Music Kinection:

Musical Sound and Motion in Interactive Systems

Even Bekkedal

Department of Musicology, University of Oslo

2012

“Without a soundtrack, human interaction is meaningless...”

Chuck Klosterman

Abstract

Sound is often used as a feedback modality in technological devices. Yet relatively little is known about the relation between sound and motion in interactive systems. This thesis examines what happens in the intersection between human-computer interaction, motion and sonic feedback. From the connection of music and motion, coupled by technology, we can draw the expression “Music Kinection”. A theoretical foundation accounts for the relationships that exist between sound and motion, and cognitive foundations for these relationships. This study of literature on music and motion, and music cognition theory, shows that there are many aspects that support various relationships between sound and motion. To see if it is possible to detect similarities between users of an interactive system, a user-study was performed with 16 subjects playing commercially available video games for the Kinect platform. Motion capture data was recorded and analyzed. The user-study showed that there is an overall similarity in the amount of motion performed by the user, but that there is some deviation in amount of motion performed by body parts important to the gameplay. Many users will choose the same body part for one task, but will apply different tactics when using this limb. Knowledge from the theory and observation study was used in the practical explorations of sound-action relationships. Two installations, Kinect Piano and Popsenteret Kinect installation, was made, together with two software prototypes, Soundshape and Music Kinection. The practical study showed that working with full-body motion capture and sound in human-computer interaction is dependent on good motion feature extraction algorithms and good mapping to sound engines.

Acknowledgments

Taking on a master's thesis might be, for many, the first project they do of such large scale. This is certainly the case for me. The mere beginning of this project seems very far in the past by now. I have been lucky to have been surrounded by a great community of very talented like-minded people throughout the course of this project. First and foremost, I would like to thank my supervisor Alexander Jensenius for being the pedagogic and structured person he is. Being a part of the fourMs/Robin lab have made the work on this thesis comfortable, giving me access to state-of-the-art equipment, great facilities, inspiring and helpful people, and, not to forget, an espresso-machine. I would like to thank everybody at the lab, and especially Kristian Nymoen, Kyrre Glette, Arve Voldsund, and Ståle Skogstad for helping me through my technological endeavors. Thank you, Kristian, for reading the thesis and giving valuable feedback. Many thanks and warm feelings go to Kai Arne Hansen, who proof-read the thesis and inspired me to "man up" through the rough turns. Thanks to all the subjects who let me dress them up in MoCap suits and jumped willingly around in broad daylight. I would also like to point out that I have had much pleasure and inspiration from following the lectures of Rolf Inge Godøy. Great input and good discussions was provided by contributors and participants of the SMC and ISSSM summer-schools of 2011. Last, thank you Iselin, for giving general support and motivation.

Table of Contents

Abbreviations

Selected keywords that appear frequently throughout the thesis.

Arena - NaturalPoint OptiTrack motion capture software

CV - Computer Vision

GUI - Graphical User Interface

HCI - Human Computer Interaction

Kinect - Motion sensor for the Microsoft Xbox 360

Max - Max/MSP/Jitter programming environment

MoCap - Motion Capture: Recording of movement

NUI - Natural User Interface

Optitrack - Infrared marker-based optical motion capture system from NaturalPoint

OSC - Open Sound Control

SDK - Software Development Kit

SID - Sonic Interaction Design

SMC - Sound and Music Computing

QoM - Quantity of Motion

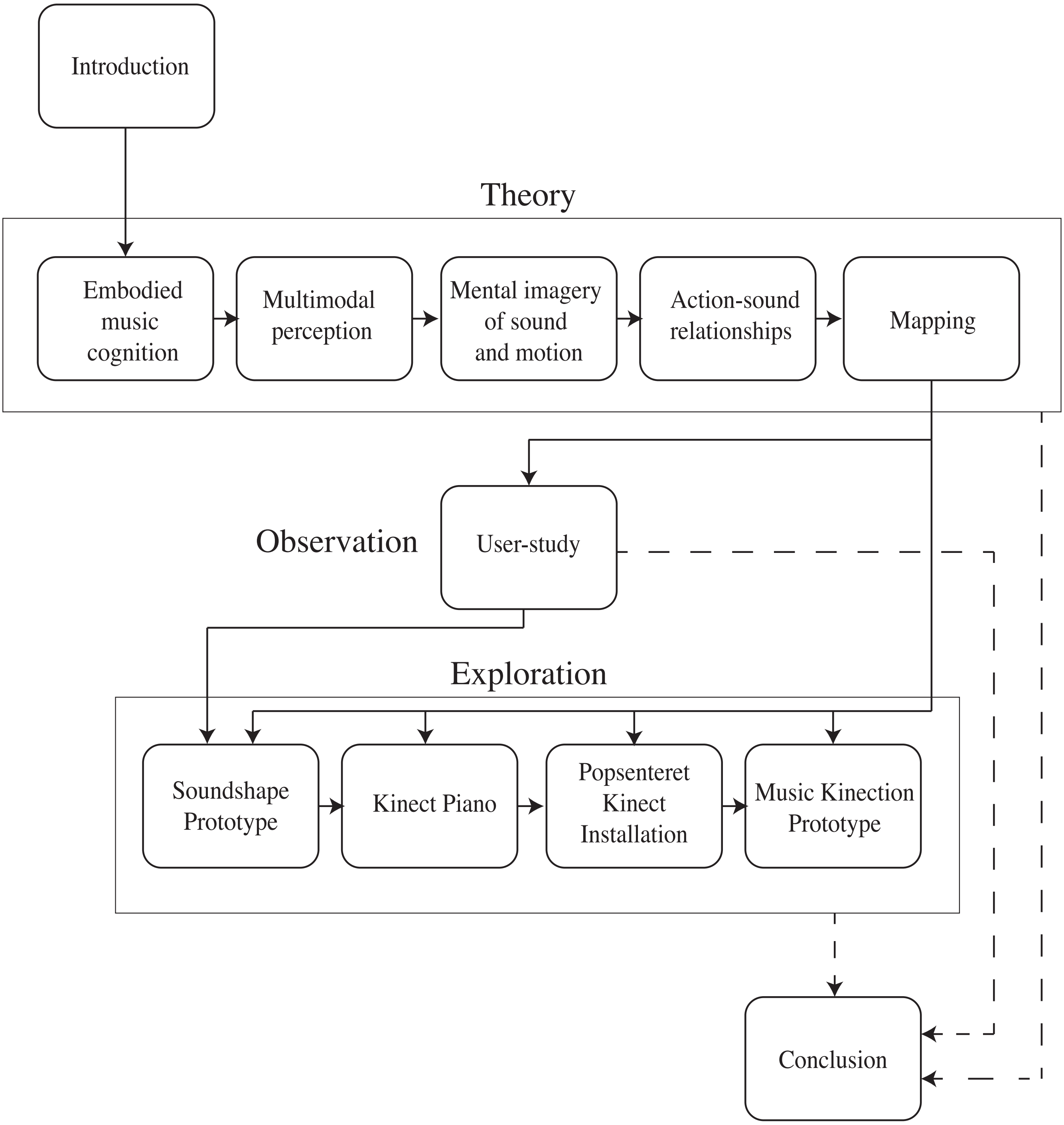


Figure 1: Visual overview of the thesis.

Chapter 1

Introduction

The introductory chapter presents the inspiration and motivation for this master project, as well as the research questions and limitations of the project, and an outline of the thesis.

In a society increasingly dominated by technology, the control that this dependency places upon us permeates our everyday lives. Because the presence of computational routines in the global society is inevitable, it is possible to claim that endeavoring explorations, developments and improvements of *Human-Computer Interaction* (HCI) solutions can be considered as paramount for the optimization of new technological solutions. It is of my belief that research of music and sound within HCI confinements holds an important influence on the evolution of such solutions. As we will see, another aspect of HCI systems that are become increasingly more important, is that of human motion input. This thesis is a study of human motion, sound, and how these elements act and communicate in interactive human-computer dialogues. The term “Music Kinection” is derived from the imagined connection between music and motion in interaction with technological systems.

Most technological devices today are based on some form of sonic feedback. The feedback presented by interactive systems can convey confirming or guiding information, or be purely decorative or entertaining. Following the findings of recent research on *Sonic Interaction Design* (SID) (Rocchesso, 2011) and *sonification* (Hermann et al., 2011), we can assume that audible feedback presented to users during an interaction process will greatly influence how they interact with, and experience the system. Whether the system has an educational (Nordahl et al., 2008), work-related (Hermann and Kõiva, 2008), or other everyday purpose, the sound and music implementation can play a role just as important as any other feedback output from the system (e.g. visual or tactile).

The fourMs research group at the University of Oslo was contacted in 2010 by Johan Basberg of Gatada Games.[1] He presented his ideas for a family oriented game, intended for a new video game technology under development by Microsoft. At the time, the technology was known as project Natal, but was later renamed Kinect. This technology was ultimately able to do full-body *motion capture* in a (pseudo) three-dimensional space, as well as receiving voice commands. As such, the Kinect enables HCI by using the human body itself as a controller, free from any handheld or wearable devices. Using free body movement as input demands a whole new approach to interaction. Basberg's initial idea, with the working title Soundshape, was a game where different audible shapes were to be presented in a room, inviting the user to search the room according to sonic guidance. The user would eventually find the outline for the shape, and guess what was represented. I was connected to Basberg and started to work with ideas concerning programming solutions for his ideas. Although the Soundshape project never expanded to its potential, due to lack of funding, I was inspired to pursue my master's on topics related to the project.

The work I did with Basberg and further development of programming solutions resulted in this *practical* master thesis. Being a practical thesis implies that 50% of the work performed through the duration of the master program is practical work, resulting in the development of interactive installations and software prototypes. The material produced during the practical work is presented as a set of programming code, public installations, and demos. This will be explained in further detail in Section 1.3. The last 50% of the work consists of the theory and analyses presented in this written part. As a direct result of the distribution of work, the written part of the thesis does not elaborate on e.g. theoretical aspects at the same depths in which a theoretical thesis would. This written part presents a theoretical framework and a case study that serve as a foundation for the practical work.

1.1 Research questions

The thesis is based on a set of research questions that concern various aspects of the topic. In light of the inspirations leading into this project, the main research question concerns the interaction between humans and technology, including large-scale actions, e.g. waving for attention

[1] www.gatada.com

or controlling a cursor with the arm. Contrary to this, a small-scale action could be a finger-swipe. The emphasis in the main research question is then placed on the coupling between sonic feedback and large-scale actions in the interaction systems we meet in our everyday ventures:

- *How does musical sound influence full-body motion in everyday interactive systems?*

The term *musical sound* is here used to differentiate from other types of sound, e.g. speech or environmental sounds. This term is also used to focus on the audible sound itself, as opposed to other musical features, e.g. score, cultural or social aspects, etc. Thus musical sound can not be compared with longer and more complex musical structures, but still covers important musical features, e.g. pitch, timbre, texture. *Full-body motion* indicates that the whole body takes part in the interaction, exceeding small-scale movements as e.g. pushing a button or swiping a touch-screen with your fingers. *Everyday interactive systems* implementing this input can include anything from motion sensitive light-switches to exercise equipment and video games.

Based on the main research question, I propose three sub-questions. As a point of departure and support for the observation and exploration performed in this thesis, it is necessary to derive a foundation from a theoretical framework. This is because a theoretical framework will direct focus towards relevant aspects of sound-motion interaction that will be interesting and useful for analysis and exploration. This leads to my first subquestion:

1. *What kind of relationships exist between sound and motion? And what are the cognitive foundations for such relationships?*

Sound and motion relationships and their cognitive premises will be explained in Chapter 2. Following from this, empirical knowledge of how users behave when interacting with a motion-based system is necessary for further research of the topic. This requisite resulted in the second subquestion:

2. *What similarities and differences can be detected between users of a motion-based interactive system?*

This question forms the basis for the case study that will be presented in Chapter 3. After establishing theoretical and empirical knowledge of motion and sound relationships, and how these are presented in current commercial products, it is relevant to explore the development of future solutions for presentation of musical sound in interactive systems. Consequently, the third and final subquestion is:

3. *How can we improve action-sound couplings in interactive systems?*

Action-sound couplings are here understood as a cognitive concept, whereas the technological implementation of these couplings will be what I call *action-sound mappings*. The explorations performed to answer this question are described in Chapter 4.

1.2 Limitations

The research questions presented in the previous section can be approached from various perspectives. My approach is derived from cognitive and technological research fields. It is possible to see this master as an interdisciplinary effort based on the following disciplines:

- *Systematic musicology* - Systematic musicology differs from other musicological discipslines (i.e. ethnomusicology and historical musicology) by being based on a combination of theory development and analysis of empirical data (Clarke and Cook, 2004).

- *Embodied music cognition* - Embodied music cognition is a direction within systematic musicology that concerns the relationships between the human body and musical perception and practice (Leman, 2008; Godøy and Leman, 2010).

- *Music technology* - Music technology can be defined as all use of technology that enables musical practice, such as recording, composition, performance, analysis, etc (Braun and for the History of Technology, 2002).

- *Sound and Music Computing* (SMC) - SMC can be seen as the convergence of various aspects of sound and music research, e.g. synthesis, modeling, and psychoacoustics and musical acoustics (Serra et al., 2007).

- *Sonic Interaction Design* (SID) - SID can be located in the intersection between SMC and interaction design and involves the research and exploration of ways to sonically mediate information in HCI solutions (Rocchesso, 2011; Hermann et al., 2011).

- *Procedural audio* - Procedural audio, also known as generative music or algorithmic composition, can in this context be understood as a community concerned with the creation of processes that will ensure changes in the music and sound design based on input provided by the user (Dorin, 2001).

- *Human-Computer Interaction* (HCI) - HCI involves research and design of strategies involving interaction between human users and computers (Dix, 1998).

- *New Interfaces for Musical Expression* (NIME) - NIME can be seen as a sub-division of HCI, devoted to the research and development of new technological strategies and devices for musical expression and performance (Miranda and Wanderley, 2006).

This thesis will take inspiration from all these disciplines. One of the main concerns reflected in the research questions is the relationships between music and motion. The theoretical framework presented in Chapter 2 is built upon aspects from systematic musicology and embodied music cognition. Approaches from the systematic musicology tradition is also considered in the design and setup of the case study presented in Chapter 3. The remaining disciplines, music technology, HCI, SMC, SID and NIME, are manifested in the practical part of the thesis. Traditionally, research results from these disciplines are revealed through practical exploration and development, as is the case in this thesis.

In the practical exploration of sound and motion in interactive systems, it is fruitful to consider the work in context of SID. According to SID approaches, sound should communicate "information, meaning, aesthetic and emotional qualities in interaction contexts" (Rocchesso, 2011, p. 3). SID, in and of itself, is a vast subject that encompasses many topics, but might be accused of lacking an in-depth focus of corporeal presence and embodied cognitive processes in the interaction process. Corporeal presence can be understood as the presence of the human body and its produced actions (Leman, 2008). A combined HCI and SID approach might be able to more holistically consider a subject corporeally and mentally in contact with a virtual or digital environment and the presented sound design. Nevertheless, SID discourse is still relevant for this project since it is dealing with how users conduct to music in everyday interaction settings. Ideas derived from the SMC and NIME communities are useful when considering synthesis, acoustic and psycho-acoustic approaches in sound design.

As the project was initially inspired by challenges related to a new video game technology, I continued to base my research on technology derived from this category. Video game technologies have shown to be cheap and good solutions also for other HCI uses (Isbister, 2011) as well as for digital music controllers (Jensenius, 2007). Devices such as the Nintendo Wii controller has even been used to perform motion measurements, e.g. the WiiDataCapture software (Toiviainen and Burger, 2011) or studies performed at the IPEM group in Ghent (Leman et al., 2009; Amelynck et al., 2011). I chose to base my research on the Kinect sensor. The Kinect was at the time I started working on my thesis a fairly new device, which had not yet been the subject of much exploration. Still, there seemed to be a small internet community that embraced this new technology and discoveries of how the device could be exploited were constantly being shared. The information shared by this community was enough to get me started with initial driver installation and setup for the Kinect. At this stage, I was very intrigued by the possibilities the Kinect presented.

Free full body movement in HCI addresses many new concerns, and also a new set of design considerations. As an example, the overall motion of the user is now in a much larger scale. This affects e.g. how the user navigates through menus and enable an option. I believe that this in turn should affect implementations of sound design, which seems to be neglected by many developers of video games and other HCI solutions. The Kinect sensor can be considered as a *motion capture* (MoCap) interface, and will be able to serve as a mediator between human motion and sound. MoCap is the recording and storage of motion in the digital domain (Skogstad et al., 2010). State-of-the-art MoCap systems typically consist of multiple infrared cameras that will emit infrared light reflecting in markers placed on object of study. Naturally, working with a state-of-the-art motion capture system would provide much higher resolution and accuracy. However, the Kinect presents a commercial product that will provide a pseudo three-dimensional MoCap, at an affordable price for average consumers.

1.3 Thesis outline

This is a practical master thesis, meaning that during the period the work was conducted, 50% was dedicated to the written part and 50% was dedicated to practical exploration. The work is presented in two parts:

1. The written thesis

2. A set of programming code, installations, and prototypes

1.3.1 Thesis

The thesis is organized around three parts, based on each of the sub-questions presented in Section 1.1.

Chapter 2: Theory

This chapter develops a theoretical framework for the thesis. Reviewing current theory on relationships between musical sound and motion is necessary to understand why we need a larger focus towards corporeal integration in the design of interaction systems. By looking at cognitive foundations for and couplings between sound perception and motor awareness, it is possible to establish whether it is likely to detect potential relationships between sound and motion.

Chapter 3: Observation

This chapter presents observations made from a case study of users interacting with a typical everyday interactive system. The interactive system tested in this study was a Microsoft Xbox 360 with a Kinect sensor. Motion capture recordings were made of 16 subjects playing commercially available Kinect games. Quantitative and qualitative observations make up the results of the study.

Chapter 4: Exploration

This chapter presents the practical exploration by the development of software and installations. Two interactive installations and two software prototypes were created in this process and make up the result of the practical work of this thesis.

1.3.2 Practical Results

Being a practical master, care has been taken to include documentation of all work completed through the duration of the master. This includes:

- Programming code (Max, Matlab)
- Installations (“Kinect Piano”, “Popsesenteret Kinect Installation”)
- Prototypes (“Soundshape”, “Music Kinection Prototype”)
- Video-recordings of demonstrations.

A presentation of the practical work is provided in Chapter 4 and a complete overview is provided in the appendix. All data is also included in the attached DVD disc.

Chapter 2

Theory

"Never confuse movement with action"

Ernest Hemingway

This chapter presents a theoretical framework for relationships between sound and human motion and the cognitive foundations for how a user senses and makes sense of musical sound in relation to motion. The chapter concludes with a discussion on the relevance for this theory in an interactive setting.

2.1 A note on terminology

When reviewing theory concerning motion, especially in musical and HCI contexts, it is easy to get confused by the different terms used about the topic. In this thesis I interpret the term *motion* in a rather general way, describing displacement of the human body or its limbs in space. *Action* is used to describe more specific, goal-directed motion. I here follow the ideas of Jensenius et al., about actions being understood as "coherent chunks of gestures, or delimited segments of human movement having an intentional aspect" (Jensenius et al., 2010, p. 13). Alexander Jensenius et al. argue that, when used in a musical context, the term *gesture* can be used successfully (Jensenius et al., 2010, p. 12). In particular, this is because this term arguably closes the gap between motion and meaning. It is possible to divide gestures into three potential ways of conveying meaning (Jensenius et al., 2010, p. 14); as communication, as control, or as metaphor. In this thesis, the term gesture is primarily concerned with conveying meaning as control. Gestures can be understood as control bearing when they act as components of HCI. When interacting with a computer system, motion needs to express specific control bearing meanings to be able to be interpreted by the system.

2.2 Embodied music cognition

Theoretical knowledge of sound and motion relationships is important to acknowledge in this thesis, since I later wish to consider models rooted in such relationships in the exploration part of the thesis. It is possible to understand what relationships that exist between sound and motion through what is known as *embodied music cognition* (Leman, 2008; Godøy and Leman, 2010). The emerging field of what we can regard as embodied video games is an interesting example of how performance is measured by how well we move (to collect items, avoid obstacles, run fast, throw far, etc). In these games, you will as a user be placed in an artificial (virtual) environment where you have a task to overcome. All perceived stimuli presented within this artificial environment will affect how you move, with the most obvious stimulus being audible and visual.

2.2.1 Ecological and environmental knowledge

Embodied music cognition is derived from concepts of *ecological psychology* and *environmental psychology* developed by Roger G. Barker (1903-1990) and James J. Gibson (1904-1979) in the 1960's and 70's. Their concepts drew upon phenomenological philosophy, founded by Edmund Husserl (1859-1938). Later, what is known as phenomenological perception was established by French philosopher Maurice Merleau-Ponty (Merleau-Ponty, 1968; Gallese, 2003). Gibson was one of the first to establish that there were close connections between perception and action (Gibson, 1966, 1979). He proposed that our cognitive system is not a detached processing entity, but part of a bigger interactive process, involving the mind, our corporeality, and the environment surrounding us.

Gibson's ideas later inspired works on *auditory scene analysis* (Bregman, 1990) and *ecological listening* (Clarke, 2005), which corroborates on the importance of ecological knowledge in the perception of sound. Albert S. Bregman explains how we perform *scene analysis* by executing *grouping* and *stream segregation*, and thus sort out single events from continuous auditory input. Bregman's factors for segregation includes fundamental frequency (pitch), timbre, temporal proximity, harmonicity, spatial origin, etc. Especially if auditory streams evolve with respect to time, segregation is likely to follow principles of common fate, derived from Gestalt psychology. Furthermore, Clarke explains how a musical sound can be recognized by ecological knowledge of sonic features such as shape, mass, and density. He claims that the features mediated by the sound “resonates” with prior knowledge about the sound production. Instead of performing a complex decoding of the stimulus, this ecologic resonance enables us to detect e.g. pitch, rhythm, and instrument identification.

2.2.2 Motor theory of perception

A more specific interpretation of embodied cognition was presented as *motor theory of perception* (Liberman and Mattingly, 1985). This helps us understand the link between corporeal involvement in environmental perception and perception of sound. Studies performed by Liberman and Mattingly showed that speech learning and production is derived from motor mimetic behavior. Upon hearing a word, we will subconsciously perform motion patterns that potentially would recreate the original word. The perception process will perform an automatic conversion of acoustic features into motion features. Rolf I. Godøy has shown how this is relevant also for the perception of more complex sounds, as well as the perception of more complete musical structures (Godøy, 2003). Interestingly, what is stored in memory is not necessarily auditory information, but rather kinematic sequences of the sound-producing action. These sequences can be chunked, stored in a hierarchical manner.

Embodied music cognition models argue for a common representational system for perception and action. Such models are useful to consider with regard to motion based interaction contexts, since they assume that appropriate actions may be produced as a result of certain sensory input.

2.3 Multimodal perception

Multimodality can be understood as the seamless integration of input from several modalities (e.g. vision, hearing, touch). To be able to understand the complexity of sound and motion relationships, it is important to consider the perception of multimodal processes in an interactive setting. As established in the previous section, the mind alone is not sovereign in cognitive processing of sensory input. Similarly, although it is possible to consider the perception of separate sensor stimulus alone, this is not sufficient for the understanding of how we react to interaction systems. Thus, it is important to recognize the cognitive processing of audible feedback as only one part of a multimodal integration of our perceptual “data handling”.

The user of an interactive system will be recipient of different feedback stimuli, either audible, visual, or tactile. These stimuli can happen as single entities occurring in serial order, or as single entities either happening at the same time or in overlapping succession. All this information needs to be perceived, organized, and processed by the user before the appropriate responding action can be carried out. Perception can be regarded as *unimodal* or *multimodal*. Unimodal perception can be understood as the processing of perception data from one modality (e.g. the visual modality) and multimodal perception as the processing of simultaneous perceptual input from several modalities.

2.3.1 Multimodal recognition

The recognition of multimodal processes occur in a certain structure known as the superior colliculus, located in the brain’s cerebral cortex (Wallace and Stein, 1997). Here, multiple neurons representing unimodal events will converge into multimodal events. To be able to merge unimodal events into a single multimodal event, our brain uses multimodal mental images (see Section 2.4) to enable the underlying integration process.

Considering the relatively slow speed of sound, it is at first possible to assume that a lack of temporal coherency might occur in perceptual “data handling”. The result of such error would possibly create failure in the synchronization of a multimodal event. However, dynamic neural mechanisms in the brain match different cues from multimodal events, meaning that we are not completely dependent on perfectly synchronized sensory data across the different modalities to create multisensory coherency (King, 2005).

Since temporal synchrony is a particularly strong binding, King explains, our perception system will automatically perform intermodal compensation on sensory input. Humans are for example able to accurately determine if visual and auditory cues occur simultaneously, despite the potential variations in arrival times at the respective modalities. I initially had a hypothesis that anticipating audio cues might help in preparing certain goal-directed actions, but this seems to be disproved by our capacity of intermodal compensation. This means that even if there are rich and informative auditory cues, they will never be able to act alone as influence on the users, but needs to be seen in perspective with other modal information. Further implications of design strategies and theory will be discussed in Chapter 4.

Even though the auditory modality can not be regarded solely by itself as a factor in perception within an interactive context, it might still be subject of manipulation. Careful use of sonic feedback has been proven that it can be used to optimize perception of quality offered by technologies (Dixon and Spitz, 1980) and improve the perceived quality of lower quality visual displays (Storms and Zyda, 2000). In the same manner it should be possible to carefully design sound for interactive systems, to induce desired corresponding action.

