution and less latency, motion can more easily be tracked with greater stability. This could mean an enormous improvement in the tracking performance of the system.

#### **7.2.7 Extending to Full Body Motion**

For the current version, the focus was on the tracking of a single joint in the body. There are many possibilities to extend this concept to full-body hierarchical motion. One next step would be to acquire data for the root of the skeleton and determine if full-body tracking in a small space is possible. In this manner, gross movement of the body could be tracked around a small course. This might be useful for aiding the blind for learning movement in interior spaces. Beyond that we could look at tracking of more than one joint concurrently, so that we could analyze and compare more complex motion consisting full body, arm and leg movements.

### **7.2.8 More Elaborate Motion Study**

It would be interesting to gather more data on different sound mappings, using different movements on a larger sample of test subjects. One challenge is that for any given test subject, it was found that once the motion is learned in one test, using the same motion will bias results in other tests because the user can predict the move after learning it in a previous experience. To alleviate this, it would require the motion sample to change for each test in order to prevent the user from learning the motion.

For a kinesiology or physical therapy major, it would be an interesting thesis to design a larger experiment that would explore different motion libraries and sound mappings in order to determine which ones could be most effective.

## References

- [1] Steven P. Frysinger, "A Brief History of Auditory Data Representation to the 1980s", First Symposium on Auditory Graphs, Limerick, Ireland, July 2005.
- [2] Bradley S. Mauney and Bruce N. Walker, "Creating Functional and Livable Soundscapes for Peripheral Monitoring of Dynamic Data", The Tenth International Conference on Auditory Display, Sydney, Australia, July 2004.
- [3] Alfred Effenberg, Ursula Fehse, and Andreas Weber, "Movement Sonification: Audiovisual benefits on motor learning," The International Conference SKILLS 2011, Montpellier, France December 2011.
- [4] David K. McGookin and Stephen A. Brewster, "Understanding concurrent earcons: Applying Auditory Scene Analysis Principles to Concurrent Earcon Recognition," ACM Transactions on Applied Perceptions, vol. 1, pp. 130–155, October 2004.
- [5] "sonification.de» The Sonification Handbook." [Online]. Available: <http://sonification.de/the-sonification-handbook>. [Accessed: 18-Mar-2014].
- [6] Douglass L. Mansur, Meera M. Blattner, and Kenneth I. Joy, "Sound Graphs: A Numerical Data Analysis Method for the Blind", Journal of Medical Systems vol. 9, pp. 163–174, 1985.
- [7] Perry R. Cook, "Real Sound Synthesis for Interactive Applications", A K Peters/CRC Press, July 2001
- [8] James McCartney, "SuperCollider: a new real time synthesis language," in Proceedings of the 1996 International Computer Music Conference, Hong Kong 1996, pp. 257–258.
- [9] Ge Wang, "The Chuck Audio Programming Language ' A Strongly-Timed and On-the-Fly Environ / Mentality", Doctor of Philosophy Thesis, Princeton University, September 2008
- [10] Niklas Rober and Maic Masuch, "Interacting with Sound", Proceedings of ICAD-04 -Tenth Meeting of the International Conference on Auditory Display, Sydney, Australia, July 2004.
- [11] Alfred Effenberg, Joachim Melzer, Andreas Weber, and Arno Zinke, "MotionLab Sonify: A Framework for the Sonification of Human Motion Data", Proceedings of the Ninth International Conference on Information Visualization, 2005.
- [12] Sandra Pauletto and Andy Hunt, "The Sonification of EMG Data", Proceedings of the 12th International Conference on Auditory Display, London, UK June 2006.
- [13] Katharina Vogt, David Pirro, Ingo Kobenz, Robert Holdrich and Gerald Eckel, "PhysioSonic - Evaluated movement sonification as auditory feedback in physiotherapy", Proceedings of the 15th International Conference on Auditory Display, Copenhagen, Denmark, May 2009
- [14] "Synapse for Kinect." [Online]. Available: <http://synapsekinect.tumblr.com>. [Accessed: 07-Mar-2014].