2.3.2 Environmental awareness and motor cognition

As we established, the embodiment of music and sound perception can be derived from what is known as the motor theory of cognition. As part of an interactive system, it is important to recognize that the user – of whom we are evaluating the cognitive abilities – is present in a surrounding environment. This environment can be regarded as what Nordahl called a *region of exploration* (REX) (Nordahl, 2008). The REX is a a 360° environment of possible investigation and action. In a video game condition, this environment is commonly known as a *virtual environment*. A virtual environment involves the output of potential sonic, as well as visual and tactile stimuli. It is important to look at how we corporeally interact with this environment to be able to see if there can be created a relationship between sound and motion in the design of interaction systems.

2.3.3 Proprioception and kinesthesia

The user will, within the boundaries of an interactive environment, have *a sense of joint position* and a *sense of movement* of his or her own body. Charles S. Sherrington (1906) introduced the term *proprioception* about the sense of the positions of the joints in relation to the body.

The term *kinesthesia* was coined by Henry C. Bastian (1880, p. 543) and can be understood as the *sense of movement*. These models are important for the user's feeling of presence and capability of navigation, especially in an environment relying on large-scale body motion. Further, the models are considered central components for muscle memory and hand-eye coordination.

2.3.4 Body image

Relating to the notions of proprioception and kinesthesia is the term *body image*, used by Godøy and Leman (2010, p. 8–9) as part of a musicological approach to motion theory. Body images represent our mental awareness of our actions in correspondence with surrounding environment. This includes an “offline” (non-realtime) concept of global gesture, or “online” (realtime) awareness of gestures.

An important aspect of this model is in regard to *structured interactions*. The awareness we keep of our actions allows us to chunk perceptions of motion into hierarchies of action patterns (i.e. kinematic sequences). Action patterns can then be regarded as single units, acting as parts of a bigger structure of corresponding patterns. Bob Snyder explains how we perform *chunking* of not only gestural, but all perceptual data in (Snyder, 2000).

2.4 Mental imagery of sound and motion

As we established in Section 2.3, the construction of multimodal events relies on *mental images* of these events. Derived from this, we can assume that mental imagery of motion is part of the multimodal activation process. Based on various sensory input, we create mental images of action gestures derived from prior ecological knowledge. The term image can here be understood as mental imagery, what Nigel J. T. Thomas explains as signifying and superficial perceptual encounters that mirror real perceptual encounters when the original stimuli is missing (Thomas, 2008). These encounters can be regarded as conceived kinematic chains or sensations of effort and dynamics (Godøy, 2003, p. 318). Sound and musical imagery can be understood as a cognitive capacity for imagining musical sound even when the original audible sound source is missing (Godøy and Jørgensen, 2001).

A pinnacle from the works of French *musique concrète* composer Pierre Schaeffer (1910-1995) was what he called *sonic objects*; short fragments of sound, typically within a few seconds of duration. These sounds are holistically perceived, and typically originate from a single cause (e.g. a breaking glass). Within the duration of the sound, several feature evolutions can exist (e.g. timbral, textural, dynamic). As a direct inspiration of Schaeffer’s work, Godøy advocates that the perception of sonic objects is closely linked to gestural concepts (Godøy, 2010). Sonic objects were considered useful compository tools for Schaeffer and musique concrète composers, as well as a following tradition of electronic and electro-acoustic music. Godøy argues that these indeed are *musical objects*, and will be able to act as parts of a bigger musical structure, or as discrete sound effects. Sonic objects are relevant and important to take into consideration for interactive sound design, especially since they can induce images of gestural information.

2.4.1 Gestural-sonic imagery

Translated into the physical domain, the sonic images might be converted into motions related to musical features such as onsets, timbres, etc. in the audible stimuli. Godøy explains how we continuously recode musical sound into what he called *multimodal gestural-sonorous images* (Godøy, 2006, p. 153).[1] Inspired by this idea, he further proposed that our mental imagery of musical sound can be founded on a continuous mental "tracing" of significant features describing the sounds we hear. The features that are traced are dependent on how we perceive and process the sounds. Godøy believes that it is possible to detect foundations sound-motion relationships especially through the energy features of the sound, or what is normally referred to as the sound's envelope.

Motion corresponding to the musical features, are often related to *sound-producing actions* (Godøy, 2006, p. 149). In the same way that we recode mental images of sound into motion, we will also be able to imagine certain sounds, by performing a corresponding sound-producing action. Schaffer's categories of excitatory gestures corresponding to sound-producing action might be considered:

Impulsive excitation is a single effort followed by a rebound, e.g. hitting a drum.

Sustained excitation is a continuous effort, e.g. violin bowing.

Iterative excitation is a repetitive effort, often merging into what seems as one sound, e.g. a drum roll.

It is expected that a person who perceives sounds associated with the respective sound-producing action is likely to ”visualize´´and possibly mimic motion based on these models.

2.4.2 Body schema

How we structure actions in relation to the surrounding environment can bee seen through what is known as *body schema*. As a centerpiece in his research on cognitive psychology, Neisser argued that

> " [a] schema is that portion of the entire perceptual cycle which is internal to the perceiver, modifiable by experience, and somehow specific to what is being perceived. [...] [I]t directs

[1] Godøy originally referred to the term *sonorous* in earlier publications. He has later abandoned this in favor of the term *sonic*.

> movements and exploratory activities that make more information available, by which it is further modified" (Neisser, 1976).

Body schemata can be explained as automatically triggered motor programs we use in our interaction with the environment (Godøy and Leman, 2010, p. 8). Included in these are automatic reactions such as grasping a glass of water, or catching a ball that is thrown at you. These motor programs require little or no mental processing, and once an action is initiated these programs can appear to carry out muscle-functions without our awareness. Motor programs are learned through repetition by watching gestures performed by others. Ecological knowledge of how to interact with our surrounding environment is embedded in body schemata.

2.5 Action-sound relationships and couplings

Following the ideas from motor theory of perception and mental imagery (Section 2.4), we know that sound can be perceived and stored in our memory as simulations of the sequences of actions leading up to the production of the sound. We can regard this as ecological knowledge of links between sound and motion. By connecting this knowledge to new perceptual input, it is possible to imagine that sounds can induce certain actions. It is still necessary to consider that the link between action and sound can be divided into action-sound couplings and action-sound relationships. The differences between these are explained by Jensenius (2007, p. 21–33). The action-sound *couplings* we make are naturally mechanically mapped, e.g. the sound that is produced by striking a piano key. Perceived action-sound *relationships* however, can also include artificial relationships, e.g. the sound that is produced by striking a key of an electronic piano.

Further, Jensenius argues that these relationships are strongly connected to our cognitive processing and that we take this knowledge with us when we encounter synthetic sound devices or virtual realities. Action-sound relationships can range from very weak to very strong, and it is only when they are strong that we might consider it a *coupling*. An *action-sound palette* might be understood as a span of various possible actions and the corresponding sounds. The action-sound palette is dependent on physical properties (size, shape, material, etc.) of the objects, and mechanical properties (distance, speed etc.) of the action.

2.5.1 Action-sound relationships in objects

In their discussion of interaction between the mind and physical world, F.J. Varela et al. propose the idea that audible stimuli can be regarded as action-objects (Varela et al., 1991). This model, together with Godøy's models on imagined actions presented in Section 2.4, can be combined with Jensenius' model on action-sound relationships to understand how sound can be perceived as gestural sensations in a virtual (video game) environment. Jensenius argues that the action-sound couplings in mental imagery also will be valid in our perception of artificial (virtual) action-sound relationships (Jensenius, 2007, p. 27). These relationships are based on a virtual object-action-object system for action-sound relationship knowledge.

Considering a virtual reality, action-sound palettes could of course be limitless. If the goal of the interaction experience is to create lifelike and natural motion-interaction, it would be necessary to use correct couplings in the sound design. At the other end of the scale, it can be surprising and fun for the user if the action-sound couplings in the design are completely unnatural, but the result could be a confusing motion-interaction experience. We can thus say that the weakness or robustness of the action-sound relationship in the artificial environment can depend significantly on the sound design. From this we can derive that video game sound is more comprehensible if sound was designed so that real-world properties of objects are matched. In addition to this, sounds caused by the user would also seem more comprehensive if action properties of the sound-producing action are matched in correlation with the real-world properties.

2.5.2 Object-action relationships

An object presented in the gameplay can affect how we move. Upon hearing a sound, we always possess prior knowledge of the sound within the environment it appears. Various research has been performed on our capacity to recognize physical properties of audible input. An overview of this is provided by Rocchesso and Fontana (2003). The understanding of the objects and actions involved in producing a sound can be presented with an object-action-object system, as by Jensenius (2007, p. 22). From interaction with objects in the daily life, we gain an experience of acoustic features, based on e.g. size, material, and surface, in the objects involved in the production of sound. This of particular relevance, since we in gameplay are presented with virtually constructed object-action-object systems.

Since the Kinect sensor allows for more natural motion in the interaction process compared to traditional handheld controllers, the effect of prior knowledge of these systems will be stronger. Considering that we have a mental imagery of motion, Godøy shows that how we recognize sound-producing actions also could be based upon motor images of a sound excitation (Godøy, 2001). Furthermore, we are also able to create images of the sound source's material resonance. Considering how actions might be affected by knowledge of an object's acoustical features, it is important to assign carefully designed sounds to objects presented in an interactive system. If these sounds are lifelike and natural, the interaction will also feel more natural.

Another way to observe motion in objects is through the notion of *affordance*. Derived from Gibson's ecological knowledge, and in particular based on knowledge about action-sound relationships, it is possible for objects, as well as for sounds, to contain affordances. Affordance can be compared to the notion that if we see a chair, we possess knowledge of its use (i.e. it can be used to sit on). The chair's *gestural affordance* can then be said to be the action of sitting down. Godøy (2006) explained how gestural-sonic objects implies that sound-induced movement share many properties with the corresponding sounds. Models based on affordance can directly account for relations between sound and action.

2.5.3 The action-reaction cycle

In order to understand how sound can affect our immediate reaction movement in an interactive process, we can examine the *action-reaction cycle* related to sound (Leman, 2008; Godøy and Leman, 2010). The model, derived from the cognitive research of e.g. (Neisser, 1976), continuously consider action features embedded in perceived sound. If we consider a performer playing an instrument, an example of an action can be plucking a string and causing physical vibrations in the air. As the vibrations are picked up and processed by the human perception system, the performer will react to this, make a judgement of the action related to the perceived sound, and possibly adjust physical parameters before the next action is executed. This model is vital for the understanding of how we can use sound to adjust actions in the interaction process.

2.5.4 Entrainment

To perform well by moving between obstacles and goal-objects, it is fair to claim that the right rhythm between the actions carried out is important. Sound can influence rhythm in motion through *entrainment*. Entrainment can be explained as synchronization between two or more independent rhythmical or pulsating systems (Clayton et al., 2004, p. 2). This phenomenon is rooted in studies of biological, physiological, and cultural rhythms. Entrainment can happen between non-human processes, e.g. metronomes, and also in interpersonal processes. In this thesis however, the most interesting effect is the synchrony between a person's body or body-parts, and the music and sounds in the interactive system. As Leman (2008, p. 71) explains, this might originate from biological resonances that is used in survival mode to transfer "physical energy into action-relevant concepts". This is an ecological model that places action-perception processes as a central function of how humans interact with the environment. Clarke (1999) suggest that pulse and rhythm in music can generate (involuntary) movements. This can result in tapping of feet or hands, nodding with head, or moving other body-parts in synchrony with the music. Essentially, a pulse within the music will be able to affect the tempo of periodically repeating actions.

2.6 Mapping

To be able to practically exploit knowledge of music and motion models in sound design, it is necessary to consider how sound and control of soon is *mapped*. In the traditions of NIME and SMC, mapping is typically defined as the "process of relating the elements of one data set onto another" (Hunt and Wanderley, 2002, p. 98). In the design of *digital music intruments*, this often means the linking of action inputs to control parameters. The discrepancies found in these links are one of the main challenges for mapping designs (Thelle, 2010, p. 26). It is possible to consider four types of mappings (Miranda and Wanderley, 2006, p. 15–16):

- *One-to-one* is the mapping of one input action to one control parameter.
- *One-to-many* is the mapping of one input action to several control parameters.
- *Many-to-one* is the mapping of many input actions to one control parameter.
- *Many-to-many* is the mapping of many input actions to many control parameters.

The many-to-many mapping model (demonstrated in Figure 2.1) seems to be what most acoustic instruments are based on (Jensenius, 2007, p. 101). As with mechanically mapped action-sound relationships, control parameters in an acoustic instrument are coupled. Performers tend to prefer the many-to-many coupled mapping model between a few action inputs and control parameters (Hunt et al., 2003). We will take these models into consideration for mapping designs involved in the exploration in Chapter 4.

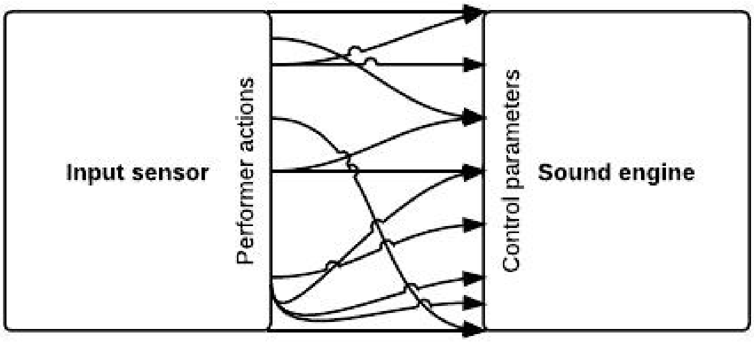


Figure 2.1: Many-to-many mapping (Jensenius, 2007)

2.7 Discussion

In this chapter we have examined the possibilities of considering relationships between sound and motion through embodied music cognition. It is possible to segregate musical features from continuous auditory input by applying prior ecological knowledge of the presented sounds. The motor theory of perception helps us to understand the link between sound perception and motion. Motor mimetic perception involves subconscious performances of the action we think was involved in the production of the perceived sound. These corporeal models enable us to consider sound producing actions as kinematic sequences that can be chunked, stored, and recalled.

The perception of sound, together with motor involvement, are parts of a multimodal perception system. We recognize multimodal processes as a convergence of unimodal events through what is considered multimodal mental images. The strong effect of temporal bindings help us synchronize events perceived by the various modalities. Stimulation of the auditory modality has shown to increase perceived overall quality when presented with lower quality visual feedback.

We have awareness of the positioning of our limbs and a sense of motion in relation to our surrounding environment through proprioception and kinesthesia. This is important to consider in an interactive context where full-body motion is regarded, as we will be interacting with the whole surrounding region of exploration. Our awareness with the surrounding environment is also shown through what is known as body image. This model allows the perception of our own motion into chunks, and organized into kinematic sequences.

As with perception of motion, we also use mental imagery in perceiving sound and music.

Sonic objects can be regarded as smaller chunks of a larger musical structure. We are still able to make out several distinguishable features from the sound, even if the sonic objects are of short duration. These features are closely linked to mental images of effort and dynamics, and explain how we can relate sonic objects to actions. In addition, we will often possess knowledge of the original sound-producing actions of a sound. Involuntary actions can be explained through body schemata (automatically triggered motor programs). Actions can then be evoked if a person is presented with a sound that he or she associates with an action "hard-coded" in the cognitive system.

The relation between action and sound can, depending on the strength of the relation, be regarded as either relationships or couplings. Action-sound couplings are mechanically mapped and thus perceived as having the strongest link. Action-sound relationships, however, can also include artificial relationships and are perceived as having a weaker link. It also possible to experience action-sound relationships in a virtual environment. The knowledge we hold about objects and their relation to sounds and actions is known as the object's affordance.

It is possible to say something about how we continuously adjust our actions in regard to sonic feedback through the concepts of the action-reaction cycle and entrainment. Through the action-reaction model of sound, we continuously evaluate and adjust our actions through sonic feedback. Entrainment can be explained as a more biological synchronization to a perceived pulse in the sonic feedback.

A user will control interactive devices by performing gestures that are *mapped* to the various actions the system is designed to perform. I will take sound and motion relationship models presented in this chapter into consideration as possible mapping solutions in the developments performed in the exploration part of this project. Certainly, it should be able to exploit such concepts as *body-schema*, *entrainment*, etc., by implementing relevant sound design. The implementation of this is what I in the introduction referred to as *sound-action mappings*, which will be discussed further in Chapter 4.

In reference to action-objects in perception, it might also be possible to base sound designs on these ideas. Although Leman argues that there is no immediate evidence of natural mappings between stimuli features and sounding objects (2008, p. 48), it seems like his argument is based on the perception of higher musical structures. If we consider more basic musical sounds, such as sonic objects, it should be more intuitive to work with mapping solutions. Following Jensenius (2007, p. 28), we might also assume that it is possible to bring knowledge about action-sound relationships into the virtual domain. This means that careful use of sonic objects in the design of e.g. game audio, can in fact make an impact on our choice of gestural action, and needs to be taken into consideration in design strategies explored in Chapter 4.

Chapter 3

Observation

This chapter presents an observation study of subjects playing motion based video games. First, method and conditions are presented, before the results are presented and discussed.

3.1 Case study on Kinect Games

A user-study was performed by recording motion capture data of subjects playing a variety of mini-games chosen from commercially available games for the Xbox 360 Kinect platform. Mini-games can be defined as one of several sub-games offered in a commercial game, often presenting only one task, and with little or no storyline. The research goal for this study was to analyze full-body human body movement in Kinect gameplay and to gain knowledge about whether or not music and motion relationships exist in the sound design of already commercially available games for the Kinect platform. The idea was that the data retrieved from this study would also be useful for determining if it is possible to detect a potential inter-subject *gesture repertoire* by studying inter-subject movements related to different tasks.

3.2 Method

3.2.1 Subjects

16 subjects, 5 girls and 11 boys, were recruited from personal and university networks, based on creating a diversity of musical and video-gaming background. The subjects were between the age of 19 and 39 and the average age was 28. To gain knowledge about the subjects, an initial part of the questionnaire presented to the subjects included questions about their background in

video games, music, and dance. It was possible to check off more than one option.

- 3 subjects answered that they had little video gaming experience, 9 subjects answered that they played now and then, 3 subjects answered that they play regularly, and 1 subject answered that he played a lot.
- 4 subjects had no musical background, 2 subjects were self-taught on an instrument, 2 subjects had basic musical education, 7 subjects had higher musical education, and 1 subject answered that he was a professional musician.
- 7 subjects answered that they had no training background for dance, 7 answered that they danced for fun, and 4 subjects answered that they had basic dance training.

3.2.2 Technology

In this study an *optical infrared marker based motion capture* (IrMoCap) system was used. *Motion capture* (MoCap) is the recording and digital storing of movement. It is commonly used within two main groups of applications; analysis or synthesis (Skogstad et al., 2010). The analysis approach is typical for medicine, rehabilitation, and sports research, while the synthesis approach is often used to create life-like animations for movies or video games.

A typical IrMoCap system consists of more than six cameras set up around the space of desired capture volume. The cameras emit infrared light, which is reflected off markers attached on the object of observation and again captured by the cameras. Each camera will record a two-dimensional image, but with the help of triangulation techniques the system can calculate absolute position in three-dimensional space. Triangulation can be explained as the calculation of a points location by measuring angles to the point from a known baseline. The point's location will be determined as the third point of a triangle with one known side and two known angles. IrMoCap systems are regarded as state of art for motion capture, since they perform at high speeds and with great accuracy and precision.

In this study, an OptiTrack system from NaturalPoint was used. The orientation of the axes in the data from the OptiTrack system are arranged so that the x axis is from left to right, the y axis is up and down, and the z axis is back and forth (see Figure 3.2.2). It is important to keep a good idea of the orientations, especially when we later will look at XY, XZ, and YZ plots of the subject's motion.

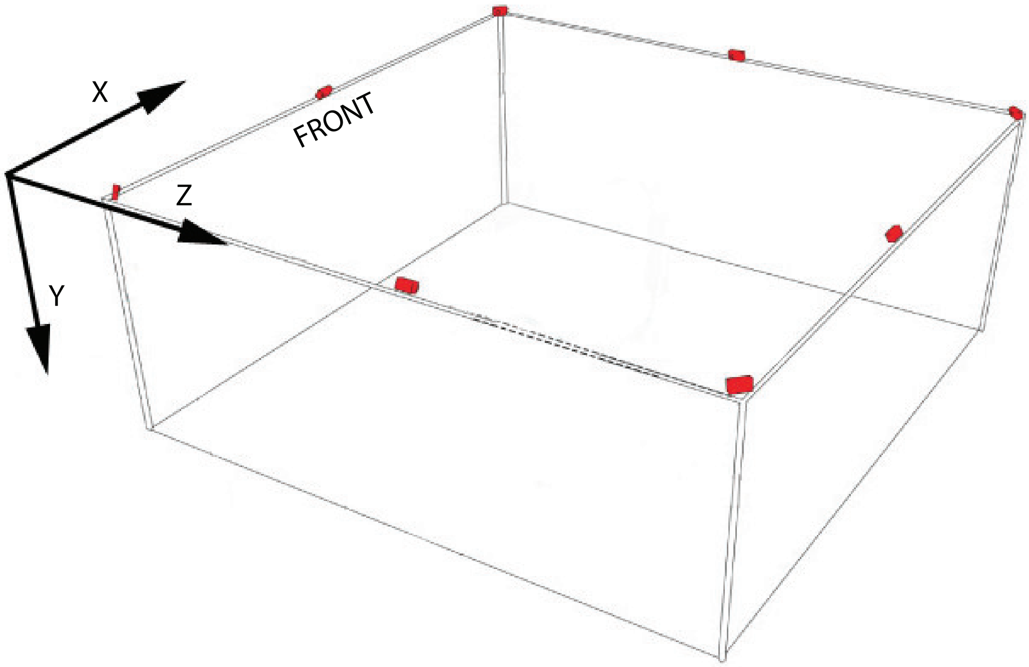


Figure 3.1: Orientations in Optitrack data

If three or more markers are combined in a fixed constellation, it is possible to identify certain unique objects. These objects are often referred to as *rigid bodies*, and allow detection of angular orientation data (how the object is oriented in space) in addition to absolute position. By assigning rigid bodies to several limbs of a subject's body, it is possible to combine these into a *skeleton model*. A skeleton model (also known as a kinematic model) takes the joint angles between rigid bodies into consideration, as well as the absolute position. This is an effective way to combine and label data sets, instead of being forced to handle large amount of single marker data.

The users wore a full-body OptiTrack MoCap suit, enabling the recording of 38 marker positions (see Figure 3.2). To be able to form a skeleton model, the markers were placed according to the setup described in the Arena (OptiTrack software) skeleton wizard. The Arena software will record two-dimensional recordings of marker positions from each camera. Later the two-dimensional recordings can be "trajectorized", performing a triangulation of two-dimensional recordings, into a three-dimensional recording. These recordings can be exported as .c3d files. The data was analyzed with the MoCapToolbox for Matlab (Toiviainen and Burger, 2011) and the .c3d format was the only possibility that was both supported by the Arena export function and the MoCapToolbox.

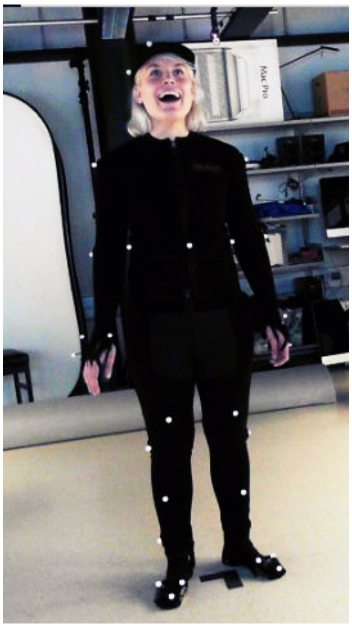


Figure 3.2: All subjects wore a full-body motion capture suit with 38 markers.

In addition to the MoCap recordings, video was recorded of both the screen and the subjects. Unfortunately, due to hardware limitations, there is no good way to record a direct video stream from the Xbox 360 while simultaneously projecting it on a screen. The video recording was performed by two Microsoft Life-cam HD web-cameras (see Figure 3.3), while the audio was directly routed from the Xbox 360 into an Echo AudioFire12 audio interface. The interface's low-latency direct hardware monitoring option was crucial for routing the audio signal to be presented for the users.

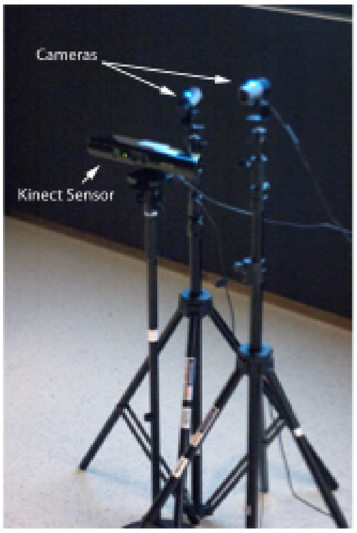


Figure 3.3: Overview of QoM for selected markers of all subjects in Rallyball

A patch was programmed in the Max[1] programming environment to help synchronize the video, audio, and motion capture recordings (see Section ??). This patch received a frame count from the Arena software, and enabled audio and video recordings as the recording button was pressed in Arena. The patch was also able to gather the MoCap data itself and store it to a text file, but I decided to work with the .c3d files so this option was left off in the recording process.

3.2.3 Task

Three different commercially available Kinect games were presented to the subject: Kinect Adventures!, Dr. Kawashima's Body Brain Exercises, and Kinect Sports. These games were chosen on recommendation from Johan Basberg, and was evaluated to represent the most representable games released for the platform at the time. This evaluation was based on the premise that the design of the gameplay they offered best represented the concept of the Kinect platform. At the time, there were not many available releases for the Kinect platform. The users were asked to navigate to a given mini-game within the presented game, choose this mini-game and follow the instructions presented on the screen. Five different mini-games were chosen according to consideration of what would present the subject with different kinds of "motion-tasks" (see Table 3.1). Other than being asked to play the games, no further instructions were given to the subjects. The subjects were asked to answer a short questionnaire after they had played the five sub-games.

Table 3.1: Tasks presented in gameplay

Game	**Sub-game**	**Task**
Kinect Adventures!	Rallyball	Small-scale movements: Arms and legs
Kinect Adventures!	Reflexridge	Large-scale movements: Side-steps, jumps, and ducks
Body and Brain Connection	Touch 'n Go	Dissociated directional movements of both arms
Body and Brain Connection	Traffic Control	Associated movements with both arms
Kinect Sports	Track and Field	Synchronization, timing, velocity in both arms and legs

3.2.4 Games

The subjects were first presented with two mini games from Kinect Adventures!. Rallyball (Figure 3.4a) places the player on a court designed as a rectangular hallway with a wall in the

[1] http://cycling74.com/

end. In the front of the wall the game presents different formations of wooden crates and static or moving goal objects. The object of the game is to serve a ball and hit the presented crates and goal objects before the time runs out.

Reflexridge (Figure 3.4b) is a game where the player stands on a rail tricycle. By jumping up and down, the tricycle will travel faster. The goal of the game is to avoid approaching obstacles and collect objects to score as many points as possible as fast as possible .

In Dr. Kawashima's Body and Brain Exercises the subjects were presented with two more sub-games. Touch'n Go (Figure 3.4c) asks the player to control two characters known from Pac Man by moving both hands within two separate confined spaces. The goal of the game is to keep the characters away from the chasing "ghosts".

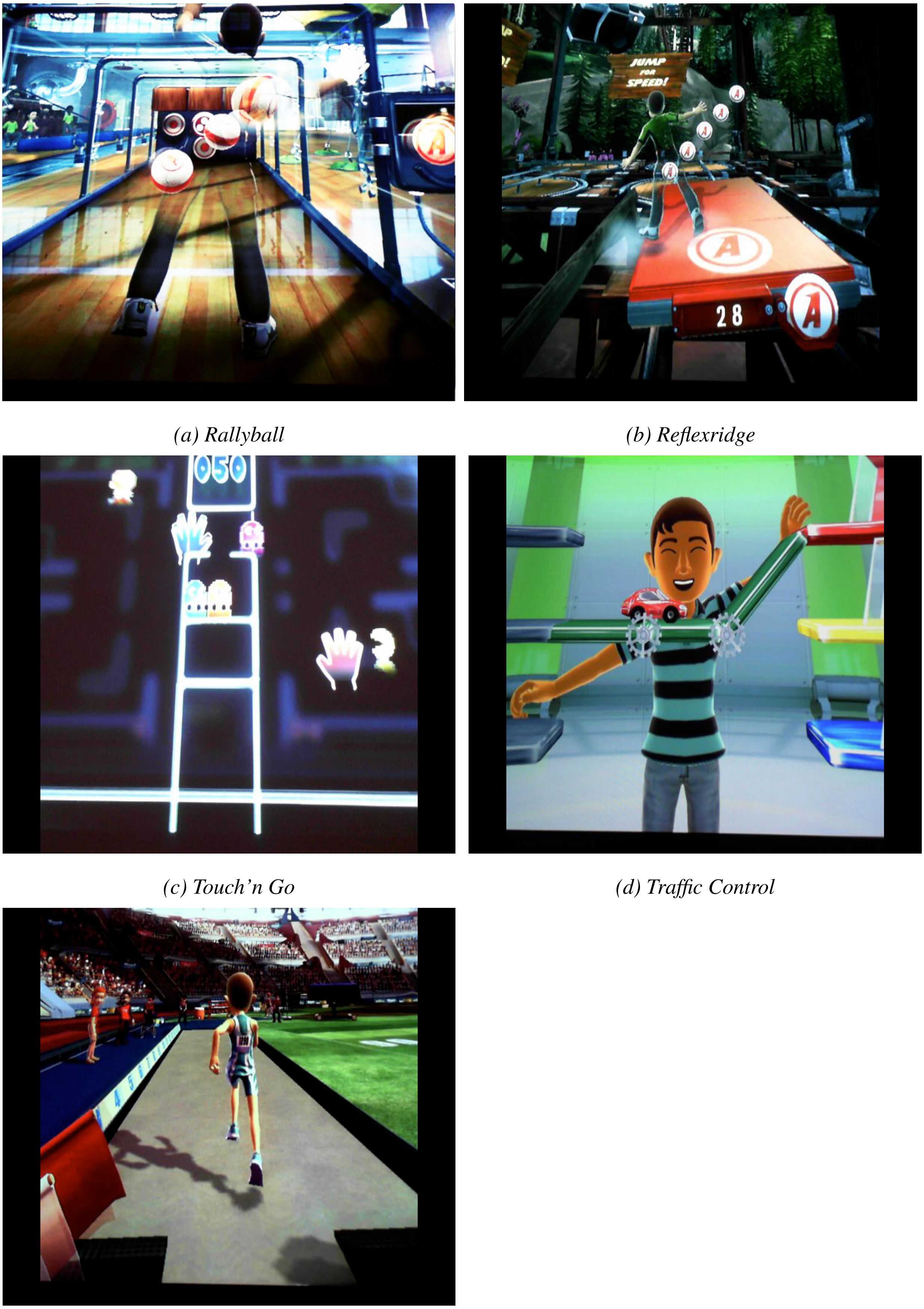
Traffic Control (Figure 3.4d) places the player in the middle of the screen with three platforms in respectively head, torso and waist position on both sides. The three platforms on the right side of the screen are colored in red, blue, and yellow. Three different cars in the same colors are presented randomly on the left platforms and the goal of the game is to position your arms so they form a bridge that will lead the right colored car to its belonging platform.

Finally, the subjects were presented with the Track and Field game from Kinect Sports (Figure 3.4e). In this game, the subjects compete in five disciplines; Sprint, javelin, long jump, discus, and hurdles. The obvious goal of this game is to perform as well as possible in the different disciplines.

3.2.5 Preprocessing

Initial challenges early emerged concerning compatibility between the .c3d format exported from NaturalPoint Arena software and the script for reading .c3d files in the MoCapToolbox. When loading longer files, the frame count would appear as a negative number. After some troubleshooting, it seemed the problem was how Arena coded the exported files. To be able to read the files, a modification needed to be done to the readc3d.m script in the MoCapToolbox (see Section A.1.1). What initially was declared as a signed integer, needed to be changed into an unsigned integer. While signed integers are able to represent negative numbers, unsigned integers will only represent non-negative numbers. The following modification was made in the code:

```
H.EndVideoFrame =fread(fid ,1,'uint16 ');
```



(a) Rallyball

(b) Reflexridge

(c) Touch'n Go

(d) Traffic Control

(e) Kinect Sports

Figure 3.4: Screenshots from games

After being able to import the files and performing rudimentary analyses, it was evident that the data still did not produce clear results. I had to go back to the Arena software, where a closer examination of some recordings showed errors in the marker position data. Errors included marker swaps, where two close markers would swap identifier names for a certain time, or marker identifier drop-outs. In the case of a marker identifier drop-out, the marker will for the period of the drop-out be named “Unidentified” followed by a number. A marker swap might only happen for a short time, and then swap back to original position. In an unlucky situation, the swap might be over a longer time, and it might swap with yet another marker, and not back to original position.

The reasons for such errors can originate from many factors, e.g. markers placed too close on the suit, poor calibration, poor lighting or camera interference. Arena includes an editing tool for the marker position data. This tool will do simple plots of each marker’s position on the x, y and z axes. It is possible to display the plots of as many markers as desired, to compare them. It is possible to perform correctional operations on the displayed plots, such as swap fixes, identifier fixes, gap filling, etc. Performing these operations on cluttered data is however an immense and time consuming task.

3.2.6 Analysis

Due to the unforeseen preprocessing workload and time limitations from also working on the practical cases of this master, I chose to focus the analysis on the first mini-game (Rallyball). This does not allow a comparative qualitative analysis across games, since only the MoCap data from this game is considered. However, it allows an inter-subject comparison of this game, since all subjects played the game.

To gain a perspective on the global movement of the subject, we calculate and compare the *quantity of motion* (QoM) of all subjects. This can be done by using the *mccumdistance* function of the MoCapToolbox to calculate the distance traveled by a marker, and dividing this by time. A script was written to perform this calculation and write out a box plot displaying the QoM of each marker (Section A.1.2). Some afterwork was applied in Adobe Illustrator to correct X axis labels. A box-plot includes five-number summaries (from bottom): Minimum value, lower quartile, median (red line), upper quartile, and maximum value. The median will split the results from the dataset in two, while the upper and lower quartiles will respectively represent the 25th and 75th percentile of the dataset. The percentiles are variables that split where a certain percent of the observation falls. Displaying the data in such a manner is helpful for indicating dispersion and skewness in the dataset. Minimum and maximum values are displayed by the whiskers growing out from the percentile box. The plus signs indicate outliers, numbers that are highly deviant from the rest of the dataset. The script provided is ready to use for analyzing recordings performed and exported from the Arena software. Another interesting aspect is the subject's limb-to-task choice. With *limb-to-task* choice I here mean the part of the body a user chose to perform the action demanded by a certain task. Since the plot shows separate QoM values for the different markers, it is possible to say something about the use of different limbs.

To be able to tell something about trajectory directions of the subject's actions, it is necessary to approach the data from a more qualitative approach. A qualitative study of the data can be facilitated by plotting marker position data over time. A script (Section A.1.3) was designed to create plots of the subjects motion in three planes: Transverse, sagittal, and coronal. The transverse plane can be explained as looking from over the subject's head and down, the sagittal plane can be explained as looking at the subject from the side, and the coronal plane can be explained as watching the subject from the front. These perspectives are gained from combining position data from the XY, XZ, and YZ axes respectively. All markers are left in the plots since we are interested in looking at the motion of the whole body. The plots are automatically scaled, something that will hide the extension of the motion in the room, but rather give a normalization of the motion. I personally think this is a good way of displaying the nature of motion for a subject, disregarding the subject's body size and natural reach. The script is a modification of a script created by Alexander Jensenius to match the data in this observation. It is now possible to use this script to analyze recordings performed and exported from the Arena software.

A discussion of missing analyses that would be necessary for further work is provided in Section 3.4.1.

3.2.7 Questionnaire

The subjects were asked to answer a short questionnaire after playing the games. In addition to asking about video gaming, musical, and dancing background, the questionnaire also asked about the experience of the games. These questions were particularly aimed at the motion aspect of the games, and all of them were a rating from 1 – 5.

1. The first question asked to what degree the subject payed attention to the music while playing.

2. The second question asked to what degree the subject payed attention to the sound effects while playing.

3. The third question asked to what degree the subject experienced the difficulty of performing tasks with the arms.

4. The fourth question asked to what degree the subject experienced the difficulty of performing tasks with the legs.

5. The fifth question asked to what degree the subject experienced the difficulty of performing tasks with the whole body.

6. The sixth question asked to what degree the subject experienced the difficulty of performing tasks with separate body parts in separate directions. This question was especially directed to the tasks presented in the mini-games from Dr. Kawashima's Body and Brain Exercises.

7. The seventh question asked to what degree the subject experienced the difficulty of keeping up with the tempo as it increased.

8. The eighth question asked to what degree the subject experienced the overall difficulty of playing the games.

3.3 Results

The result of the QoM analysis for all subjects is presented by the box plot in Figure 3.5).

As can be seen in the plot, there are similarities in the amount of motion in the hip and head markers, as well as the leg markers. There are some deviation in the first chest marker, which is the marker placed at the C7 (commonly known as the neck). This dispersion can indicate some variation in the overall motion of the upper body, some subjects were standing more in place, while others would move around within the region of the gameplay. Most dispersion of QoM is shown by the arms and hands. This dispersion shows that the subjects had various tactics in the gameplay. Some subjects would leave their hands more in place as a blockage for the balls, while other would swing their arms to hit the ball. The subjects that hit the balls would be left

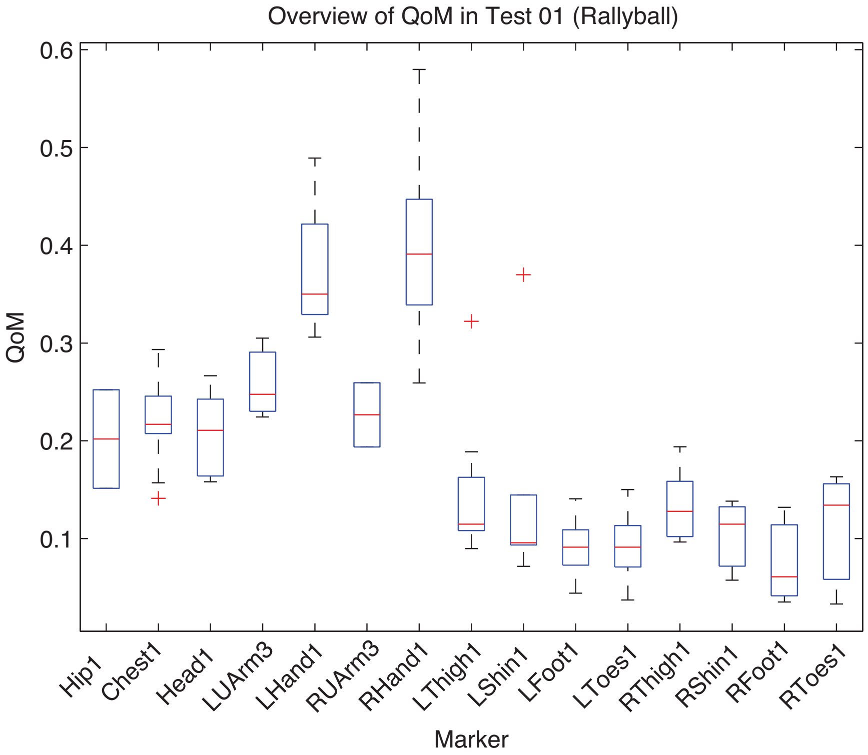


Figure 3.5: Overview of QoM for selected markers of all subjects in Rallyball

with more points, since this boosted the ball and caused more damage to the crates that was supposed to be destroyed.

The QoM figure will also be able to give us an idea of limb-to-task choice. As expected, the left and right hands are the most active. Even though the balls that the subjects are required to catch also will approach close to the floor, it is apparent from the plot that the legs are not widely used through the task. Looking back at the video recordings of the subjects, it is evident that many will duck down to catch the balls with their arms, instead of using their legs. It is possible to witness some inter-marker correlations in hip, head, and leg markers, suggesting an even distribution in motion between these limbs. The left thigh markers show a very high deviation marked by the outliers. These outliers indicate one or more subjects that have used their legs more actively than the others.

The next set of plots shows the marker position data over time for each subject through the Rallyball test. Figure 3.6 shows a plot seen from a transverse perspective (XY), Figure 3.7 shows a plot seen from a sagittal perspective (XZ), and Figure 3.8 shows a plot of each subject's motion through the same test, seen from a coronal perspective (YZ).

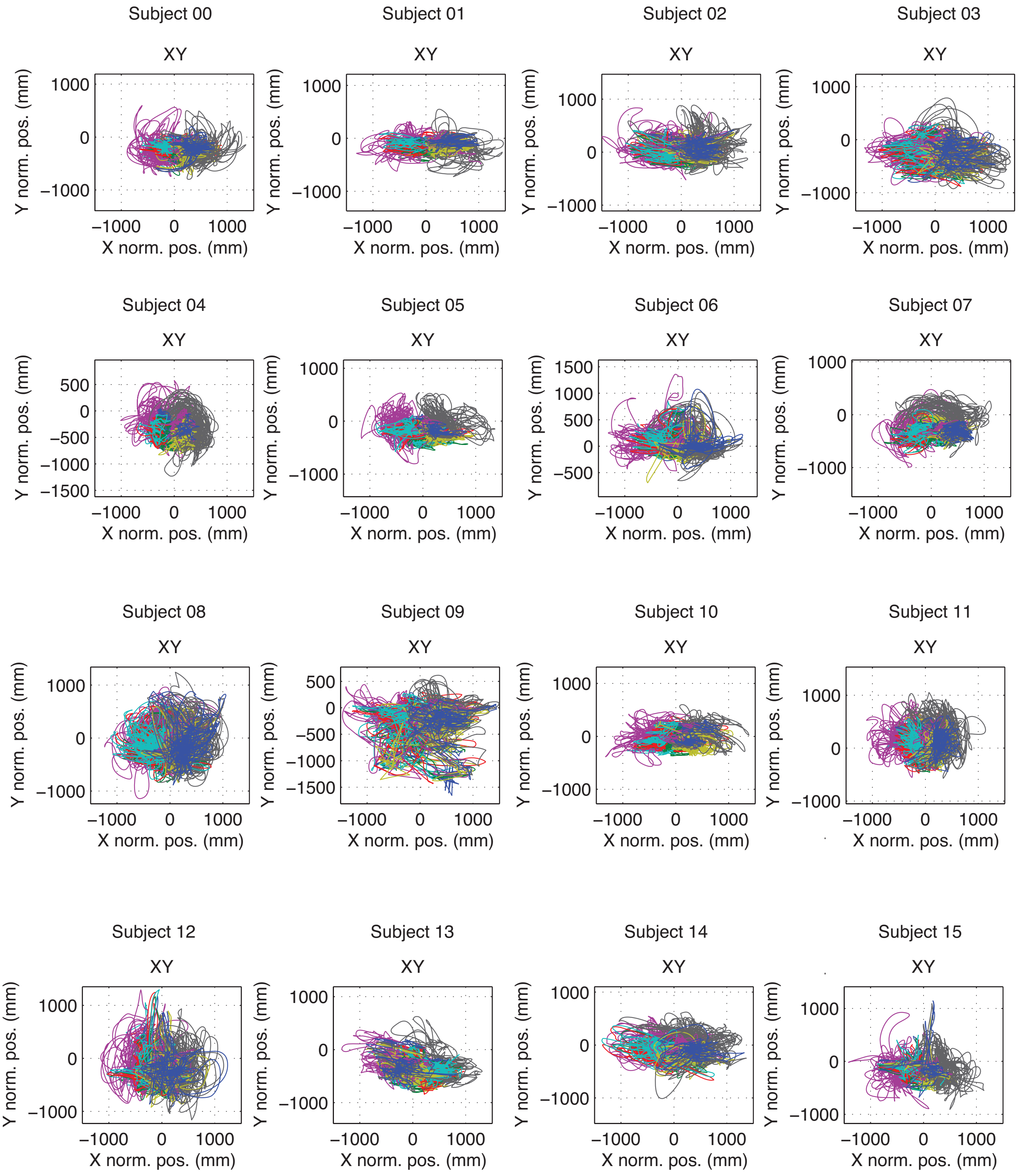


Figure 3.6: Plot of each subject's motion from the X and Y axes in Rallyball

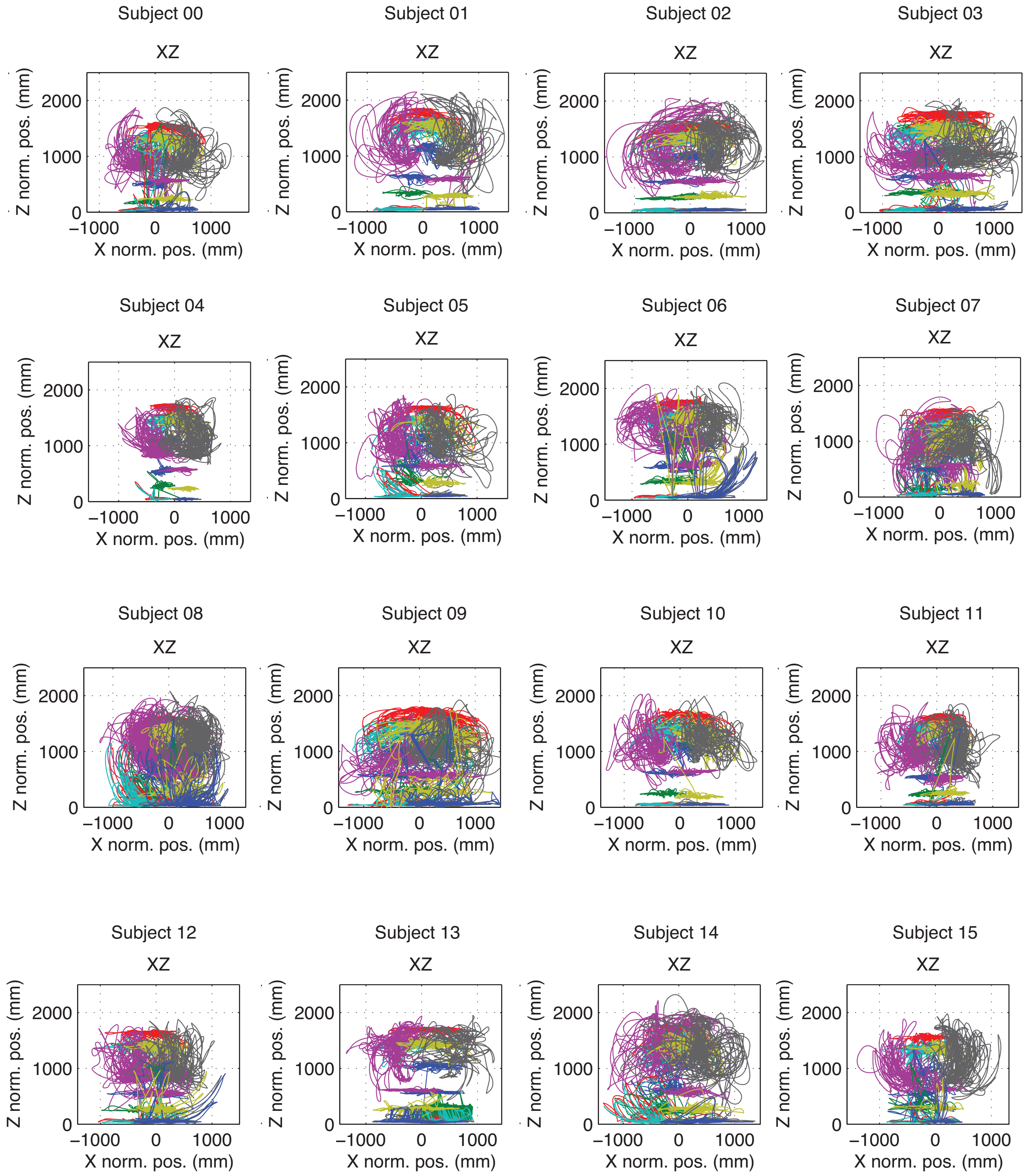


Figure 3.7: Plot of each subject's motion from the X and Z axes in Rallyball

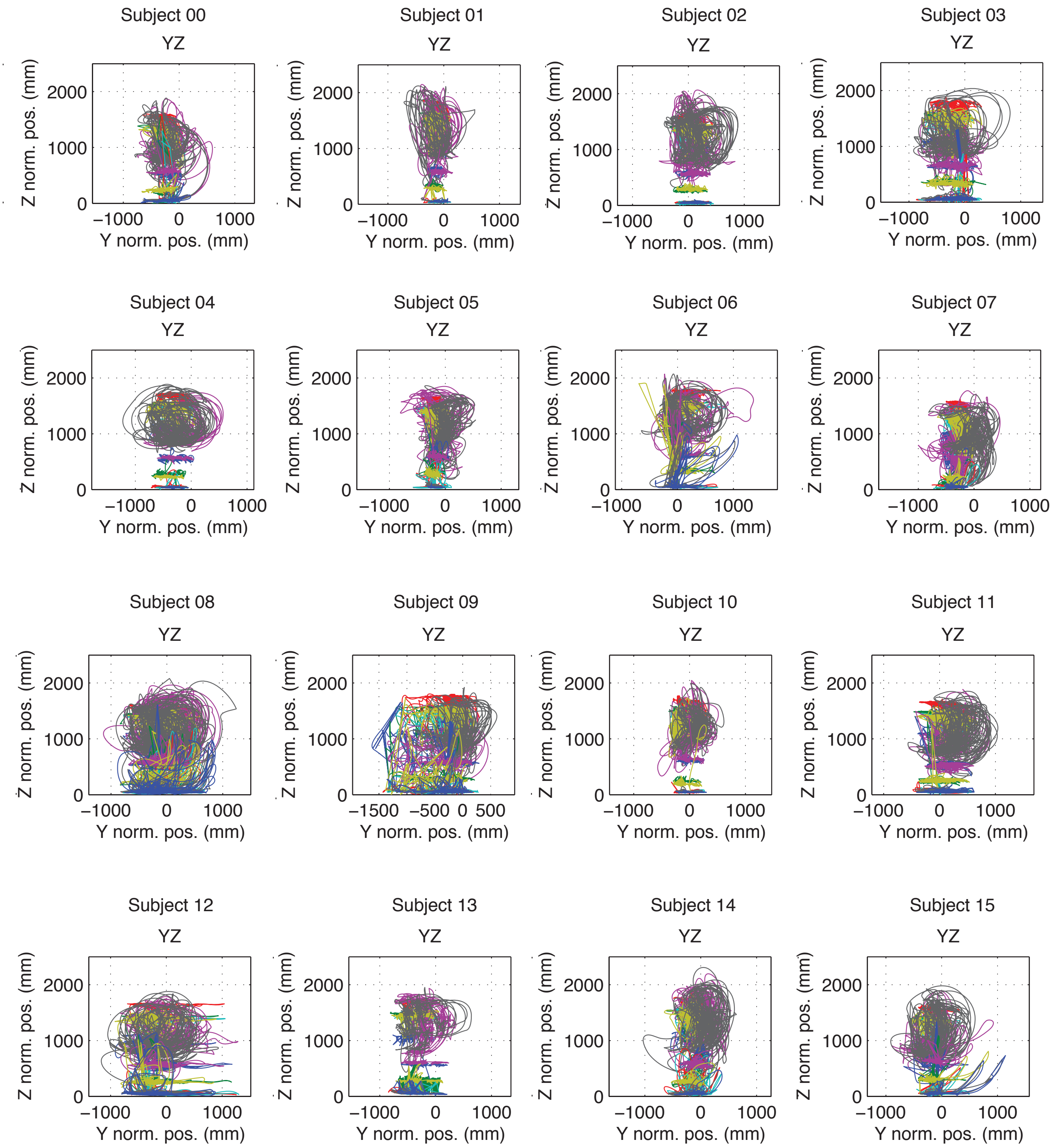


Figure 3.8: Plot of each subject's motion from the Y and Z axes in Rallyball

Looking at this test, it is possible to do qualitative observations of the trajectories and general motion patterns. Looking at the XY plots (seen from above), it seems to be roughly three types of motion. The first is horizontally stretched, indicating small amount of motion with the upper body and legs, using arms as mere blockage (e.g. Sucject_01 & Subject_10). The second is horizontally stretched with spikes towards the positive end of the y axis, indicating small amount of motion with the upper body and legs, but using the arms to hit the balls (e.g. Subject_02 & Subject_06). The third is a circular pattern, indicating an overall more active motion pattern (e.g. Subject_04 & Subject_09).