- [15] "Augmented Reality: The V Motion Project." [Online]. Available: [http://www.wired.com/beyond\\_the\\_beyond/2012/07/augmented-reality-the-v-motion-project/](http://www.wired.com/beyond_the_beyond/2012/07/augmented-reality-the-v-motion-project/). [Accessed: 07-Mar-2014].
- [16] Ajay Kapur, George Tzanetakis, Naznin Virji-Babul, Ge Wang, Perry R. Cook, "A Framework for Sonification of Vicon Motion Capture Data", Proceedings of the 8th International Conference on Digital Audio Effects (DAFX-05), Madrid, Spain, September, 2005
- [17] "Vicon | Systems." [Online]. Available: <http://www.vicon.com/System/Bonita>. [Accessed: 25-Feb-2014].
- [18] "James Cameron - Performance Capture re-invented." [Online]. Available: <http://www.motioncapturesociety.com/resources/articles/miscellaneous-articles/84-james-cameron-performance-capture-re-invented>. [Accessed: 25-Feb-2014].
- [19] "Kinect for Windows Sensor Components and Specifications." [Online]. Available: <http://msdn.microsoft.com/en-us/library/jj131033.aspx>. [Accessed: 27-Feb-2014].
- [20] "Leap Motion | Mac & PC Motion Controller for Games, Design, & More." [Online]. Available: <https://www.leapmotion.com/>. [Accessed: 27-Feb-2014].
- [21] "Kinect Sports (Xbox 360) - Sales, Wiki, Cheats, Walkthrough, Release Date, Gameplay, ROM on VGChartz." [Online]. Available: <http://www.vgchartz.com/game/45607/kinect-sports/>. [Accessed: 25-Feb-2014].
- [22] Object Management Group, "OMG Unified Modeling Language" (OMG UML), Superstructure Specification Version 2.4.1, August, 2011.
- [23] "The SuperCollider Book." [Online]. Available: <http://supercolliderbook.net/>. [Accessed: 24-Feb-2014].
- [24] "Processing.org." [Online]. Available: <http://www.processing.org/>. [Accessed: 24-Feb-2014].
- [25] "Open-source SDK for 3D sensors - OpenNI." [Online]. Available: <http://www.openni.org/>. [Accessed: 24-Feb-2014].
- [26] "Index of /ecmc/docs/supercollider/scbook." [Online]. Available: <http://ecmc.rochester.edu/ecmc/docs/supercollider/scbook/>. [Accessed: 24-Feb-2014].
- [27] "SuperCollider » About." [Online]. Available: <http://supercollider.sourceforge.net/>. [Accessed: 24-Feb-2014].
- [28] "erase ±/ SuperCollider Client for Processing." [Online]. Available: <http://www.erase.net/projects/processing-sc/>. [Accessed: 24-Feb-2014].
- [29] Adrian Freed and Andy Schmeder, "Features and Future of Open Sound Control version 1.1 for NIME," New Interfaces for Musical Expression (Conference) -Pittsburgh, PA, 2009.
- [30] "hexler.net | TouchOSC." [Online]. Available: <http://hexler.net/software/touchosc>. [Accessed: 24-Feb-2014].

- [31] “Andreas Schlegel - oscP5.” [Online]. Available: <http://www.sojamo.de/libraries/oscP5/>. [Accessed: 07-Mar-2014].
- [32] “simple-opensi - OpenNI library for Processing - Google Project Hosting.” [Online]. Available: <https://code.google.com/p/simple-opensi/>. [Accessed: 24-Feb-2014].
- [33] “LaTeX – A document preparation system.” [Online]. Available: <http://www.latex-project.org/>. [Accessed: 04-Mar-2014].
- [34] “PGFPlots - A LaTeX package to create plots.” [Online]. Available: <http://pgfplots.sourceforge.net/>. [Accessed: 07-Mar-2014].
- [35] Michael Brambring, “Divergent Development of Manual Skills in Children Who Are Blind or Sighted,” *Journal of the Visual Impairment and Blindness*, pp. 212–225, April 2007
- [36] Barry D. Wilson, “Development in video technology for coaching,” *Sport Technology Journal*, vol. 1, no. 1, pp. 34–40, June 2008.

## **Additional Publications and Media**

Derivatives of this work have been **accepted** for publication at the following conferences/events.

1. K. Smith and D. Claveau, “The Sonification and Learning of Human Motion”, *The 20<sup>th</sup> International Conference on Auditory Display (ICAD 2014)*, NYU Steinhardt, New York, June 22-25 2014.
2. Competition Entry: California State University Student Research Competition 2014, CSU East Bay, May 2-3, 2014 (Research Summary and Presentation)

A demonstration movie of *SoundTracer* can be found at the URL:

<https://www.youtube.com/watch?v=YLH6gW52Xt8>