Looking at the XZ plots (seen from the front) it is evident that the arms play an important role, and most subjects move their arms in arched swipes or circular patterns on each sidhe of the body. On these plots it is also evident which subjects are most active using their legs (e.g. Subject_06 & Subject_12). It is very interesting to see how little some subjects moved the hips and legs (e.g. Subject_01 & Subject_10).

From these analyses, the YZ plots (seen from the side) were possibly the most interesting to study. In these plots, it is possibly to detect clear trajectories in different limbs. Obvious kicks are detected in Subject_06 & Subject_15. It is also evident that most subjects perform some kind of arched type motion toward the screen with their arms, but to very different degrees. Subject_01 has a very small arched motion towards the screen, while Subject_03 performs a deep arched motion.

3.4 Discussion

This chapter has described the user-study of commercially available Xbox Kinect games. Recordings of video, audio, and MoCap data were performed on 16 subjects playing five subgames of three different commercial Kinect releases. All recordings involved full-body motion. At the end of the recordings, the subjects answered a short questionnaire.

Analyzes of the data was performed using the MoCapToolbox for Matlab. As a measure of global motion, QoM was calculated by the distance traveled by each marker divided by time. It was possible to detect a similarity in the amount of motion carried out by the hips and legs of subjects. There was however, more deviation in the most active limbs, i.e. the arms. It was evident that the arms were clearly the most used limbs for this task, even when approaching objects were directed towards the legs.

By approaching the dataset with a qualitative approach, it is possible to examine the kinematics and trajectories of the subject's actions, and how the discrete limbs will act. This approach as facilitated by plotting the markers' position data over time, and visualized from the perspective of the transverse, sagittal, and coronal planes, through the corresponding axes.

To be able to relate this to sound design, we need to look at the background theory. As discussed in Chapter 2, the user of an interaction system will act through ecological knowledge and motor cognitive processes linked to the perceived sound. Considering limb-to-task choices, it might be difficult to imagine sound-design strategies that will facilitate this. It is, however, possible to imagine that previous knowledge of an instrument prominently presented in the musical mix might affect such a choice. This is because we hold information of which limbs are used to carry out the sound-producing actions on the presented instrument.

It has been shown that even people with no musical training have the ability to carry out air instrument performances with detailed knowledge about the real instrument (Jensenius, 2007). This is what is known as motor-mimetic gestures (Godøy, 2003). Even though further testing would be required, a person would theoretically be encouraged to be more active with his or her hands if e.g. a lead guitar part was prominent in the musical mix. The same might apply to a distinguished "four-to-the-floor" (hitting on every downbeat in a measure) bass-drum pattern playing on all down beats in a musical measure, if the task asks for a certain periodical vertical right foot activity. Trajectory aiming might also be related to mimetic behavior through what is known as goal directed imitation (Wohlschläger et al., 2003). This might also be explored through Pierre Schaeffer's ideas of sonic objects (Godøy, 2006) (see Section 2.4).

3.4.1 Analysis for further work

This chapter represents an examination of motion patterns in users of video games. However, no analyses are performed on the sonic material of these tests. What is most unfortunate is the lack analysis of music and motion coherence in the sound design. This needs to be subject for further work. Another especially interesting aspect would be to study the inter-subject correlation of synchronization accuracy. This would be an important aspect when closing in on correlations between performed action and sound. Some other aspects that would be interesting to analyze in further work are summed up in table 3.2.

Table 3.2: Examples of parameters to analyze

Motion features	Local motion
	Synchronization accuracy
	Hand flux
	Hand speed
	Hand distance
Musical features	Tempo
	RMS energy std
	Fluctuation entropy
	Number of onsets
	Pulse clarity

Chapter 4

Exploration

"Let's get physical"

Olivia Newton John

This chapter starts with an overview of the background of motion-based video game technology and continues with an explanation of the various explorations of sound and motion interactions that was performed for this thesis. A discussion of the findings concludes the chapter.

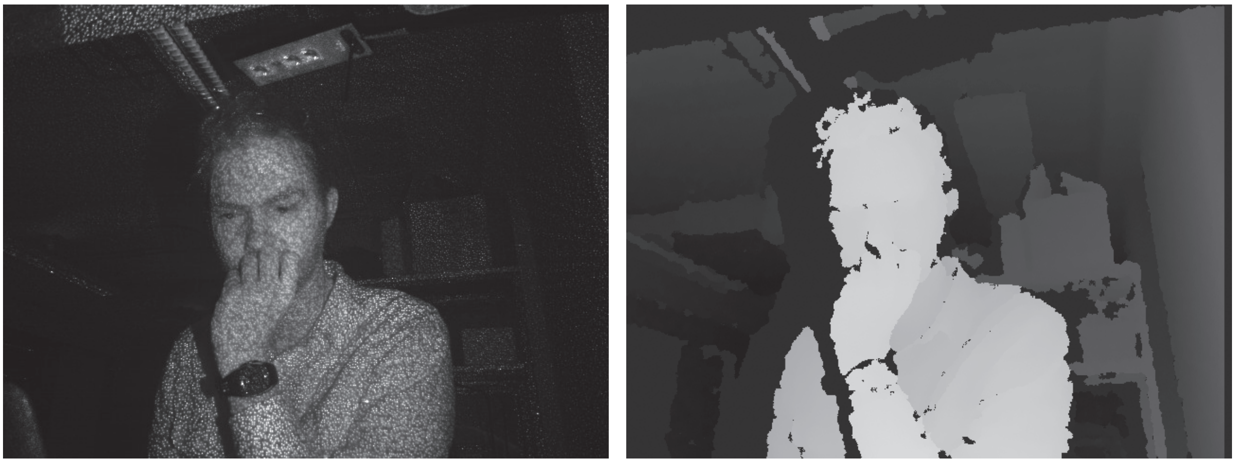
4.1 Introduction

Based on the theoretical framework presented in Chapter 2, and the observations presented in Chapter 3, explorations were carried out through the development of software and installations. The Kinect Piano and Popsenteret Kinect installations were set up at respectively Idéfestivalen at the University of Oslo and Popsenteret in Oslo. Two software prototypes, Soundshape and Music Kinection, was developed. The implementation in all cases was based on the Kinect sensor and focused on how sensor data from this device can be mapped to musical sound.

4.2 A Short History of Video Game Technology

Over the last decade there has been a tendency towards developing video game technology that is based on human motion input and gesture recognition algorithms. Sony released the first serious commercial attempt at a motion-based video game platform in 2003 with the EyeToy, combining traditional gameplay ideas with ideas of *computer vision* (CV) and so-called *gesture recognition*. The concept of EyeToy was developed by Richard Marks, who's idea was to create a video game environment where body movement was translated into game controls, allowing users to physically interact with the game without being attached by a cable (Marks, 2010). A different direction was introduced by the Nintendo Wii system, available on the market in 2006. Here, the physical interaction is conducted through infrared and accelerometer data, omitted from a handheld controller.

In 2007, Sony released the Playstation Eye, a successor to the EyeToy. Based on the same principles, the Eye included a sampling rate of 60 Hz at 640x480 pixel resolution and 120 Hz at 320x240, as well as a multi-directional microphone array. The higher sampling rate and resolution allows for more precise gesture detection, and, together with the microphone array's abilities to do vocal location tracking and noise cancelation, added a new dimension to the interactive experience. Sony introduced Playstation Move in 2010. This system includes a hand-held controller to accompany the Playstation Eye. The Move controller included inertial sensors, a three-axis linear accelerometer, and a three-axis angular rate sensor. These sensors can be used to track rotation and overall motion, and this data combined with the CV routines creates an intelligent motion capture technique.



(a) Illustration of the structured light technique performed by the Kinect

(b) The resulting depth map produced by the Kinect sensor

Figure 4.1: Kinect depth sensor

However, the Playstation Move technology was a step back from the idea of a controller-free game experience. In the same year as the release of the Move technology, Microsoft went in the opposite direction and introduced the Kinect sensor for the Xbox 360 platform. Taking advantage of an infrared laser projector and a monochrome sensor to capture depth images. Using a technique called *structured light* (Figure 4.1a), the Kinect is capable of reconstructing 3D representations of objects in front of it. The resulting three-dimensional analysis can be illustrated by the depth map in Figure 4.1b. It is this depth map that allows the recognition of discrete body parts, and the construction of a trackable skeleton model.

Kinect is currently the only game platform free of any form of traditional handheld controllers, relying completely on gestural input from the whole body of the user. This makes the Kinect an interesting device for the research in this project, offering a way to theorize the interaction between sound and corporeal involvement unrestrained by controllers or markers. Intricate full-body gestures can now be captured and processed by an inexpensive device within the walls of our own living rooms.

4.3 Soundshape

4.3.1 Idea

By the time of the development of the Soundshape prototypes, the Kinect sensor itself was in a pre-release prototype stage (under the name project Natal). Before the release of the Kinect, Microsoft released footage from demonstrations of the Natal, both proof of concept videos and from live demonstrations and testing at the E3 convention. The first Soundshape prototype was inspired by these videos.[1]

4.3.2 Implementation

I developed two prototypes, programmed in Max. The first prototype was based on a web-camera solution using CV and voice recognition techniques. Here, CV routines were programmed using cv.jit,[2] a library of external objects for Max developed by Jean-Marc Pelletier. Since this was a web-camera based solution, no depth sensing was available.

The next prototype I developed was using the Qualysis infrared marker-based optical motion capture system available at the fourMs laboratories. Now the prototype was able to receive precise position data from markers placed on the user. Three markers were placed in a fixed "satellite" formation attached to a glove (see Figure 4.2). In the Qualysis interface software,

[1] http://www.youtube.com/watch?v=g_txF7iETX0

[2] http://jmpelletier.com/cvjit/

QTM, the markers were identified and combined into a *rigid body object*. The position data (x, y, z) of this rigid body object was then allowed to be transmitted through the OpenSound Control (OSC) communication protocol. OSC gives many advantages in this type of programming, e.g. simultaneous streaming of data packets through UDP/TCP, avoiding the potential latency of the serial order MIDI protocol, and also URL-based address format that makes it easy to keep multiple input nodes systemized.[3]

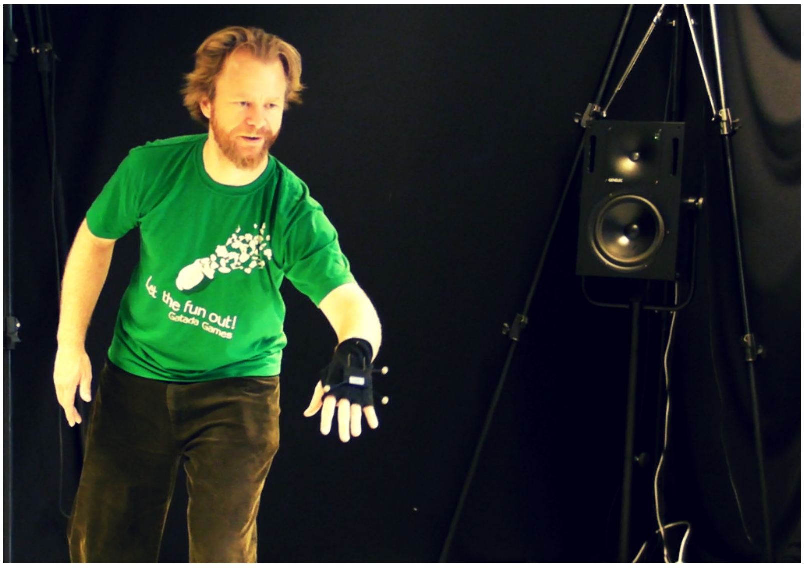


Figure 4.2: Johan Basberg testing the second Soundshape prototype

4.3.3 Usage

Both of the Soundshape prototypes presented a game where the user was supposed to look for an "invisible" object. The object was only discoverable by sonic feedback and the user had to guess what the object was supposed to represent based on audio cues.

In the first prototype, sonic cues were mapped according to positions in a two-dimensional plane. The object in the task presented in this prototype was supposed to be a bell, positioned upwards and to the side of the user. If the user reached straight to the side, the sound of swaying a rope (supposedly attached to the bell) would play. By reaching in the area of the bell's placement, a sound mimicking touching and stroking along the surface of a bell would play. If the user was to do a quick downwards movement right under the bell's placement, as if pulling the attached rope, a ringing bell sound would play. Finally, the user would guess what the object

[3] For more information about OSC, see Wright et al. (2003).

represented by saying the word out loud (in this case “bell”).

In the second prototype, the sonic cues were mapped in a three-dimensional space. The user would wear the glove with markers and explore the area. The region of exploration was significantly larger than the first prototype. A synthesized sound would be produced as the user’s broke through the “boundaries” of the object presented, and would keep sounding until the user’s hand went outside the object. For example, see the video **/videos/soundshape.mov** on the accompanying disc.

4.3.4 Evaluation

The CV routines implemented in the first prototype worked well and allowed for some exploration of the room based on the audible feedback. The space available for user exploration was however limited, particularly due to the two-dimensionality of the input. However, I believe that the interesting part of the interaction in this prototype was the nature of the sounds presented and the connection between them, and not the region of exploration in itself.

The second prototype allowed a larger region of exploration, and this became something interesting by itself. A user would walk around “in the dark”, and be completely reliant on the sonic cues suddenly presented. In this version, the object presented was only a rectangle. In further work on this prototype, it would be interesting to look at the possibility of loading and using coordinates for openGL models as boundary lines for the presented object. The sound design should also be developed further, possibly to match the presented objects as well.

The development of the Soundshape prototypes was, for me, an introduction to the use of MoCap techniques, both with rudimentary CV and a state-of-the-art system. I gained useful knowledge, such as how to define action areas within three-dimensional space and how to set up basic motion feature extraction (e.g. velocity and acceleration) in Max. For me, the focus in the development of these prototypes was not on motion and sound relationships in the interaction process, but on creating a foundation for software implementation that could facilitate such exploration.

4.4 Kinect Piano

4.4.1 Idea

As a part of Idéfestivalen at the University of Oslo, I was asked if it was possible to implement a Yamaha Disklavier (MIDI-controllable mechanical piano) residing in the fourMs laboratories in an installation setting. I decided to look into the possibilities of controlling this MIDI piano with a Kinect sensor, the process ultimately resulting in the Kinect Piano. The system required, in addition to the Disklavier, only an Apple iMac and a Kinect sensor. An interface software was written in Max to take the input from the Kinect sensor and turn it into MIDI note data.

4.4.2 Implementation

Given that this installation was to be placed in a crowded area, the user interface needed to be simple and able to react to interaction immediately, without any calibration necessary. I decided to use the Max external object jit.freenect.grab,[4] developed by Jean-Marc Pelletier. This object gains access to the Kinect's RGB image and depth map within Max, without the necessity of any additional drivers or libraries to be installed. Although this object doesn't recognize user skeletons or calculate position data (see Section 4.5), it will input the raw sensor data from the Kinect directly into Max. Through the access of the Kinect's sensor data, it is possible to perform more advanced CV routines than with a basic web camera solution. For the installation, I placed the iMac and the Kinect sensor on top of the piano (see Figure 4.3).

Exploiting the Kinect's depth map, it is possible to assign a delineated area on the z axis (from the sensor to the user). By doing this I could mark a position on the floor a certain distance away from the piano. The user was then able to stand at the mark without interacting with the piano. Once the user reached out his or her arm, the interaction would initiate. From here, CV routines based on the cv.jit objects were performed to calculate the position of the centroid from the depth map image. The software then produced monophonic (one-note-at-the-time) chromatic MIDI note-on data that was transmitted through an M-Audio Midisport 2X2 USB-MIDI interface and into the piano's MIDI input.

Two axes were converted into MIDI data; The position along the horizontal axis was converted to pitch data, and the position along the vertical axis was converted to MIDI velocity.

[4] http://jmpelletier.com/freenect/

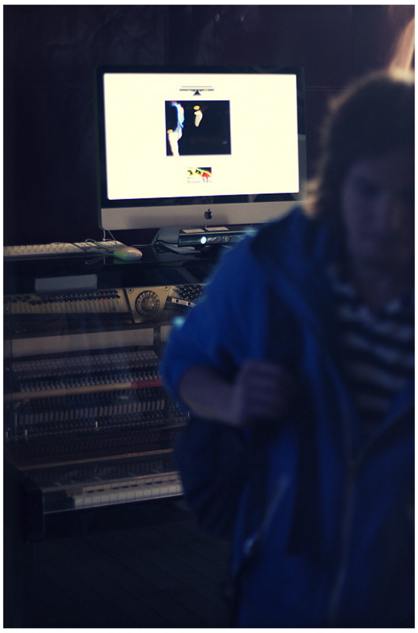


Figure 4.3: The Kinect Piano

This allowed the user to control both pitch and dynamics of the notes transmitted to and mechanically played by the piano. It should be said that even if the MIDI velocity is a dynamical parameter, it is very limited with only 128 possible outcomes. Since the software interpreted the centroid of detected motion, it would transmit the position in the middle if the user reached out two hands. This implies that the user would gain control and accuracy of the sound-production if only using one hand.

4.4.3 Usage

A simple GUI,[5] was designed where the user would get a simple image of themselves whenever they "broke through" the barrier that initiated the sound production (Figure 4.4).

In addition to the "mirror" image, the visual feedback included sidebars with markers indicating horizontal and vertical position. A mark was placed on the ground where the user would be in the best position for reaching the arms forward and evoke the piano to produce sound.

4.4.4 Evaluation

The installation was very successful. Children especially, but also many adults, tried the installation. The software and hardware worked exemplary, and never crashed during the time of

[5] Graphical User Interface

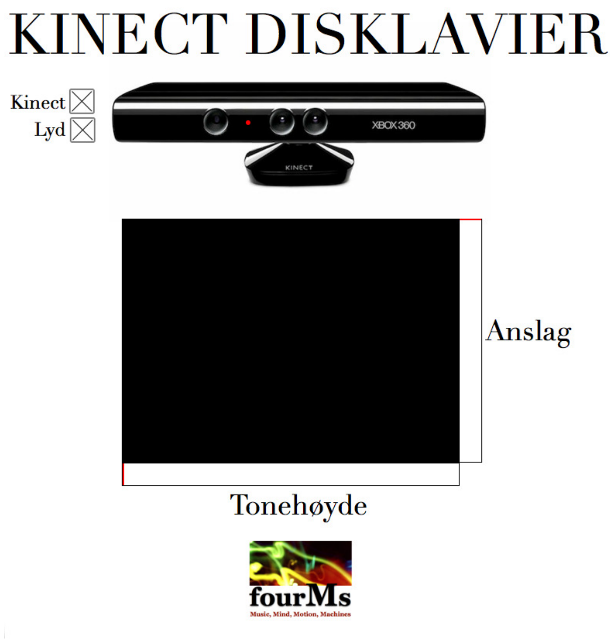


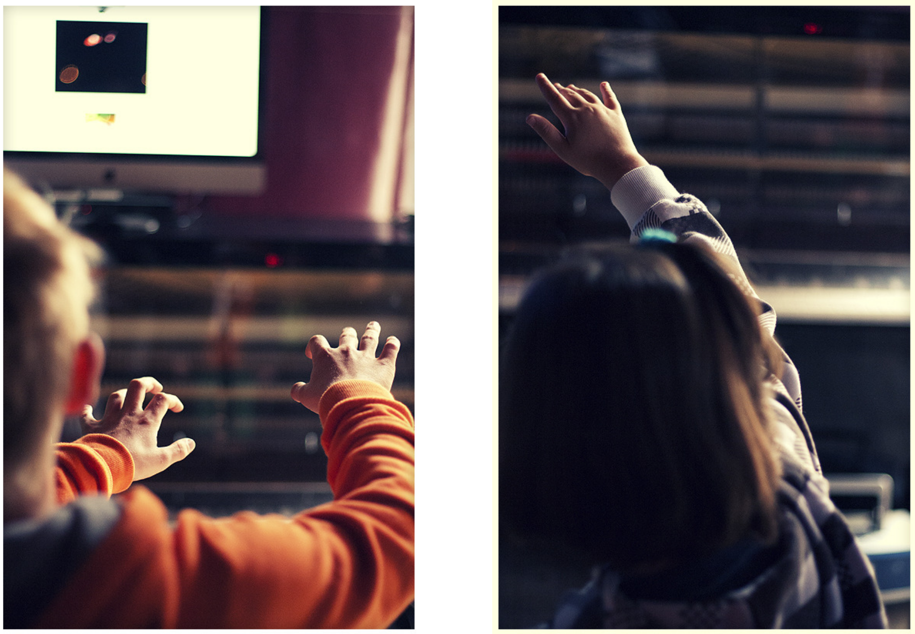
Figure 4.4: The Kinect Piano's GUI

operation. It seemed like the direction the playing hand aimed, if only one hand was presented, corresponded well with the key that would be played. Naturally, since the piano's affordance (as explained in Section 2.5) is suggesting a two-handed approach, many users would do this at first try (Figure 4.5a). Although, after discovering that this did not result in the control they expected, many would switch over to using only one hand. Many users, assumably with less musical background, would also approach the installation with a one-handed approach (Figure 4.5b). The one-handed approach suggests a musical control similar to that of *sound tracing* discussed in section 2.4. How this particular mapping evoked this particular action of the users was a very interesting observation.

4.5 Popsenteret Kinect Installation

4.5.1 Idea

I designed an installation based on input from a Kinect sensor at Popsenteret, a public exhibition center for popular music history in Oslo. With this installation, I wanted to work with full-body motion as input, as well as both sound and video as feedback for the users.



(a) Traditional

(b) "Tracing"

Figure 4.5: Children playing the Kinect Piano chose either a traditional approach (a) or more of a "tracing" approach (b).

4.5.2 Implementation

To be able for anyone to interact with the system, the installation is based on Microsoft's official Kinect SDK.[6] This SDK allows for detection of the user's skeleton without the necessity to perform a calibration pose (as discussed in Section 4.6.2). This means immediate involvement without any instructions necessary for initializing the interaction. The software KinectCapture2[7] provided the detection of the user's skeleton and transmission of the skeleton data (joint positions) via OSC. The OSC message was formatted as a bundle of six objects (SIFFFF)[8] with the name-space prefix "/joint" (see Table 4.1). This is a typical format of OSC messages, used by several of the Kinect interface softwares I have come across.

The OSC stream was gathered in a Max patch. When working with a full-body skeleton model, routing and organization of joint data input is an immersive task. A well organized routing patch allows for easy and intuitive prototyping of mapping solutions. The code includes a sub-patch where the OSC-route[9] external object routes the specific joint IDs from the OSC

[6] Software Development Kit

[7] http://www.908lab.de/?page_id=325

[8] Defining string, integer, or float objects

[9] http://cnmat.berkeley.edu/downloads

Item in list	Object
0.	Joint ID as String
1.	Skeleton Number
2.	Joint Position X
3.	Joint Position Y
4.	Joint Position Z
5.	Frame Number

Table 4.1: Hand movement to sound parameter mapping in Popsenter Installation

message address space, the data is scaled, and formatted into a new OSC message (see Figure **??** in appendix).

This is the first exploration where I wanted to experiment with both visual and audible stimuli to accompany the motion within the interaction environment. For the visual stimuli I used a ready made puppet, “doll_soft.nmt”, that was included as an example for the Animata software (Figure 4.6a).[10] Animata is a simple two-dimensional scene and animation design software, which is capable of receiving OSC messages. When provided with correctly scaled and formatted skeleton joint position data via OSC, Animata will be able to use the “doll_soft.nmt” puppet to mimic the user’s movements (Figure 4.6b).

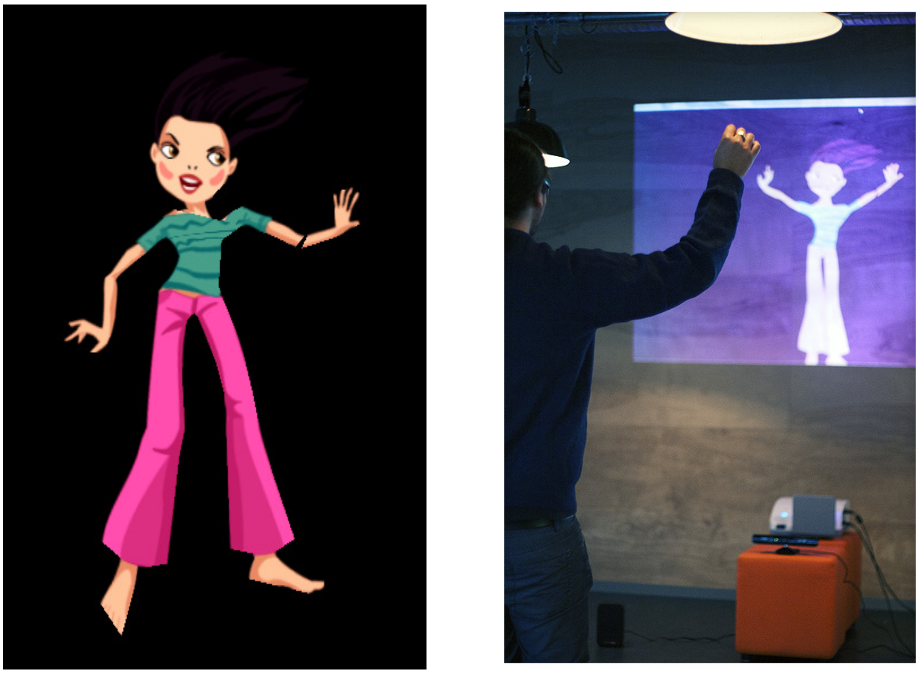
4.5.3 Usage

I decided to let the hands control the audible feedback in this installation. Both the left and right hand would control the same parameters, but be assigned to different sounds. The sounds were sampled synthesizer sounds, provided by so-called soundfont files, and played back by the fluidsynth~ external object.[11] In addition to the synth, there is a low-pass filter and an echo effect in the audio chain. Table 4.2 describes how the motions of the hands were mapped to sound parameters.

The hands’ position on the X axis (left to right) were mapped to MIDI note data that are quantized to fit a certain scale. A major pentatonic scale was set as default for both hands. Positions along the Y axis (top to bottom) were mapped to the cutoff frequency and resonance of a lores~ low-pass filter. This was supposed to give the illusion of where in elevation the

[10] http://animata.kibu.hu/

[11] http://imtr.ircam.fr/imtr/FluidSynth_for_Max/MSP



(a) Visual feedback by Animata

(b) Human control of Animata

Figure 4.6: Animata visualization

Hand movement	**Sound parameter**
Hand X position	MIDI note pitch
Hand Y position	Filter cutoff frequency & resonance
Hand Z position	Echo mix

Table 4.2: Hand movement to sound parameter mapping in Popsenter Installation

sound is placed, i.e. the lower the cutoff frequency – the lower the placement of sound. The positions along the Z axis (back and forth) were mapped to the mix of the echo signal into the audio feedback, to provide the user with an illusion of distance from the speaker to the audio stimuli.

4.5.4 Evaluation

This installation facilitated, like the Kinect Piano, an exploration of sound through “sound tracing”. The approach was here taken further, by applying it to two hands, and allowing more intricate manipulation of the sounds. Manipulation of the sounds involved filtering through Y axis motion and reverberation through Z axis motion. This could possibly have been explored further by applying more extreme settings, as the effects were not as apparent through the sound system at the venue where the installation was placed. Overall, this was not a successful installation. It was reported that the installation would break down after only a few hours of operation time. I found out that it was only the Animata software, running the visual feedback, that would crash. The software is free and open-source, so the crash could be due to that bug fixes are not thoroughly followed up.

Another possibility is that the Max software that routed the data and provided the sonic feedback was not sufficiently CPU efficient. I cleaned up the code in the newest version of the software, but due to time limitations I was not able to follow up whether or not this made a difference for the installation. A different problem emerged from the various versions of the Kinect SDK from Microsoft. I tested a couple of different Kinect data interpretation software, such as the KinectCapture2 I ended up using. It seemed like a mismatch between these softwares and the installed version of the SDK could cause problems in the system.

I believe that the impression of the visual feedback could have caused an overshadowing of the sonic interaction in the installation. This is a new technology and many users might not have experience with a detailed visualization of their own motion. Doing a similar installation with a different visual feedback, or without any, would be an interesting perspective for further research.

4.6 The Music Kinection Prototype

4.6.1 Idea

The intention of the Music Kinection prototype was to create a prototyping environment for exploration of sound design concerning music and motion paradigms in interaction systems. In addition, the prototype was supposed to be able to serve as a possible algorithm for the audio engine in the design of a game, or other interactive systems.

4.6.2 Implementation

The Music Kinection prototype was written in Max. The motion capture from the Kinect sensor, however, is performed by the independent software OSCeleton.[12] This software is based on the Kinect driver and code framework called OpenNI.[13] OpenNI is an open-source release by PrimeSense, one of the contributors of hardware for the Kinect sensor. The framework provides

[12] https://github.com/Sensebloom/OSCeleton

[13] http://openni.org/

a proxy for communication between the Kinect sensor and any programming language supporting the OSC protocol where discrete skeleton joints are tracked separately. The skeleton data is then streamed through OSC.

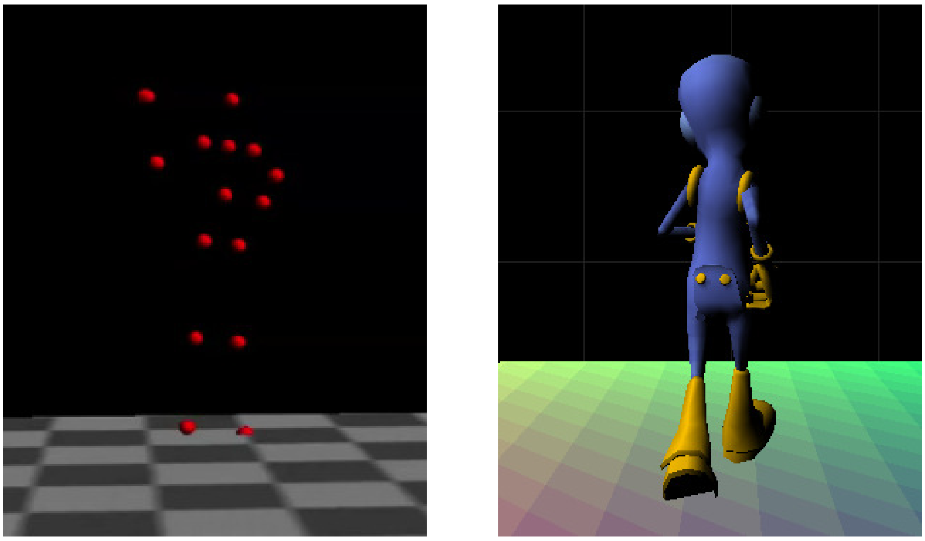
In addition to joint position, OSCeleton will also calculate and stream joint orientation (rotation) data. This is something the OpenNI framework offers, that the Microsoft Kinect SDK does not. The joint orientation data is packed in the *rotation matrix* format. A rotation matrix will describe relative position and orientation of one rigid body object with respect to another (Spong et al., 2006). OSCeleton will not send this data as a matrix, but as an OSC bundle of nine float numbers prepended by the "/orient" OSC address prefix and user number (multiple users can be tracked at a time). This data can be gathered and treated as a list in Max. Position data will be transmitted with the "/joint" prefix. To be able to treat each joint separately, the streamed data is first routed into separate streams of respective joints (see Table 4.3).

Upper body	**Lower body**
head	r_hip
neck	r_knee
torso	r_ankle
r_shoulder	r_foot
r_elbow	l_hip
r_hand	l_knee
l_shoulder	l_ankle
l_elbow	l_foot
l_hand	

Table 4.3: List of input of skeleton joints tracked and transmitted by OSCeleton

There are two possibilities for visual feedback of motion capture input implemented in the prototype. The first is a simple OpenGL based visualization of the skeleton joint positions (Figure 4.7a). Another visualization algorithm for the data is implemented as a 3D avatar, similar to a typical video game character (Figure 4.7b).

It would be important for this kind of prototype, to as closely as possible imitate a real game environment to make the experience of the music and sound-effect prototyping process as close to a finished product as possible. An attempt was made to implement a default 3D model included with Max 6. In the end it would also be possible to load 3D models rendered from various 3D design programs by following certain principles of joint node assignments.



(a) Skeleton visualization *(b) Character visualization*

Figure 4.7: The visual feedback available in the prototype

To acquire a real-like and smooth control of the 3D character, it is important that joint position and orientation data is correctly routed and assigned to animation parameters. As the jit.anim.node object would not accept the rotation matrix format, it was necessary to make a conversion algorithm for this data. This resulted in the Max tool “matrix2quat.maxpat”. The object takes the list distributed in the rotation matrix format as input, and converts this to an Axis-angle format, with the angle expressed in degrees. The code was inspired of an example by Martin Baker.[14] The 3D character was however never successfully finalised, due to the difficulty of converting orientation data from the Kinect input into correct animation data.

A principal purpose for the Music Kinection prototype was to enable the possibility of performing more advanced motion feature extraction. This is important for obtaining a rich foundation for action-sound mapping solutions. In the prototype, the sub patch “feature.extraction” performs various motion feature extraction operations. This feature extraction implementation is a further development of works by Kristian Nymoen (2011). To prevent bursts and ensure smooth input data, there are some initial filtering procedures in the algorithm. For a full list of the motion features extracted by the algorithm see the left column of Table 4.4. The feature extraction can be assigned to any joint in the main interface window. It is also possible to open monitoring for the extraction of each joint. Further features to be added would be such as absolute position, spherical AED information and information from two separate points, e.g.

[14] http://www.euclideanspace.com/maths/geometry/rotations/conversions/matrixToQuaternion/

euclidean distance.

Table 4.4: Motion features available for analysis in the Music Kinection prototype

Motion feature
x position
y position
z position
x velocity
y velocity
z velocity
Horizontal velocity
Vertical velocity
Absolute velocity
Absolute acceleration QoM

4.6.3 Usage

By performing a calibration pose (see Figure 4.8), the user will be assigned a *skeleton model*. If applied minor adjustments, the prototype would also be able to work on a Windows based platform with automatic detection of skeleton model. The prototype lets the user try a very simple game that is inspired by the Rallyball mini-game from Kinect Adventures! In this simulation, a ball is sent towards the wall and bounced back to the user, the point being to always try to catch the ball. A random impulse can be applied to the ball by hitting the enter key on the keyboard. Visual feedback of the user is only available from the Skeleton visualisation, due to the Character option not being finalised for this version. A simple GUI lets the user choose the joint that is subject to analyse and the desirable feature. A module responsible for the analysis process will appear. This module is also provided as the tool “feature.extraction.maxpat” (see A.16. The module lets the user transmit the realtime analysis data through OSC. In this version, the messages are statically set to send to localhost on port 12345. This should be made into a dynamical parameter in a future version. The “Send to MIDI” feature is not yet implemented in this version. In addition to this, the module lets the user record the chosen joint analysis feature. The recorded data will appear as tab separated .txt files in the folder where the prototype is situated.

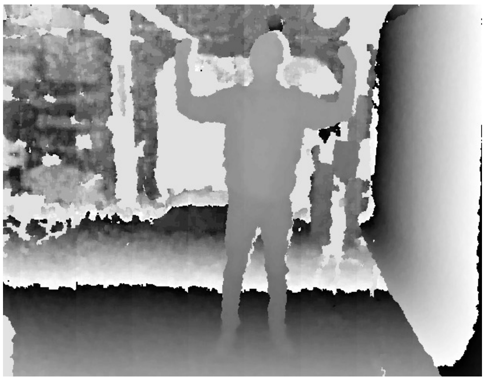
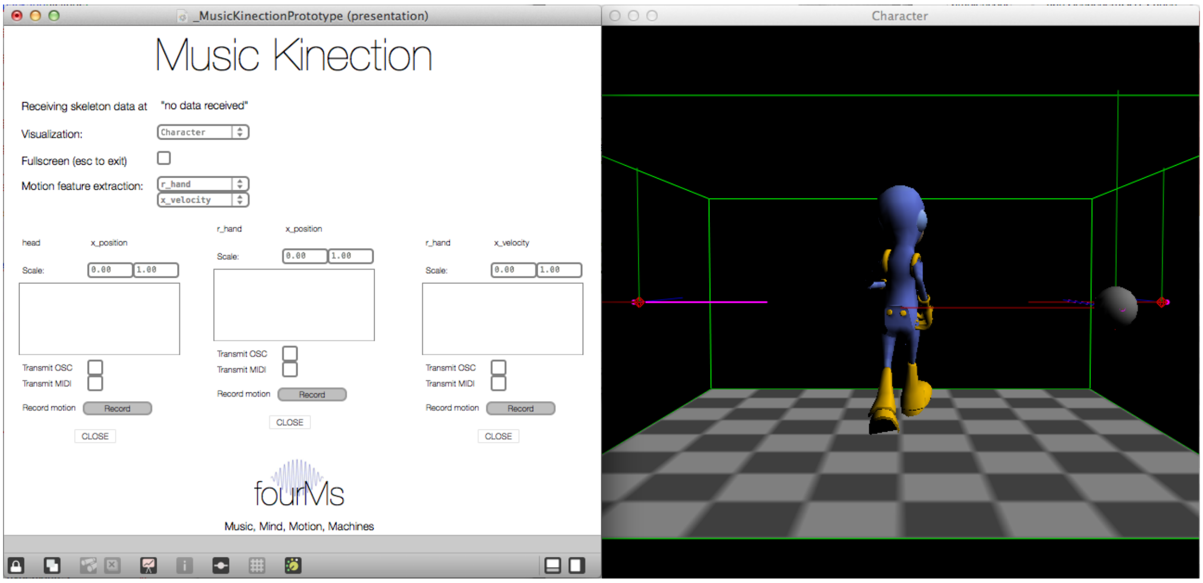


Figure 4.8: OpenNI calibration pose



_MusicKinectionPrototype (presentation)

Music Kinection

Receiving skeleton data at "no data received"

Visualization: Character

Fullscreen (esc to exit) □

Motion feature extraction: r_hand x_velocity

head x_position Scale: 0.00 1.00

Transmit OSC □

Transmit MIDI □

r_hand x_position Scale: 0.00 1.00

Transmit OSC □

Transmit MIDI □

r_hand x_velocity Scale: 0.00 1.00

Transmit OSC □

Transmit MIDI □

Figure 4.9: Music Kinection GUI and visual output

4.6.4 Evaluation

It has been argued that the focus on procedural audio interaction is important in game design (Eladhari et al., 2006; Paul, 2003). This aspect becomes more intricate and complex with regard to gestural interaction. A procedural game audio algorithm for a gestural based interaction system needs to take into consideration a variety of aspects, e.g. environmental awareness and reaction through multi-sensory perception and sound-action couplings.

Jensenius proposes that design strategies for virtual realities could be divided into practical design and creative design (Jensenius, 2007). Practical design is thought to be founded on strong and natural action-sound couplings, in an effort to strengthen the usability of the interaction system. Usability could here refer to e.g. accessibility, convenience, and intuitiveness. Based on creating unsuspected and fun action-sound relationships, creative design seeks to entertain the user. The modeling of the Music Kinection prototype aims to use practical *action-sound design* to enhance the experience of interactively using movement in a system. This can be with regard to continuous interaction with sound, but may also entail the playback of single events.

Careful use of audio in interaction processes can greatly affect gesture parameters and in turn affect performance and overall experience for the user. One of the things I found to be most interesting to explore within this thesis, was the possible exploitation of sound-action couplings in sound design for interactive systems. Possible nodes of exploitations are based on current theories, presented in Chapter 2.

4.7 Discussion

In this chapter, two installations and two software developments were presented as cases of exploration. The Soundshape prototypes were early sketches of ideas, where CV and IrMoCap systems were used to mimic the Kinect sensor. Two prototypes were then developed based on the idea of a game where a user would search for objects, only guided by sonic feedback. The development of these prototypes provided a set of solutions for treating MoCap data and performing motion feature extraction in Max.

The Kinect Piano installation presented a case were an acoustic, and traditionally mechanical, instrument was controlled by motion in the air. The control was facilitated by a Kinect sensor. This presents an interesting case since the mechanics in the piano originally will involve control based on an action-sound coupling, but the user is instead presented with a much weaker action sound relationship through the sensor. It seemed like the users would typically try to use the instrument by two approaches. The first approach involves a traditional piano technique, using two hands in a starting position. The second approach involves using one hand, with more of a "tracing" motion. In the case of how the instrument is designed at this stage, it is of little doubt that the ones who used the instrument with the one-handed "tracing" approach gained the best result. Since the software calculates the centroid of detected motion, the point between two hands would be tracked, and the positions of the hands would not coincide with the key pressed down on the piano. A further development of this installation could possible include the detection of two hands and polyphonic control of the sound, with two or more voices. The development of this installation provided knowledge of how to get raw input from the Kinect in Max and how it is possible to treat this data with the Jitter environment in Max.

The Popsenteret Kinect installation presented a case where full-body motion was tracked for control of visual feedback and control of musical sounds. Due to time-limitations, I decided to only map motion tracked by the arms to control the musical sounds. In a further development of this installation, it would be interesting to let more body parts control e.g. sounds or background music. One example could be that a QoM measurement controls the intensity of some background music presented that starts playing a user initiates the installation. I still have contact with Popsenteret, and would like to develop this further on a later occasion.

The idea for the Music Kinection prototype was to create a software that would imitate a game environment with the facilities to rapidly prototype and experiment with sound design for games implementing full-body motion as input. It was important for the software to include a thorough motion feature extraction procedure, so that various features easily could be mapped to desired external software for sound design work and exploration.

Chapter 5

Conclusion

"Music is the movement of sound to reach the soul for the education of its virtue"

Plato

This chapter will give a summary of the thesis, reflections on the discussions provided by each chapter, and some suggestions to further research and exploration.

5.1 Summary

The presented thesis is inspired by the development of early prototypes of a video game design. My interest for further research was established by the apparent lack of focus on music and motion relationships in game design. In addition to a presentation of the relevance and my motivation for the topic, Chapter 1 presented a main research question and three sub-questions. The sub-questions directly linked to the matters dealt with in Chapters 2, 3, and 4.

For Chapter 2 I wanted to account for relationships between sound and motion, and the cognitive foundations for these relationships. The result was a theoretical background for sound and motion relationships and a framework for the research performed in the project. The framework presented various possible sound and motion relationships presented by theory on sound and motor perception. Connections are particularly evident through concepts of ecological knowledge, mental imagery, and multimodal processing. Links can also be found on a biological level, through entrainment.

In Chapter 3 I asked what similarities and differences that was possible to detect between users of an interactive system. The chapter presented a case study performed by recording motion capture data of subjects playing a variety of commercially available games for the Xbox

Kinect platform. The analysis of the data used both quantitative and qualitative approaches, using the MoCapToolbox to extract information about the amount of motion (QoM) performed by the subjects, as well as plotting the trajectories of the executed actions.

Chapter 4 described four cases of explorations performed on music and motion in interactive systems. Two of these were in the form of an installation setting, while two other more directly addressed development possibilities for systems inspired by commercial products. The installations resulted in useful experiences and observations of both programming solutions and user treatment.

5.2 Reflections

Since a discussion is provided in the end of each chapter, this section will only supplement with concluding remarks. The explorations of interactive properties of video game audio design has arguably become stagnant in the domain of gestural control. One can boldly state that, with the exception of certain attempts at what is known as procedural audio (Eladhari et al., 2006; Farnell, 2007; Paul, 2003, 2008), current video game audio design as a whole lacks inspiration towards developing more radical solutions, e.g. towards a sound-action coupling approach for continuous sonic feedback. Seemingly, more effort is put towards composing movie scores to games, rather than exploring creative modeling of new technological possibilities.

A possible exception to these harsh accusations comes from the games Rez and Child of Eden, created by Tetsuya Mizuguchi of Q Entertainment.[1] Both were presented by the creators as experiments inspired by, *synaesthesia*[2] combining sensations from visual, audible, and tactile stimuli. Rez was designed for the Sega Dreamcast system, and thus still involving a handheld controller, but Child of Eden is designed for the Kinect peripheral. In both games, you add to the musical setting by shooting carefully timed shots at enemies. In Child of Eden, arm movements are used to aim and trigger shots. Enemies will emit a certain melodic pattern upon destruction. I was not able to find out whether sound design was supposed to facilitate movement, but Child of Eden would certainly be an interesting object for further comparative studies.

A solution for future video game audio should be founded on procedural audio playback in video games, based on principles of embodied cognition. This thesis has focused on the

[1] http://www.qentertainment.com

[2] A neurological condition where the stimuli of one modality can cause involuntarily experiences in a second modality.

term musical sound, as opposed to music. Complex musical structures, e.g. keys, phrases and forms, are not considered. However, musical sound can still refer to important musical features, e.g. pitch, timbre and texture. The Music Kinection prototype aims to be of help to explore what I choose to call *embodied sound design*, sound design for systems where an extensive part of the interaction demands human motion as input. It is also an aspiration that the prototype itself will be able to function as a sound and music playback algorithm, which algorithms can be ported to, and implemented in the code of e.g. a game, an installation, or other systems. Embodied sound design should draw upon action-sound models (Section 2.5), such as:

- mental tracing of pitch, texture, and dynamics
- knowledge about environment and objects
- entrainment to pulses and rhythmical figures
- spatialization and depth sensation

As mentioned earlier, one of my first hypotheses was that it could be possible to exploit the multimodal integration process by e.g. virtually anticipate certain audio cues to attract attention or prepare a specific action. This was disproved by the theory presented in Section 2.3.

5.3 Further work

Related to this project, several potential improvements can be regarded. An improvement could for example include a more thorough analysis of the user study data. By running more analyses by e.g. *music information retrieval* (MIR) techniques, musical features in the presented audio could be directly put up against simultaneous subject movement features. Furthermore, these analyses could be compared intersubjectively. Improvements would also include further exploration of mapping or synthesis solutions for the installations as well as prototypes. For the Music Kinection prototype, further development of both the simulated game environment and motion feature extraction algorithm would be beneficial.

There are also many possible aspects available for research relevant to the topics investigated in this project. From a purely technical perspective, I think it would be very helpful to perform a comparison study of the precision of the motion capture performed by the Kinect sensor, opposed to high-end motion capture systems like Optitrack and Qualisys. This could tell us something about how useful and relevant the Kinect sensor can be as a low-cost appliance in academic research.

A very interesting focus for further research on this topic would be concerning health issues and rehabilitation. Using game technology is a current topic in the health industry. As recent as within a month before the writing of this thesis, Oslo Medtech established a focus group for game, simulation, and health. “Examples of areas of use can be cognitive training, physical training activity, social activity – or Simulation and training in the acute situation, ‘hospital world’ and for Operating Theaters” (taken from the website of Oslo Medtech.[3])

The technology presented in this thesis is still young, and it is of little doubt that the evolution of motion tracking devices encourages an exciting future for HCI. It is of my aspiration that the role of sonic feedback and presentation of musical sounds are kept in mind through this evolution, especially especially considering human motion as input. I hope this thesis has helped to shed light on the topic.

[3] http://www.oslomedtech.no/News/NewsArchive/tabid/133/articleType/ArticleView/articleId/216/language/en-US/Oslo-Medtech-launch-game-simulation-and-health aspx

Bibliography

Amelynck, D., M. Grachten, L. van Noorden, and M. Leman (2011). Towards e-motion based music retrieval - a study of affective gesture recognition. *IEEE Transactions on Affective Computing* (99). 5

Bastian, H. C. (1880). *The brain as an organ of mind*. New York: D. Appleton and Company. 13

Braun, H. and I. C. for the History of Technology (2002). *Music and Technology in the Twentieth Century*. Music and Technology in the Twentieth Century. Johns Hopkins University Press. 4

Bregman, A. S. (1990). *Auditory Scene Analysis*, Volume 27. Cambridge, MA: MIT Press. 10

Clarke, E. and N. Cook (2004). *Empirical Musicology: Aims, Methods, Prospects*. Oxford scholarship online. New York: Oxford University Press. 4

Clarke, E. F. (1999). *Rhythm and Timing in Music*, pp. 473–500. Waltham, MA: Academic Press. 19

Clarke, E. F. (2005). *Ways of Listening: An Ecological Approach to the Perception of Musical Meaning*. Oxford University Press. 10

Clayton, M., R. Sager, and U. Will (2004). In time with the music: The concept of entrainment and its significance for ethnomusicology. *Time 1*(11), 1–45. 18

Dix, A. (1998). *Human-computer interaction*. Pearson education. Prentice Hall Europe. 5

Dixon, N. F. and L. Spitz (1980). The detection of auditory visual desynchrony. *Perception 9*(6), 719–721. 13

Dorin, A. (2001). Generative processes and the electronic arts. *Organised Sound 6*(01), 47–53. 4

Eladhari, M., R. Nieuwdorp, and M. Fridenfalk (2006). *The soundtrack of your mind*. New York: ACM Press. 56, 60

Farnell, A. (2007). Synthetic game audio with puredata. *Conference Proceedings, Audio Mostly*. Ilmenau, Germany. 60

Gallese, V. (2003). The roots of empathy: the shared manifold hypothesis and the neural basis of intersubjectivity. *Psychopathology 36*(4), 171–180. 10

Gibson, J. J. (1966). *The senses considered as perceptual systems*. Boston, MA: Houghton Mifflin. 10

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin. 10

Godøy, R. I. (2001). *Imagined Action, Excitation, and Resonance*, pp. 237–250. Lisse: Swets and Zeitlinger. 18

Godøy, R. I. (2003). Motor-mimetic music cognition. *Leonardo 36*(4), 317–319. 11, 14, 38

Godøy, R. I. (2006). Gestural-sonorous objects: embodied extensions of schaeffer's conceptual apparatus. *Organised Sound 11*(02), 149. 15, 18, 38

Godøy, R. I. (2010). Images of sonic objects. *Organised Sound 12*, 54–62. 14

Godøy, R. I. and H. Jørgensen (2001). *Elements of Musical Imagery*. Swets & Zeitlinger, Lisse. 14

Godøy, R. I. and M. Leman (2010). *Musical gestures: sound, movement, and meaning*. New York: Routledge. 4, 10, 13, 16, 18

Hermann, T., A. Hunt, and J. G. Neuhoff (2011). *The Sonification Handbook*. Berlin: Logos Publishing House. 1, 4

Hermann, T. and R. Kõiva (2008). tactilesfor ambient intelligence and interactive sonification. In A. Pirhonen and S. Brewster (Eds.), *Haptic and Audio Interaction Design*, Volume 5270 of *Lecture Notes in Computer Science*, pp. 91–101. Springer Berlin / Heidelberg. 1

Hunt, A. and M. M. Wanderley (2002, August). Mapping performer parameters to synthesis engines. *Org. Sound 7*(2), 97–108. 19

Hunt, A., M. M. Wanderley, and M. Paradis (2003). The importance of parameter mapping in electronic instrument design. *Journal of New Music Research 32*(4), 429–440. 19

Isbister, K. (2011). Emotion and motion: games as inspiration for shaping the future of interface. *interactions 18*, 24–27. 5

Jensenius, A. R. (2007). *Action–Sound : Developing Methods and Tools to Study Music-Related Body Movement*. Ph. D. thesis, University of Oslo. 5, 16, 17, 19, 20, 21, 38, 56

Jensenius, A. R., M. M. Wanderley, R. I. Godøy, and M. Leman (2010). *Musical Gestures: concepts and methods in research*, pp. 12–35. Routledge, New York. 9

King, A. J. (2005). Multisensory integration: Strategies for synchronization. *Current Biology 15*(9), 339–341. 12

Leman, M. (2008). *Embodied music cognition and mediation technology*. Cambridge, Mass.: MIT Press. 4, 5, 10, 18, 19, 21

Leman, M., M. Demey, M. Lesaffre, L. Van Noorden, and D. Moelants (2009). Concepts, technology, and assessment of the social music game "sync-in-team". *2009 International Conference on Computational Science and Engineering*, 837–842. 5

Liberman, A. M. and I. G. Mattingly (1985). The motor theory of speech perception revised. *Perception 21*, 1–36. 11

Marks, R. (2010). Eyetoy, innovation and beyond. *PlayStation Blog*. Accessed July 3, 2012. `http://blog.us.playstation.com/2010/11/03/eyetoy-innovation-and-beyond/`. 42

Merleau-Ponty, M. (1968). *The Visible and the Invisible*. Northwestern University Press, Evanston (IL). 10

Miranda, E. R. and M. M. Wanderley (2006). *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Number 4. A-R Editions, Middleton (WI). 5, 19

Neisser, U. (1976). *Cognition and reality*. Number 1998. W. H. Freeman, New York. 16, 18

Nordahl, R. (2008). Sonic interaction design to enhance presence and motion in virtual environments. In *Proc CHI Workshop on Sonic Interaction Design*, pp. 29–34. Conference on Human Factors in Computing Systems, Florence. 13

Nordahl, R., S. Serafin, and O. Timcenko (2008). Contextualisation and evaluation of novel sonic interfaces using problem based learning. In *Proceedings CHI 2008 Workshop on Sonic Interaction Design*, pp. 17–22. Conference on Human Factors in Computing Systems, Florence. 1

Nymoen, K., S. A. Skogstad, and A. R. Jensenius (2011). Soundsaber - a motion capture instrument. In A. R. Jensenius, A. Tveit, R. I. Godøy, and D. Overholt (Eds.), *Proceedings of the International Conference on New Interfaces for Musical Expression*, Oslo, Norway: University of Oslo and Norwegian Academy of Music, pp. 312–315. 54

Paul, L. J. (2003). Audio prototyping with pure data. *Gamasutra*. Accessed July 3, 2012. http://www.gamasutra.com/view/feature/2849/audio_prototyping_with_pure_data.php. 56, 60

Paul, L. J. (2008). Video game audio prototyping with half-life 2. In R. Adams, S. Gibson, and S. M. Arisona (Eds.), *Transdisciplinary Digital Art. Sound, Vision and the New Screen*, Volume 7 of *Communications in Computer and Information Science*, pp. 187–198. Berlin/Heidelberg: Springer. 60

Rocchesso, D. (2011). *Explorations in sonic interaction design*. Berlin: Logos. 1, 4, 5

Rocchesso, D. and F. Fontana (2003). *The Sounding Object*. Mondo Estremo. 17

Serra, X., M. Leman, and G. Widmer (2007). A roadmap for sound and music computing. *The S2S Consortium*, 1–167. 4

Sherrington, C. (1906). *The Integrative action of the nervous system*. New Haven, CT: Yale University Press. 13

Skogstad, S. A., A. R. Jensenius, and K. Nymoen (2010). Using IR optical marker based motion capture for exploring musical interaction. In K. Beilharz, A. Johnston, S. Ferguson, and A. Y.-C. Chen (Eds.), *NIME 2010 proceedings: New Interfaces for Musical Expression++*, Sydney, Australia: University of Technology, pp. 407–410. 6, 24

Snyder, B. (2000). *Music and Memory: An Introduction*. Cambridge, MA: Mit Press. 14

Spong, M., S. Hutchinson, and M. Vidyasagar (2006). *Robot Modeling and Control*. New York: John Wiley & Sons. 53

Storms, R. L. and M. J. Zyda (2000). Interactions in perceived quality of auditory-visual displays. *Presence Teleoperators Virtual Environments 9*(6), 557–580. 13

Thelle, N. J. W. (2010). Making sensors make sense: challenges in the development of digital musical instruments. MA thesis, University of Oslo. 19

Thomas, N. J. T. (2008). Mental imagery. *The Stanford Encyclopedia of Philosophy*. Accessed July 3, 2012. http://plato.stanford.edu/archives/win2008/entries/mental-imagery/. 14

Toiviainen, P. and B. Burger (2011). *MoCap Toolbox Manual*. Jyväskylä, Finland: University of Jyväskylä. 5, 25

Varela, F. J., E. Thompson, and E. Rosch (1991). *The Embodied Mind: Cognitive Science and Human Experience*, Volume 6. Cambridge, MA: MIT Press. 17

Wallace, M. T. and B. E. Stein (1997). Development of multisensory neurons and multisensory integration in cat superior colliculus. *Journal of Neuroscience 17*(7), 2429–2444. 12

Wohlschläger, A., M. Gattis, and H. Bekkering (2003). Action generation and action perception in imitation: an instance of the ideomotor principle. *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences 358*(1431), 501–15. 38

Wright, M., A. Freed, and A. Momeni (2003). Open sound control: State of the art 2003. *Conference Proceedings, International Conference on New Interfaces for Musical Expression*, 153–159. Montreal, Canada. 44

Appendix **A**

Appendix

A.1 Matlab

A.1.1 readc3d.m

```
function data =readc3d(fname,header)
% This function will read a .C3D file and output the data in a structured
% array
% data = readc3d(fname)
% fname = the c3d file and path (as a string) eg: 'c:\documents\myfile.c3d'
% data is a structured array
%
% see also writec3d.m
%
% CAUTION: machinetype variable may not be correct for intel or MIPS C3D files.
% This m-file needs to be tested with C3D files of these types.
% This m-file was tested and passed with DEC (VAX PDP-11) C3D files
%
% CAUTION: only character, integer, and real numbers have been tested.
% see http://www.c3d.org/HTML/default.htm for information
%
% CAUTION: residuals of 3D data are not handled
%
%
%Created by JJ Loh  2006/09/10
%Departement of Kinesiology
%McGill University, Montreal, Quebec Canada
%
%updated by JJ loh 2008/03/08
%video channels can handle NaN's
%
%updated by JJ Loh 2008/04/10
%header can be outputed alone
%------------------------------------------------

mtype = getmachinecode(fname);
switch mtype
    case 84  %intel
        machinetype = 'ieee-le';
    case 85 %DEC (VAX PDP-11)
        machinetype = 'vaxd';
    case 86 %MIPS
        machinetype = 'ieee-be';
end
```

```
fid=fopen(fname,'r',machinetype);          % if "DEC" selected in export c3d options in IQ you will get an error, change to PC

%------------------------HEADER SECTION------------------------------------------------
% Reading record number of parameter section
pblock=fread(fid,1,'int8');          %getting the 512 block number where the paramter section is located block 1 = first 512 block of the file
fread(fid,1,'int8');           %code for a C3D file

% Getting all the necessary parameters from the header record
%                              word      description
H.ParamterBlockNum = pblock;
H.NumMarkers =fread(fid,1,'int16');              %2       number of markers
H.SamplesPerFrame =fread(fid,1,'int16');         %3       total number of analog measurements per video frame
H.FirstVideoFrame =fread(fid,1,'int16');         %4       # of first video frame
% H.EndVideoFrame =fread(fid,1,'int16');         %5       # of last video frame
H.EndVideoFrame =fread(fid,1,'uint16');          % EB: modified to unsigned (size problem large files)
H.MaxIntGap =fread(fid,1,'int16');               %6       maximum interpolation gap allowed (in frame)
H.Scale =fread(fid,1,'float32');                 %7-8     floating-point scale factor to convert 3D-integers to ref system units
H.StartRecord =fread(fid,1,'int16');             %9       starting record number for 3D point and analog data
H.SamplesPerChannel =fread(fid,1,'int16');       %10      number of analog samples per channel
H.VideoHZ =fread(fid,1,'float32');               %11-12   frequency of video data
fseek(fid,2*148,'bof');                          %13-147  reserved for future use
H.LablePointer =fread(fid,1,'int16');            %label and range data pointer

if nargin == 2
    data = H;
    return
end

%------------------------PARAMETER SECTION---------------------------------------------
fseek(fid,(pblock-1)*512,'bof');  %the start of the parameter block

%parameter header
fseek(fid,2,'cof');  %ignore the first two bytes of the header
numpblocks = fread(fid,1,'uint8'); %number of parameter blocks
processor = fread(fid,1,'uint8'); %processor type 84 = intel, 85 = DEC (VAX PDP-11), 86 = MIPS processor (SGI/MIPS)
switch processor
    case 84 %intel
        machinetype = 'ieee-le';
    case 85 %DEC (VAX PDP-11)
        machinetype = 'vaxd';
    case 86 %MIPS
        machinetype = 'ieee-be';
end
Pheader.NumberOfBlocks = numpblocks;
Pheader.MachineType = processor;
%getting group list
P = struct;
while 1
    numchar = fread(fid,1,'int8');                   %number of characters in the group name
    id = fread(fid,1,'int8');                        %group/parameter id
    gname = char(fread(fid,abs(numchar),'uint8')'); %group/parameter name

    if strcmp(gname,'EndVideoFrame')
        keyboard
    end


    index = ftell(fid);                              %this is the starting point for the offset
    nextgroup = fread(fid,1,'int16');                %nextgroup = offset to the next group/parameter
    if numchar < 0;                                  %a negative character length means the group is locked
        islock = 1;
    else
        islock = 0;
    end
    fld = [];                                    %fld = structured field to add to the output
    fld.id = id;                                 %fld has fields id and description
    fld.islock = islock;

    if id < 0                                         %groups always have id <0 parameters are always >0
        dnum = fread(fid,1,'uint8');                  %number of characters of the desctription
        desc = char(fread(fid,dnum,'uint8')');        %description of the group/parameter
        fld.description = desc;
        P.(gname)=fld;                                %add the field to the variable P
    else %it is a parameter
        dtype = fread(fid,1,'int8');                  %what type of data -1 = char 1 = byte  2 = 16 bit integer 3 = 32 bit floating point
        numdim = fread(fid,1,'uint8');                %number of dimensions (0 to 7 dimensions)
        fld.datatype = dtype;                         %data type of the parameter -1=character, 1=byte, 2=integer, 3= floting point, 4=real
        fld.numberDIM = numdim;                       %number of dimensions (0-7) 0 = scalar, 1=vector, 2=2D matrix,3=3D matrix,... etc
        fld.DIMsize = fread(fid,numdim,'uint8');      %size of each dimension eg [2,3]= 2d matrix with 2 rows and 3 columns
        dsize = fld.DIMsize';                         %the fread function only reads row vectors

        if isempty(dsize)                             %if dsize is empty then we read a scalar
            dsize = 1;
        end
        if length(dsize) > 2
            dsize = prod(dsize);                      %fread can only handle up to 2 dimensions
        end                                           %if it is greater than 2 dimensions, then just read all data in a single vector.

        switch dtype
            case -1 %character data
                pdata = char(fread(fid,dsize,'uint8'));
            case 1 %byte data  !!!Not tested
                pdata = fread(fid,dsize,'bit8');
            case 2 %16 bit integer
                if strcmp(gname,'FRAMES')
                    pdata = fread(fid,dsize,'uint16',machinetype); % ES: quick and dirty fix for invalid size problem
                else
                    pdata = fread(fid,dsize,'int16',machinetype);
                end
            case 3 %32 bit floating point
                pdata = fread(fid,dsize,'float32',machinetype);
            case 4 %REAL data
                pdata = fread(fid,dsize,'float32',machinetype);
        end
        dnum = fread(fid,1,'uint8');               %number of characters in the description
        desc = char(fread(fid,dnum,'uint8')');        %description string
        fld.description = desc;
        fld.data = pdata;                              %add data to parameter structured var
        P = setparameter(P,gname,fld);                %add parameter to the appropriate group
    end
    if nextgroup == 0
        break
    end
    fseek(fid,index+nextgroup,'bof');                 %go to next group/parameter.

end
data.Header = H;
data.ParameterHeader = Pheader;
data.Parameter = P;

%-------------------------------3D & Analog DATA SECTION--------------------

%first position
fseek(fid,(data.Parameter.POINT.DATA_START.data-1)*512,'bof');
%Analogue data parameters
if isfield(data.Parameter,'ANALOG')
    numAnalogue = data.Parameter.ANALOG.USED.data;
    Alabels = cellstr(data.Parameter.ANALOG.LABELS.data');
    Ascale = data.Parameter.ANALOG.SCALE.data;
%   Gscale = data.Parameter.ANALOG.GEN_SCALE.data;
    Aoffset = data.Parameter.ANALOG.OFFSET.data;
%   issigned = data.Parameter.ANALOG.FORMAT.data';              %comment 161, 162,163,164,166 if error
%       if strcmp(issigned,'SIGNED');
%           issigned = 1;
%       else
            issigned = 0;
%    end



 else
    numAnalogue = 0;
    Alabels = [];
    Ascale = [];
    Gscale = [];
    Aoffset = [];
end



%Video (3D) data parameters
numVideo = data.Parameter.POINT.USED.data;
if isfield(data.Parameter.POINT,'LABELS');
    Vlabels = cellstr(data.Parameter.POINT.LABELS.data');
else
    Vlabels = {};
end
Vscale = data.Parameter.POINT.SCALE.data;
numFrames = data.Parameter.POINT.FRAMES.data;

inc = 4*numVideo+H.SamplesPerFrame;
%inc is the increment. Increment is the number of elements in a video
%frame and this consist of:
%The number of Video Channels*4 (xdata,ydata,zdata,and residual) + The
%number of Analogue Measurements per frame;
%Note: the number of Analogue Measurements does NOT always equal the number
%of analogue channels.


%Begin to read the numbers
numdatapts = numFrames*inc;
%number of data points to read this is:
%(Number of frames)*(Number of data per frame)


%READING the DATA
if Vscale >= 0   %integer format
    AVdata = fread(fid,numdatapts,'int16',machinetype);
else             %floating point format
    AVdata = fread(fid,numdatapts,'float32',machinetype);
end



V = struct;
%data for all Video channels
offset = 1;
for i = 1:numVideo
    xd = AVdata(offset:inc:end);
    yd = AVdata(offset+1:inc:end);
    zd = AVdata(offset+2:inc:end);
    residual = AVdata(offset+3:inc:end);
    if i > length(Vlabels)
        Vdata.label = ['MRK␣',num2str(i)];
    else
        Vdata.label = Vlabels{i};
    end
    indx = findzeros([makecolumn(xd),makecolumn(yd),makecolumn(zd)]);
    Vdata.xdata = videoconvert(xd,Vscale,indx);
    Vdata.ydata = videoconvert(yd,Vscale,indx);
    Vdata.zdata = videoconvert(zd,Vscale,indx);
    Vdata.residual = residual;
    offset = offset+4;
    V.(['channel',num2str(i)]) = Vdata;
end

offset = 4*numVideo; %offset is a pointer to the first data point of the first channel of Analog data
A = struct;
for i = 1:numAnalogue
    Adata.label = Alabels{i};
    Aframedata = [];
    %A given analog channel can have multiple samples per frame of video
    for j = 0:H.SamplesPerChannel-1
        stindx = offset+i+j*numAnalogue;
        plate = AVdata(stindx:inc:end);
        Aframedata = [Aframedata,plate];
    end
    Adata.data = analogconvert(merge(Aframedata),Aoffset(i),Ascale(i),Gscale,issigned); %recombine the multiple samples to one vector
    A.(['channel',num2str(i)]) = Adata;
end

data.VideoData = V;
data.AnalogData= A;

fclose(fid);



function r = setparameter(g,name,info)
%this function will add a parameter to the appropriate group (based on the
%id) Note if no group is found, the parameter will not be added.
fld = fieldnames(g);
r = g;
for i = 1:length(fld)
    d = getfield(g,fld{i});
    if abs(info.id) == abs(d.id);
        r = setfield(g,validfield(fld{i}),setfield(d,validfield(name),info));
        break
    end
end

function r = merge(data)
%this function will recombine the analogue data because the potential for multiple
%samples per frame of video.
%each row of "data" corresponds to a single video frame;
[rw,cl] = size(data);
r = zeros(rw*cl,1);
for i = 1:cl
    r(i:cl:end) = data(:,i);
end

function r = videoconvert(data,scale,indx)
%convert the video channels to real data values
if scale >0
    r = data*scale;
else
    r = data;
end
r(indx) = NaN;

function r = analogconvert(data,offset,chscale,gscale,issigned)
%convert analog channesl to real data values
if ~issigned
    data = unsign(data);
end
r = (data-offset)*chscale*gscale;
```

```
function r = unsign(data)
indx = find(data<0);
data(indx) = 2^16+data(indx);
r = data;

function r = getmachinecode(fname)
fid = fopen(fname,'r');
pblock=fread(fid,1,'int8')-1;          %getting the 512 block number where the paramter section is located block 1 = first 512 block of the file
fseek(fid,pblock*512+3,'bof');
r = fread(fid,1,'uint8');
fclose(fid);

function r = findzeros(data)
indx = find(data(:,1)==0);
for i = 2:3
    nindx = find(data(:,i)==0);
    indx = intersect(indx,nindx);
end

r = indx;
```

A.1.2 qom_plots.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Music Kinection QoM BoxPlots  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cumdist = struct;
qom = struct;
norm_mrk = struct;


% Read all c3d files in folder
file_path = uigetdir;

cd (file_path);

% Create folder to store plots in
% [s,mess,messid] = mkdir('qom_plots');

c3d_files = dir('*.c3d');
for i=1:length(c3d_files)

        % Import each c3d file
        a = mcread(c3d_files(i).name);
    a = mcgetmarker(a, 1:38);

    % Get markernames for x axis parameter
    % xaxisparams = a(i).markerName

        % Cut away 10 first frames, in case of NaN.
        dl = length(a.data);
    a = mctrim(a, 10, dl, 'frame');
        % Match nframes with data. Problem with readc3d long file import.
    a.nFrames = length(a.data);
    % COnvert to meters

        % Get time
        a.time = a.nFrames/a.freq;
        % Get markernames
        mn = mcgetmarkername(a);

        % Store file name information to know where to output plots
        [file_path,file_name,file_extension] = fileparts(a.filename);
        % Normalizes data
        a_mean = mcmean(a.data);
        a_normalize = a; % Copy data to get metadata
        a_normalize.data = bsxfun(@minus,a.data,a_mean);

        % Calculate norms
        a_norm = mcnorm(a_normalize);


        % Calulate cumulative distance of all markers

    % Find cummulative distance
        a_dist = mccumdist(a_norm);

    % Convert to meters
    a_dist.data = (a_dist.data)/1000;

        % Find last entry which is the total cummulative distance.
        distm = a_dist.data(a_dist.nFrames,:);
    cumdist.(file_name) = a_dist;
        qom.(file_name) = distm/a.time;
%      norm_mrk.(file_name) = mcnorm(cumdist.(file_name));

    qom_c = struct2cell(qom); % Structure to cell conversion
    qom_m = cell2mat(qom_c); % Cell to matrix conversion



end

% QoM Boxplot
boxplot(qom_m);
figure(gcf)

% Change X-axis parameters


        set (gca,'XTick',1:38);
        set (gca,'XTickLabel',mn);
        % rotation = 45;

% Create xlabel
h = xlabel('Marker');
% Position the label
pos = get(h,'pos'); % Read position [x y z]
set(h,'pos',pos+[0 0.5 0]) % Move label to right

% Create ylabel
ylabel('QOM');

% Create title
title('Overview␣of␣QOM␣in␣test␣01␣(Rallyball)');
```

A.1.3 mgtplotall_c3d.m

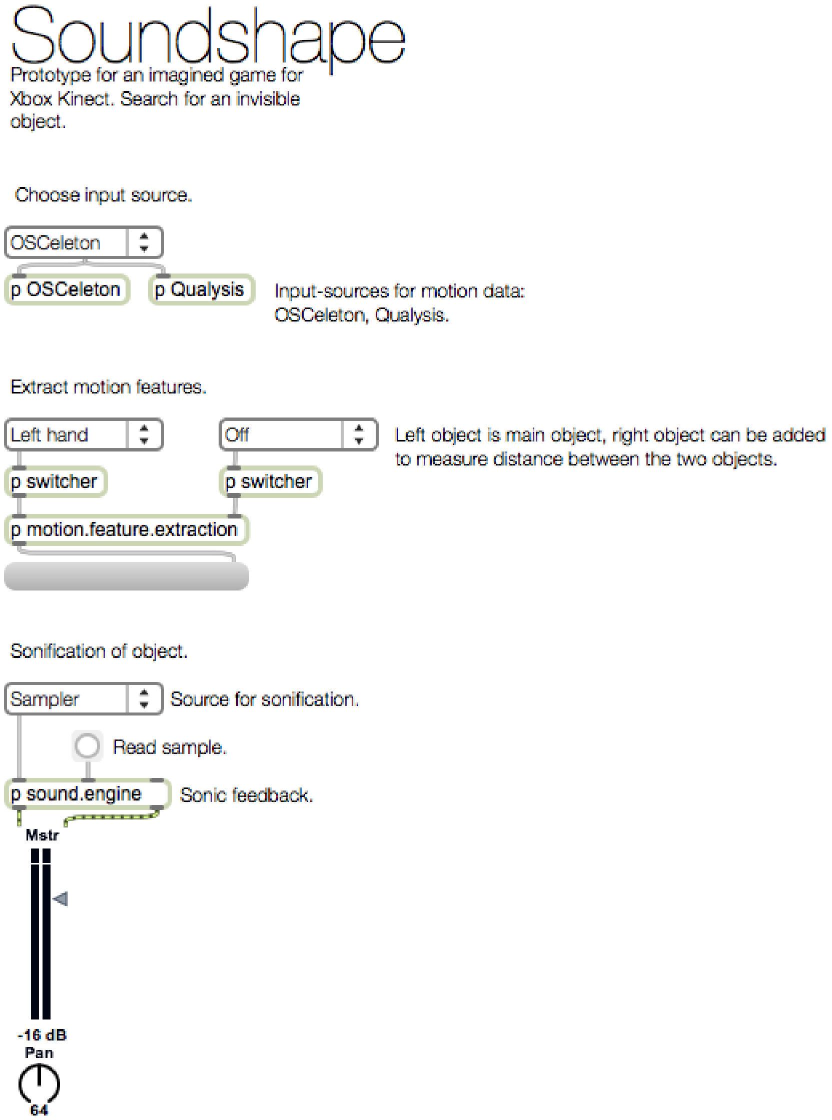
```
function d = mgtplotall(fn)
        % Reads a folder with motion capture files (c3d) and outputs a series of plots for each file
        %
        %
        % (c) Part of the Musical Gestures Toolbox, Copyright (c)2012,
        % University of Oslo

        % To avoid problems with subscript in titles
        set(0, 'DefaulttextInterpreter', 'none')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global variables
xyz = ['X' 'Y' 'Z'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Read all c3d files in folder
file_path = uigetdir;

cd (file_path);

% Create folder to store plots in with the current date and time
d = datestr(now,'yyyy-mm-dd-THH-MM-SS');
[s,mess,messid] = mkdir(d);
folder_name = d;

c3d_files = dir('*.c3d');
for i=1: length(c3d_files)

        % Import each c3d file
        a = mcread(c3d_files(i).name);
        a = mcgetmarker(a, 1:38);
        %str = ['Importing file: ' a.filename];
        %disp(str);

        % Store file name information to know where to output plots
        [file_path,file_name,file_extension] = fileparts(a.filename);

        % Cut away 10 first frames, in case of NaN.
        dl = length(a.data);
    a = mctrim(a, 10, dl, 'frame');

        % In case there are any missing markers in the data set, we fill those holes:
        a = mcfillgaps(a,100);

        % To avoid problems with different sampling rates
        %a = mcresample(a,20);

        % Create time series in seconds, to plot in seconds
        time = a.nFrames/a.freq;
        time_data = ((1:a.nFrames)')/a.freq;

        % Smooths data
        %a_smooth = mcsmoothen(a, 'acc');
        %smooth_factor = 10*a.freq; % Smoothing over 10 seconds of data

        % Normalizes data
        a_mean = mcmean(a.data);
        a_normalize = a; % Copy data to get metadata
        a_normalize.data = bsxfun(@minus,a.data,a_mean);
        %a_normalize_smooth = mcsmooth(a_normalize)

        % Calculate first and second derivatives
        a_dx = mctimeder(a, 1, [1 0.9999]); %without any filtering
        a_dx2 = mctimeder(a, 2, [1 0.9999]);

        % Calculate cummulative distance
        a_cumdist=mccumdist(a);

        % Calculate norms
        a_norm = mcnorm(a_normalize);
        a_dx_norm = mcnorm(a_dx);
        a_dx2_norm = mcnorm(a_dx2);

        % Calculate spectrum
        %[s f] = mcspectrum(a);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create combined plots for each markerfigure;

figure;
str_title = ['File:␣' a.filename ];
set(gcf,'Position',[200 200 800 700],'Name',str_title,'NumberTitle','off');

for j = 1:3
        subplot(6,3,(((j-1)*3)+(1:3))) % finds correct subplots to fill
        plot(time_data, a.data(:,[j:3:a.nMarkers*3]));
        grid on;
        str_title = [xyz(j) '␣pos␣(mm)␣'];
        ylabel(str_title);
        %               str_title = [xyz(j) ' position '];
        %               title(str_title);

        if(j==1)
                set(gca,'xticklabel',[]);
                %                                       legend(a.markerName);
                title('Positions␣for␣all␣markers')
        end;

        if(j==2)
                set(gca,'xticklabel',[]);
        end;

        if(j==3)
                xlabel('Time␣(s)');
        end;
end


subplot(6,3,[13 16])
plot(a.data(:,[1:3:a.nMarkers*3]),a.data(:,[2:3:a.nMarkers*3]));
grid on;
title('XY');
xlabel('X␣norm.␣pos.␣(mm)');
ylabel('Y␣norm.␣pos.␣(mm)');
%               legend(a.markerName);
axis equal;

subplot(6,3,[14 17])
plot(a.data(:,[1:3:a.nMarkers*3]),a.data(:,[3:3:a.nMarkers*3]));
grid on;
title('XZ');
xlabel('X␣norm.␣pos.␣(mm)');
ylabel('Z␣norm.␣pos.␣(mm)');
%               legend(a.markerName);
axis equal;

subplot(6,3,[15 18])
plot(a.data(:,[2:3:a.nMarkers*3]),a.data(:,[3:3:a.nMarkers*3]));
grid on;
title('YZ');
xlabel('Y␣norm.␣pos.␣(mm)');
ylabel('Z␣norm.␣pos.␣(mm)');
axis equal;

% Adding one legend for all plots
h = legend(a.markerName,'Orientation','horizontal');
pos = get(h,'position');
set(h, 'position',[0.13 0.4 pos(3:4)]);

% Create title
```

```
    axes('Position',[0 0 1 1],'Visible','off');
    str_title = ['www.fourMs.uio.no␣--␣File:␣' a.filename '␣--␣All␣markers␣--␣Frequency:␣' num2str(a.freq) 'Hz'];
    text(0.02,0.98,str_title);

    % export figures to EPS files
    set(gcf, 'PaperPositionMode', 'auto')   % Use screen size for exported file
    filename = [folder_name '/' file_name '_XYZ']; % Create filename
    print('-depsc', '-tiff␣', filename);
    close;


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % BOXPLOTS

    figure;
    boxplot(a_norm.data,a.markerName,'labelorientation','inline');
    %boxplot(a_normalize(:,(1:7))) % Looking only at some markers
    ylabel('mm');

    % Create title
    axes('Position',[0 0 1 1],'Visible','off');
    str_title = ['Boxplot␣of␣normalized␣position␣magnitude␣--␣File:␣' a.filename '--␣Frequency:␣' num2str(a.freq) 'Hz'];
    text(0.02,0.98,str_title);

    % export figures to EPS files
    set(gcf, 'PaperPositionMode', 'auto')   % Use screen size for exported file
    filename = [folder_name '/' file_name '_boxplot'];        % Create filename
    print('-depsc', '-tiff␣', filename);
    close;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % HISTOGRAMS

    figure;
    for j = 1:a.nMarkers
        subplot(3,ceil(a.nMarkers/3),j)
        hist(a_normalize.data(:,j));
        h = findobj(gca,'Type','patch');
        set(h,'FaceColor',[0.7 0.7 0.7],'EdgeColor','w') % set different colour
        title([a.markerName(j)]);
    end

    % Create title
    axes('Position',[0 0 1 1],'Visible','off');
    str_title = ['Histogram␣of␣normalized␣position␣magnitude␣--␣File:␣' a.filename  '--␣Frequency:␣' num2str(a.freq) 'Hz'];
    text(0.02,0.98,str_title);

    % export figures to EPS files
    set(gcf, 'PaperPositionMode', 'auto')   % Use screen size for exported file
    filename = [folder_name '/' file_name '_histo'];          % Create filename
    print('-depsc', '-tiff␣', filename);
    close;

end
```

A.2 Max patches



Soundshape

Prototype for an imagined game for Xbox Kinect. Search for an invisible object.

Choose input source.

OSCeleton

Input-sources for motion data: OSCeleton, Qualysis.

Extract motion features.

Left hand

Off

Left object is main object, right object can be added to measure distance between the two objects.

Sonification of object.

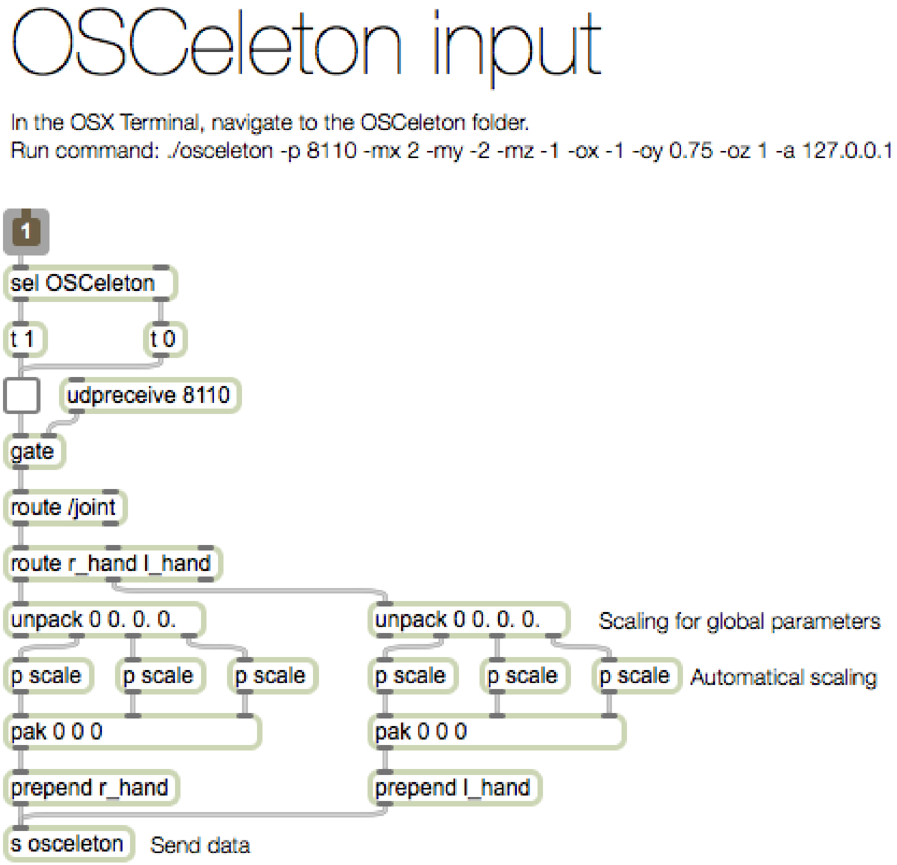
Sampler

Source for sonification.

Read sample.

Sonic feedback.

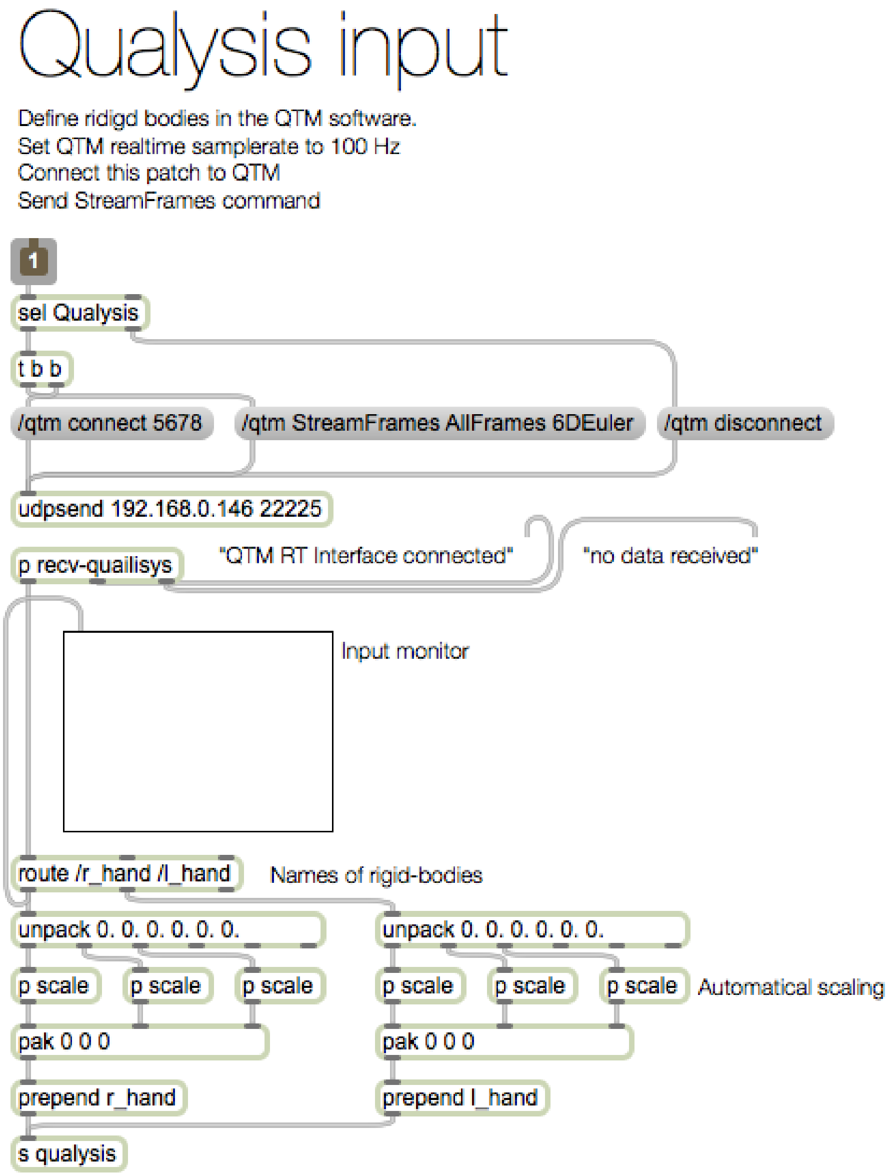
Figure A.1: Soundshape main window



OSCeleton input

In the OSX Terminal, navigate to the OSCeleton folder.
Run command: ./osceleton -p 8110 -mx 2 -my -2 -mz -1 -ox -1 -oy 0.75 -oz 1 -a 127.0.0.1

Figure A.2: Soundshape OSCeleton input



Qualysis input

Define ridigd bodies in the QTM software.
Set QTM realtime samplerate to 100 Hz
Connect this patch to QTM
Send StreamFrames command

Figure A.3: Soundshape qualysis input

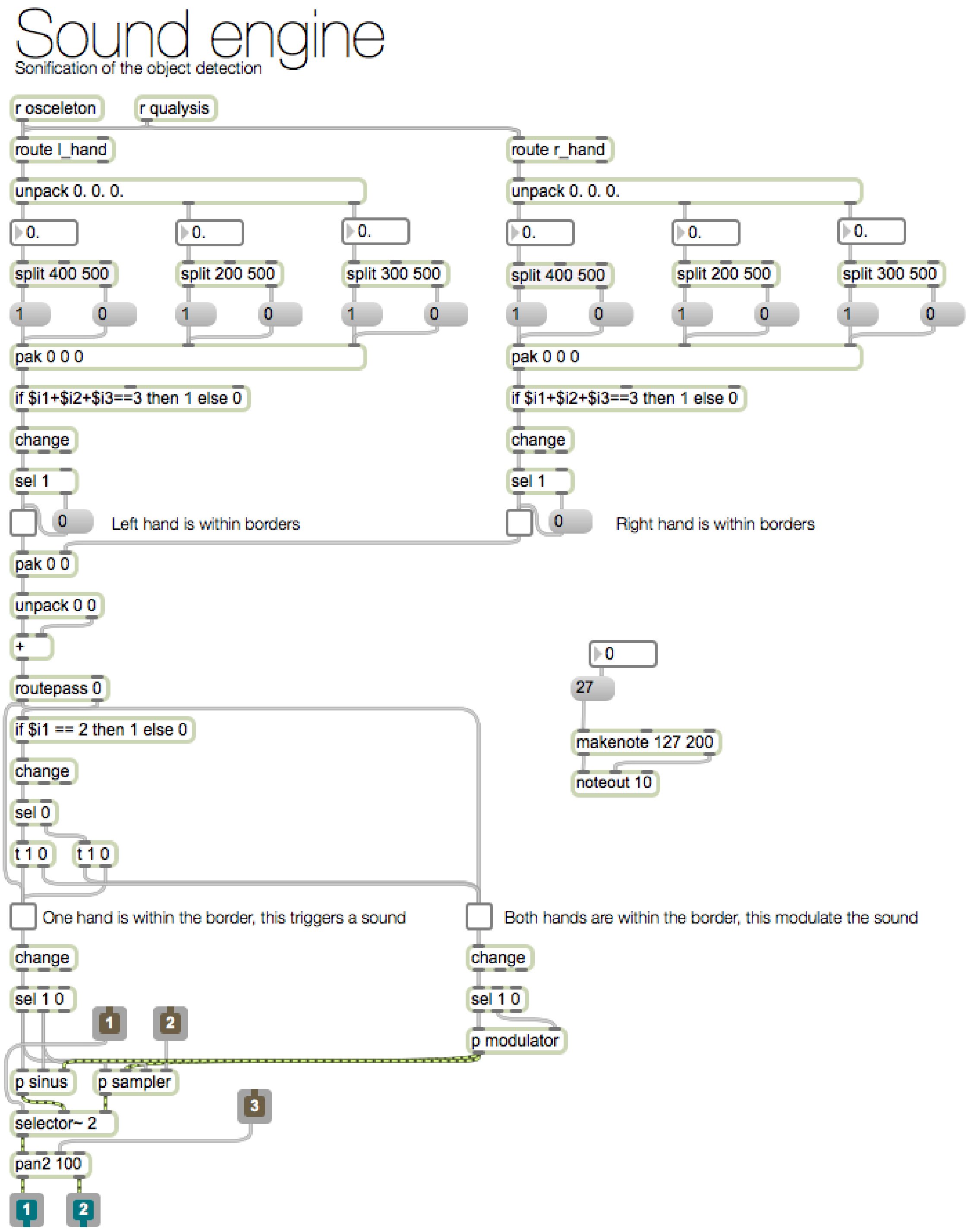


Figure A.4: Soundshape sound engine

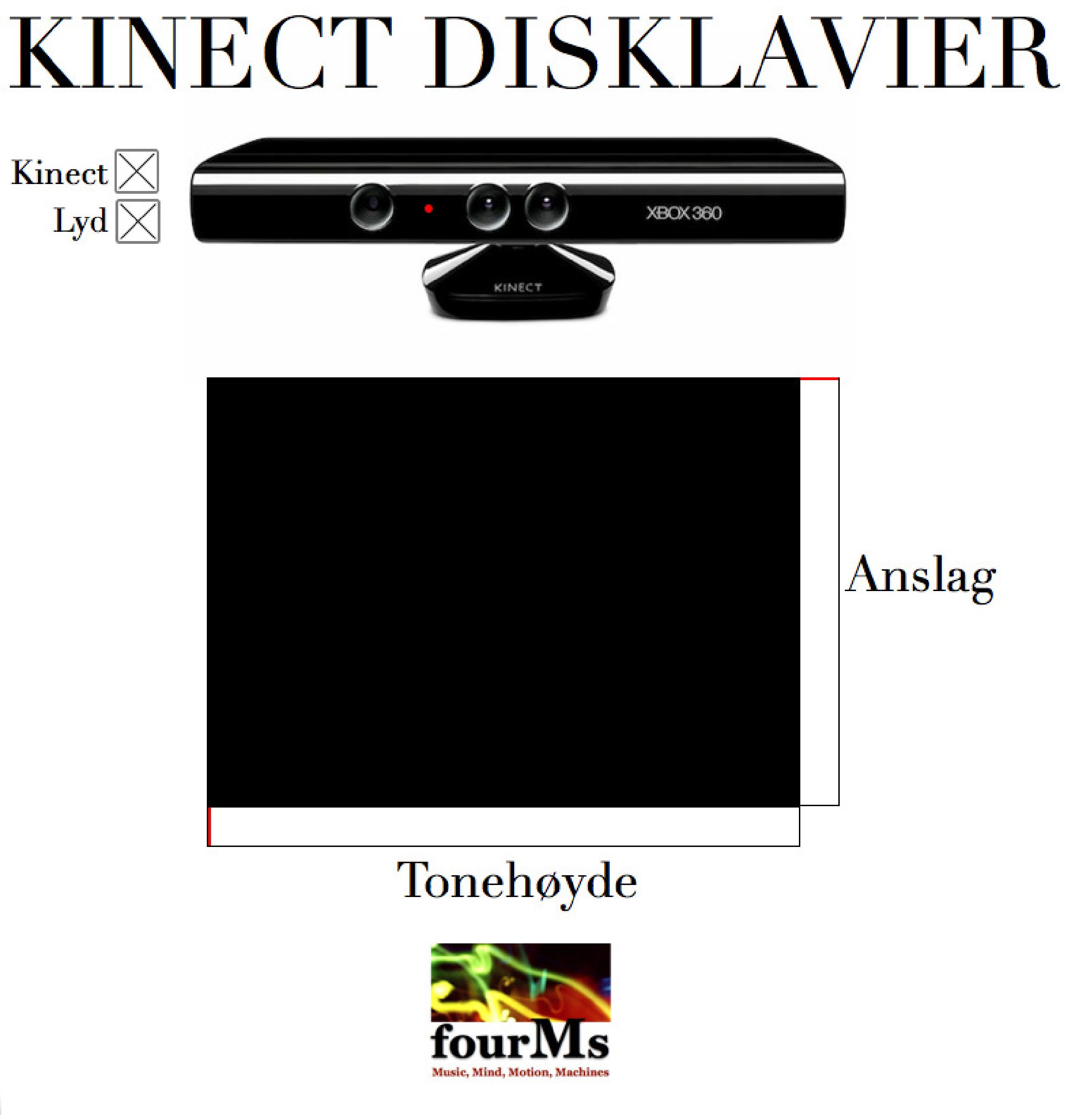


Figure A.5: Kinect Piano GUI

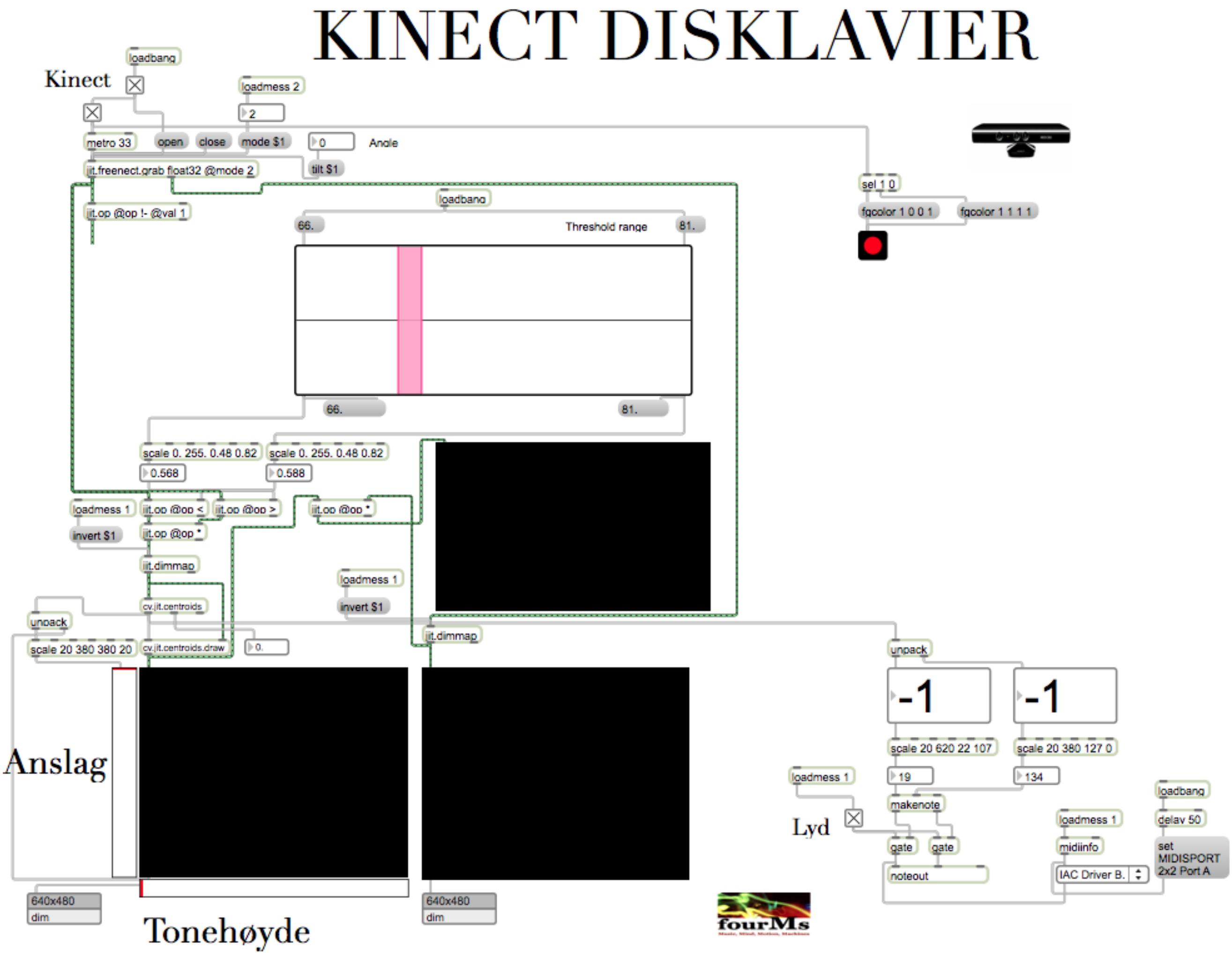
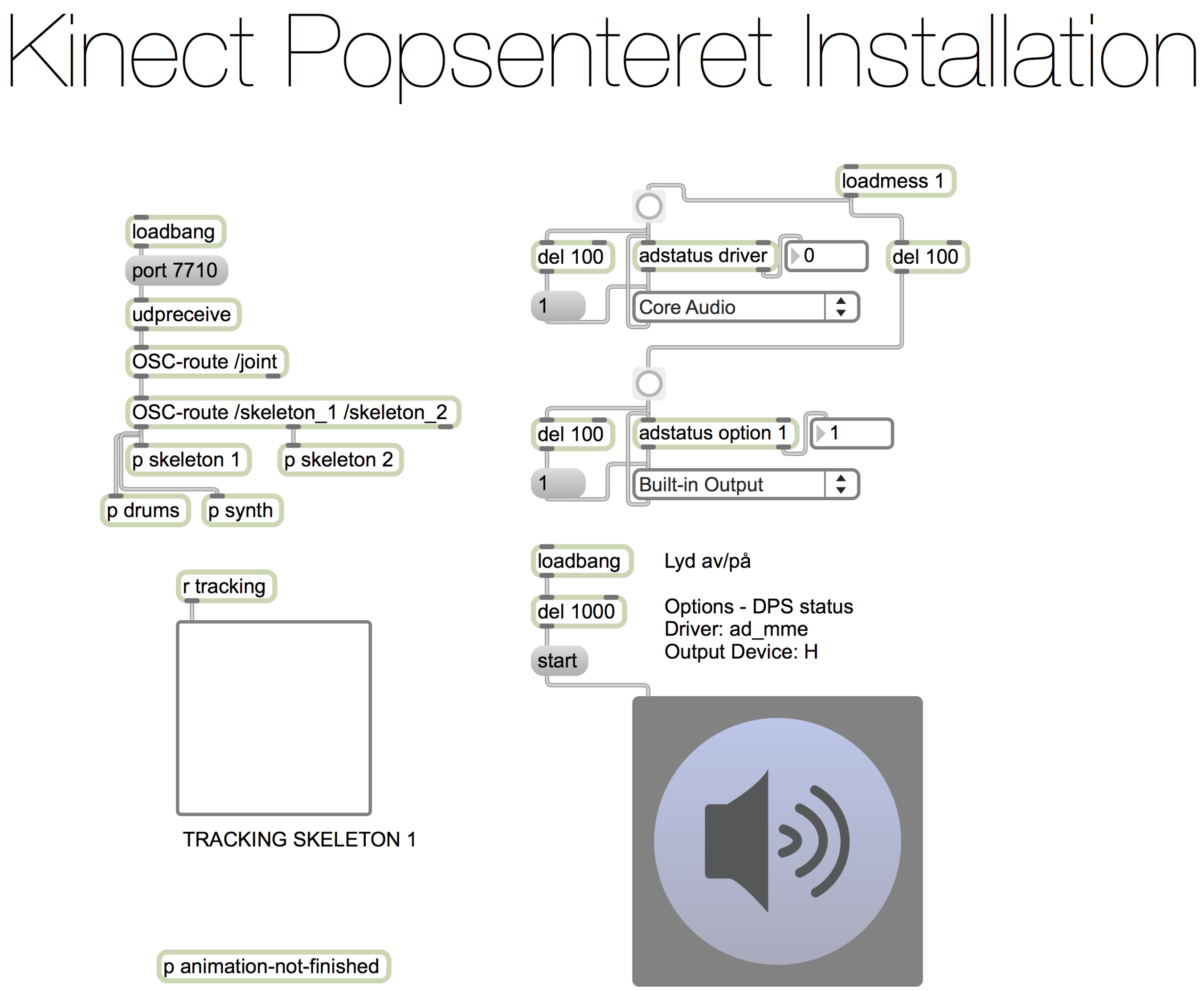


Figure A.6: Kinect Piano code



Kinect Popsenteret Installation

loadbang
port 7710
udpreceive
OSC-route /joint
OSC-route /skeleton_1 /skeleton_2
p skeleton 1
p skeleton 2
p drums
p synth

r tracking

TRACKING SKELETON 1

p animation-not-finished

loadmess 1
del 100
adstatus driver
0
del 100
1
Core Audio

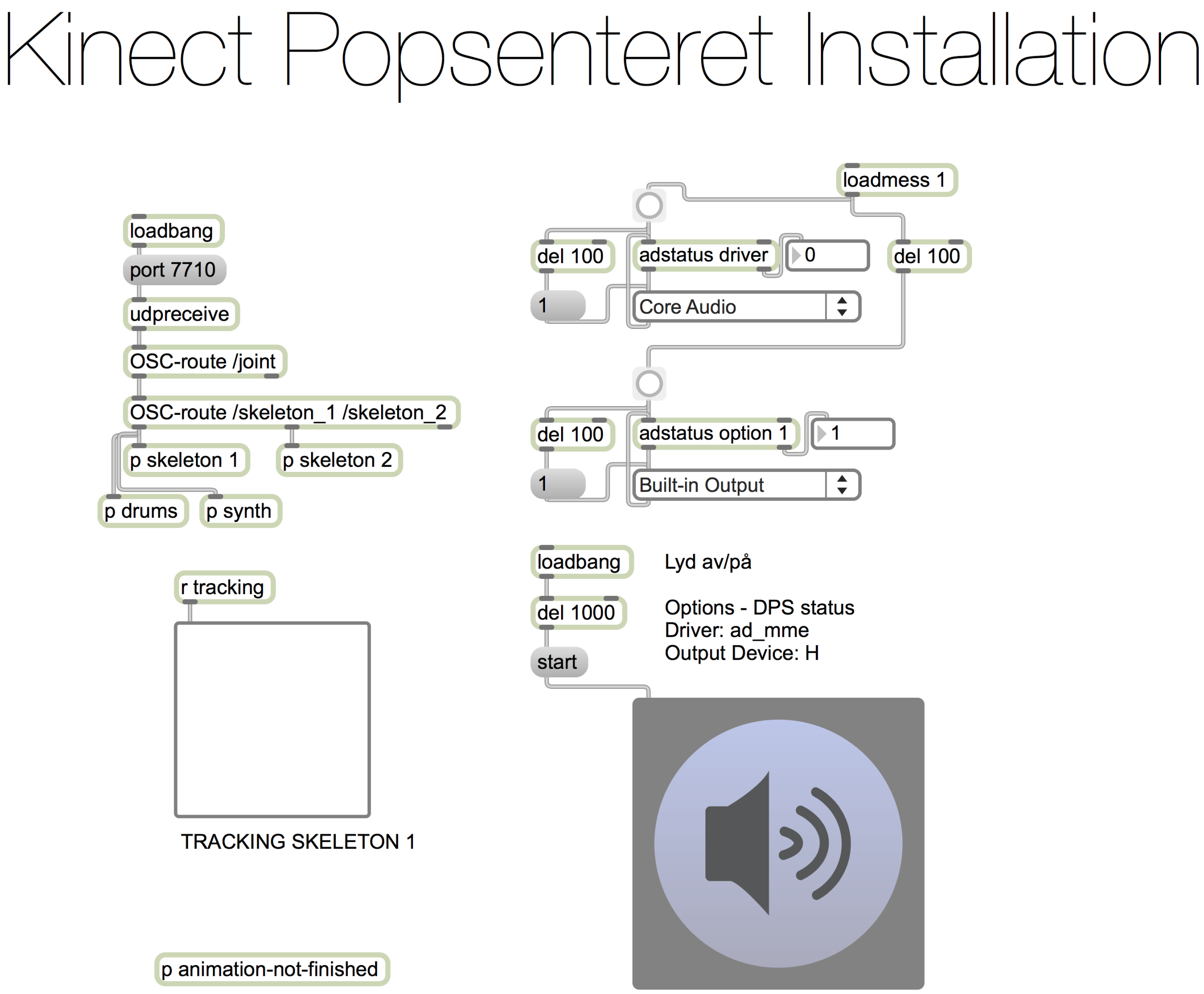
del 100
adstatus option 1
1
1
Built-in Output

loadbang
del 1000
start

Lyd av/på

Options - DPS status
Driver: ad_mme
Output Device: H

Figure A.7: Kinect popsenteret installation code



Kinect Popsenteret Installation

```
loadbang
port 7710
udpreceive
OSC-route /joint
OSC-route /skeleton_1 /skeleton_2
p skeleton 1
p skeleton 2
p drums
p synth

r tracking
```

TRACKING SKELETON 1

```
p animation-not-finished

loadmess 1
del 100
adstatus driver
0
del 100
1
Core Audio

del 100
adstatus option 1
1
1
Built-in Output

loadbang
del 1000
start
```

Lyd av/på

Options - DPS status
Driver: ad_mme
Output Device: H

Figure A.8: Kinect popsenteret installation code

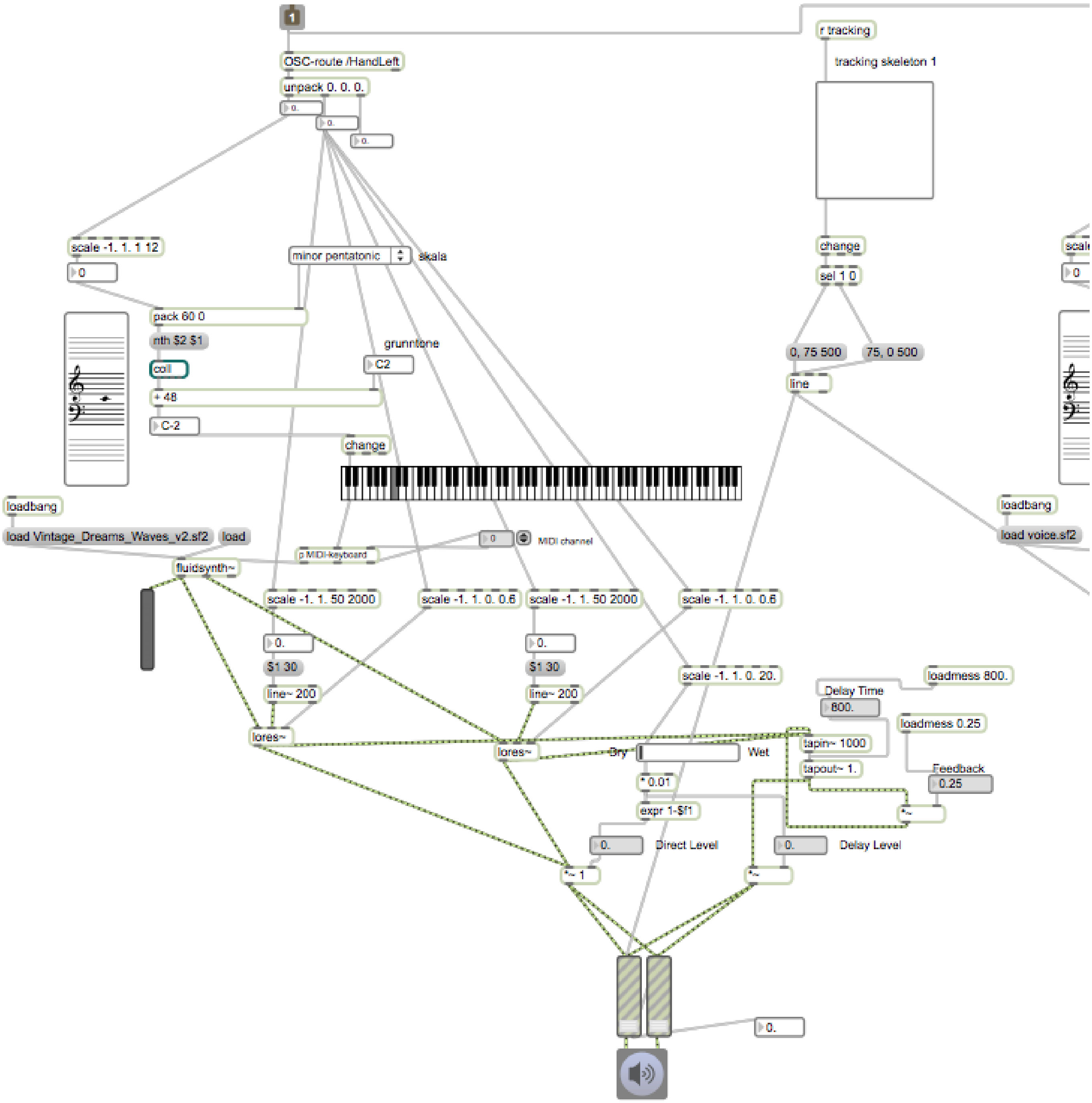
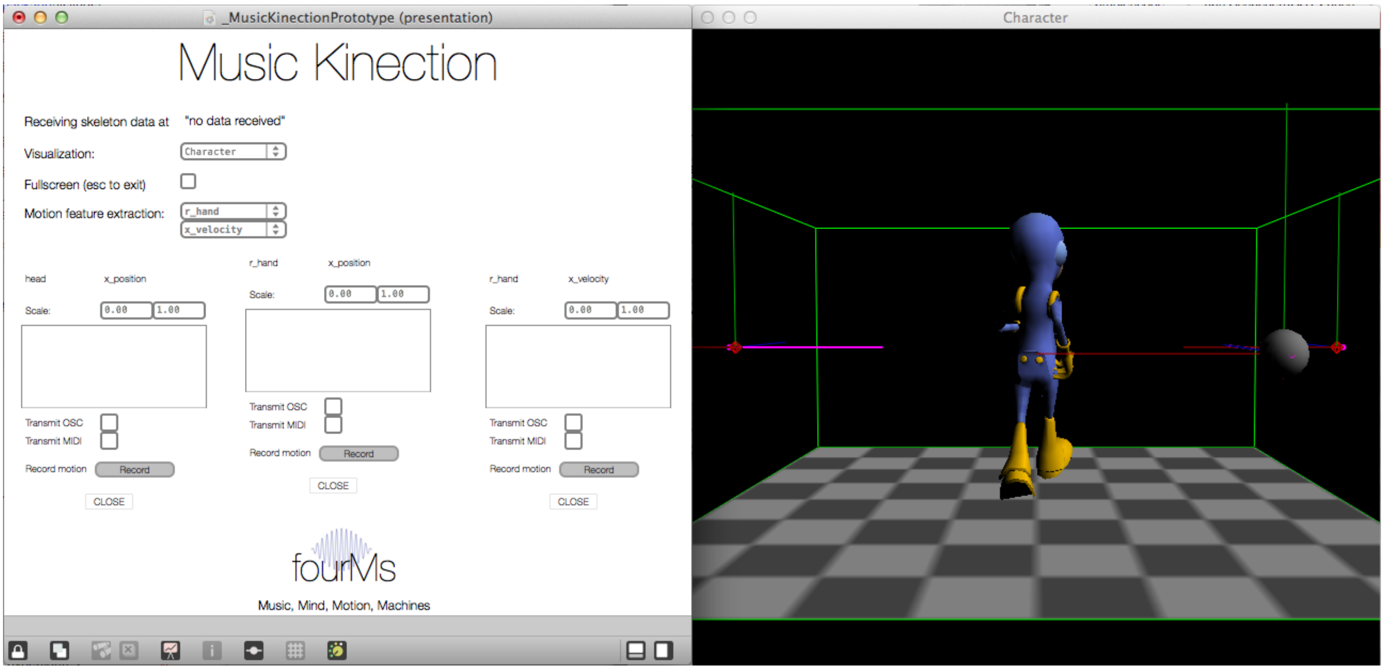


Figure A.9: Kinect popsenteret synth control (extraction)



_MusicKinectionPrototype (presentation)

Music Kinection

Receiving skeleton data at "no data received"

Visualization: Character

Fullscreen (esc to exit) □

Motion feature extraction: r_hand x_velocity

head x_position

Scale: 0.00 1.00

Transmit OSC □

Transmit MIDI □

Record motion

r_hand x_position

Scale: 0.00 1.00

Transmit OSC □

Transmit MIDI □

Record motion

r_hand x_velocity

Scale: 0.00 1.00

Transmit OSC □

Transmit MIDI □

Record motion

Figure A.10: Music Kinection Prototype GUI

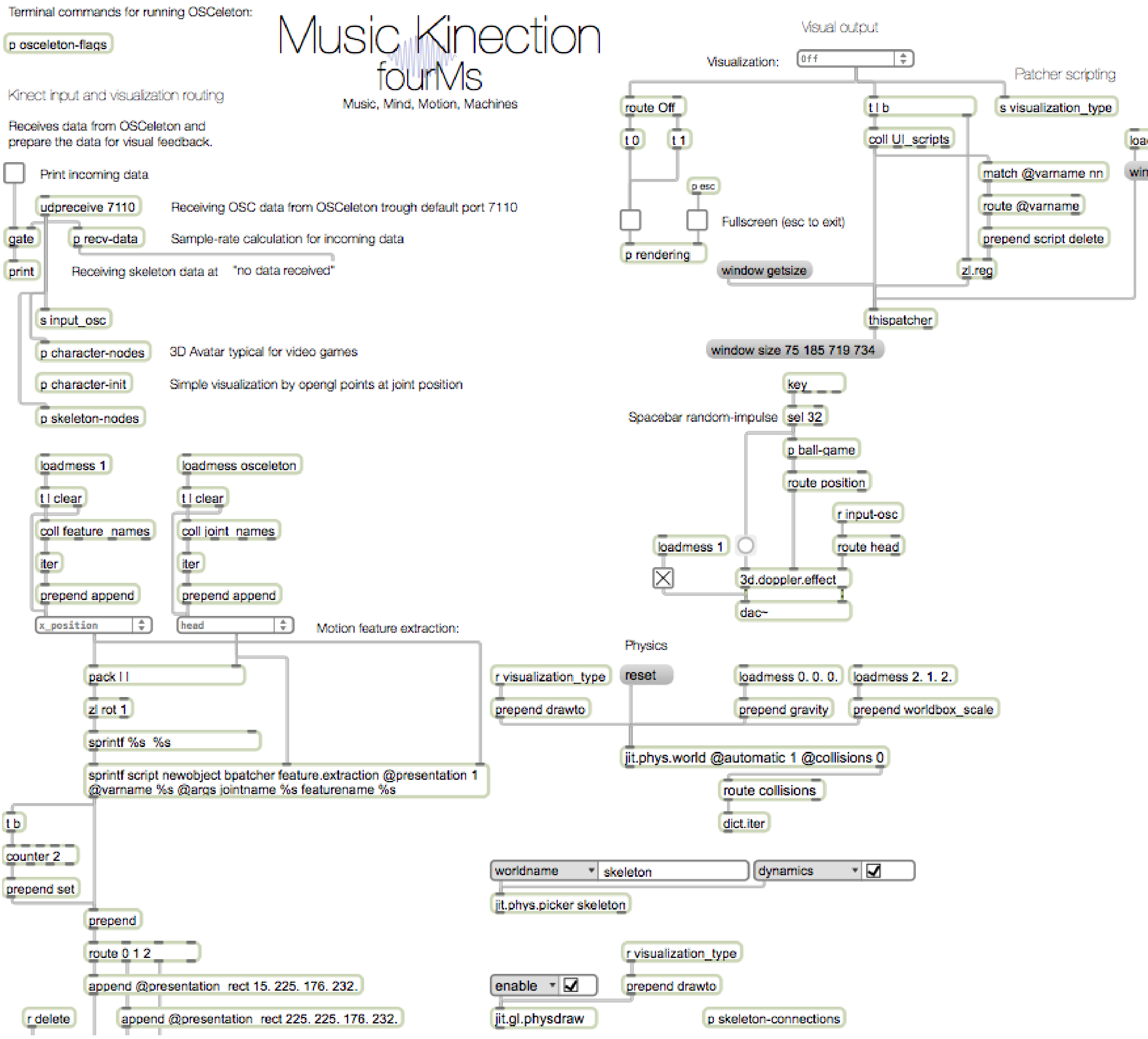
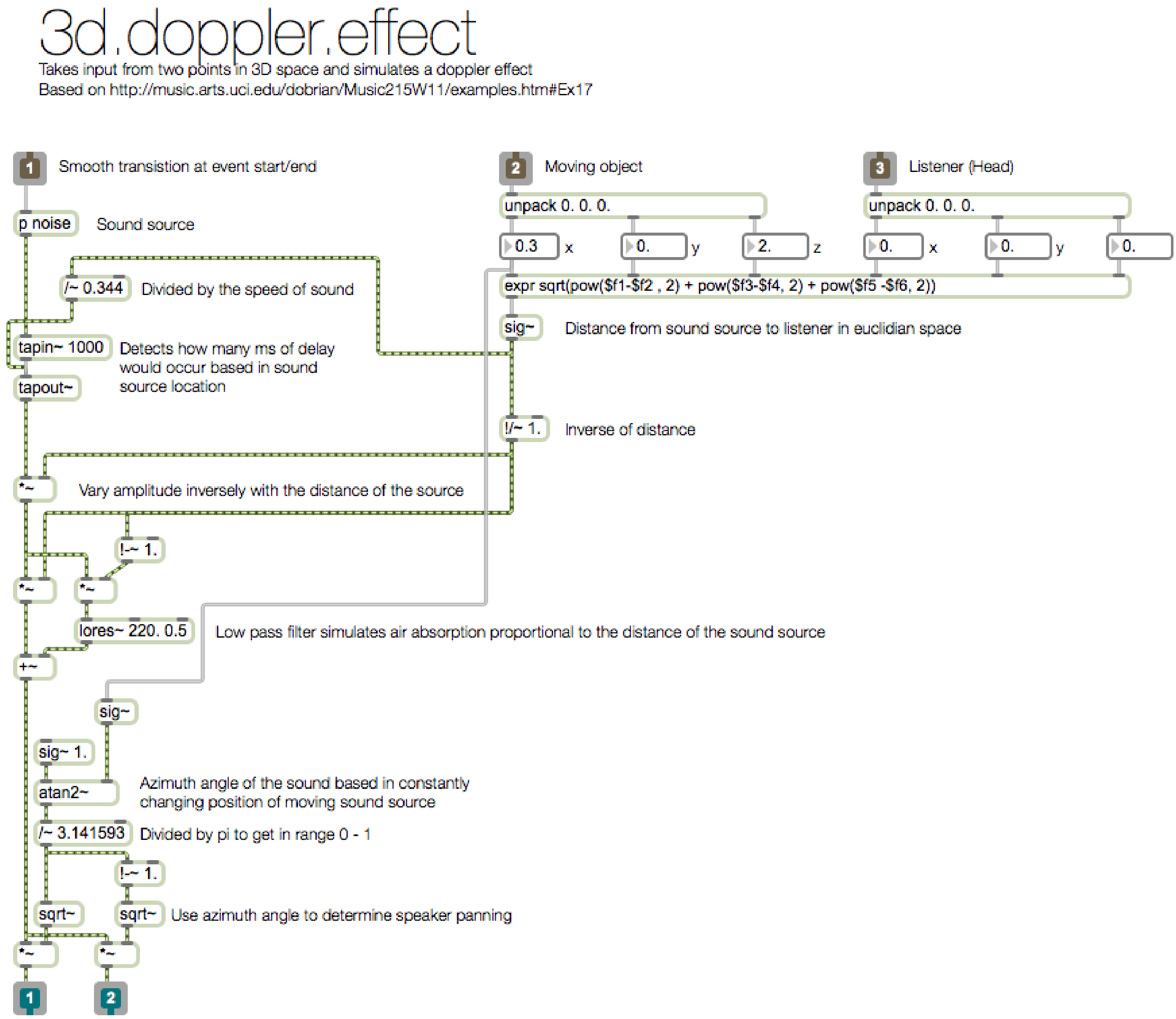


Figure A.11: Music Kinection Prototype code (extraction)



3d.doppler.effect

Takes input from two points in 3D space and simulates a doppler effect

Based on http://music.arts.uci.edu/dobrian/Music215W11/examples.htm#Ex17

Smooth transistion at event start/end

Sound source

Divided by the speed of sound

Detects how many ms of delay would occur based in sound source location

Vary amplitude inversely with the distance of the source

Low pass filter simulates air absorption proportional to the distance of the sound source

Azimuth angle of the sound based in constantly changing position of moving sound source

Divided by pi to get in range 0 - 1

Use azimuth angle to determine speaker panning

Moving object

Listener (Head)

Distance from sound source to listener in euclidian space

Inverse of distance

Figure A.12: 3D.doppler.effect code

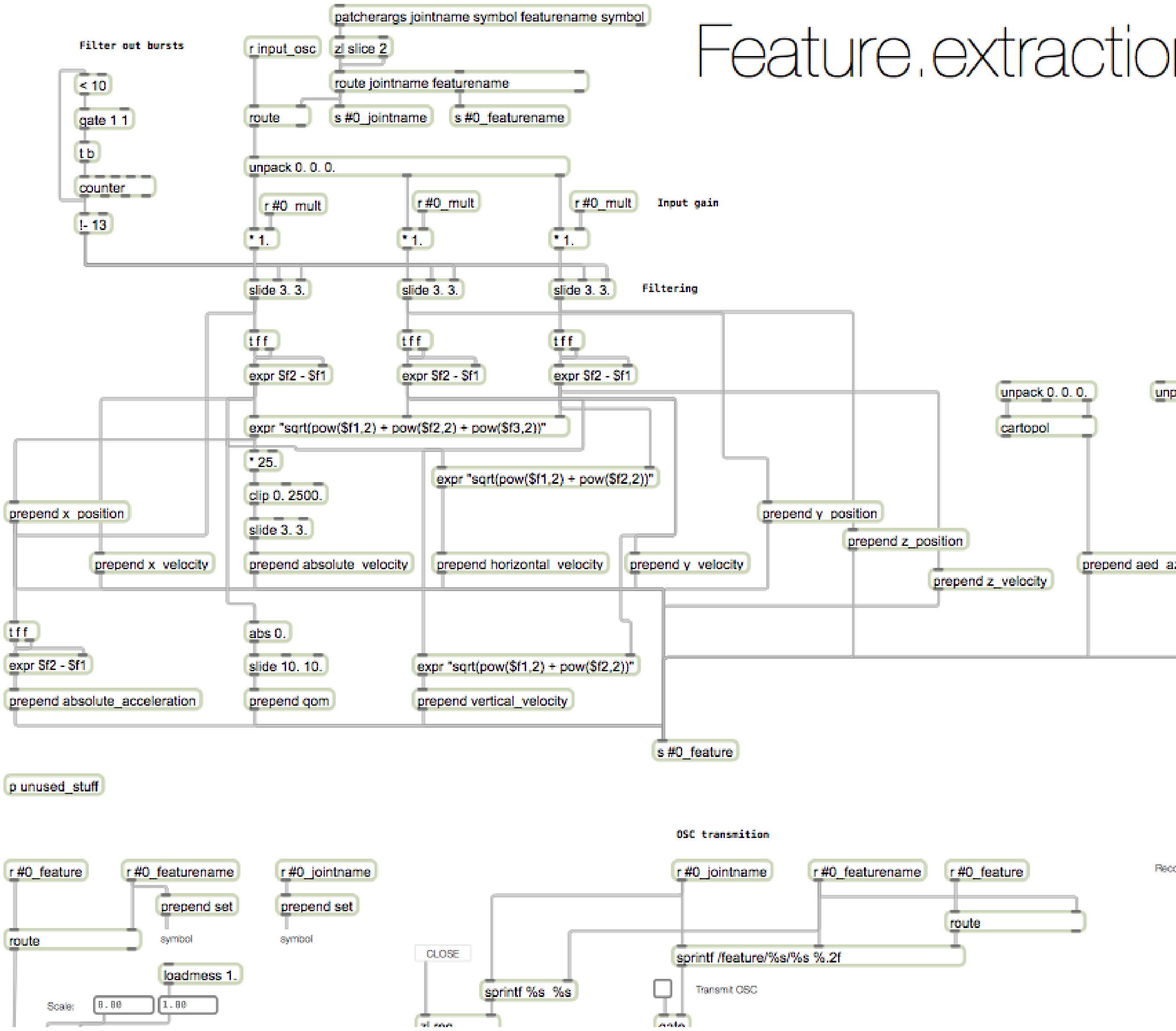


Figure A.13: feature.extraction code (extraction)

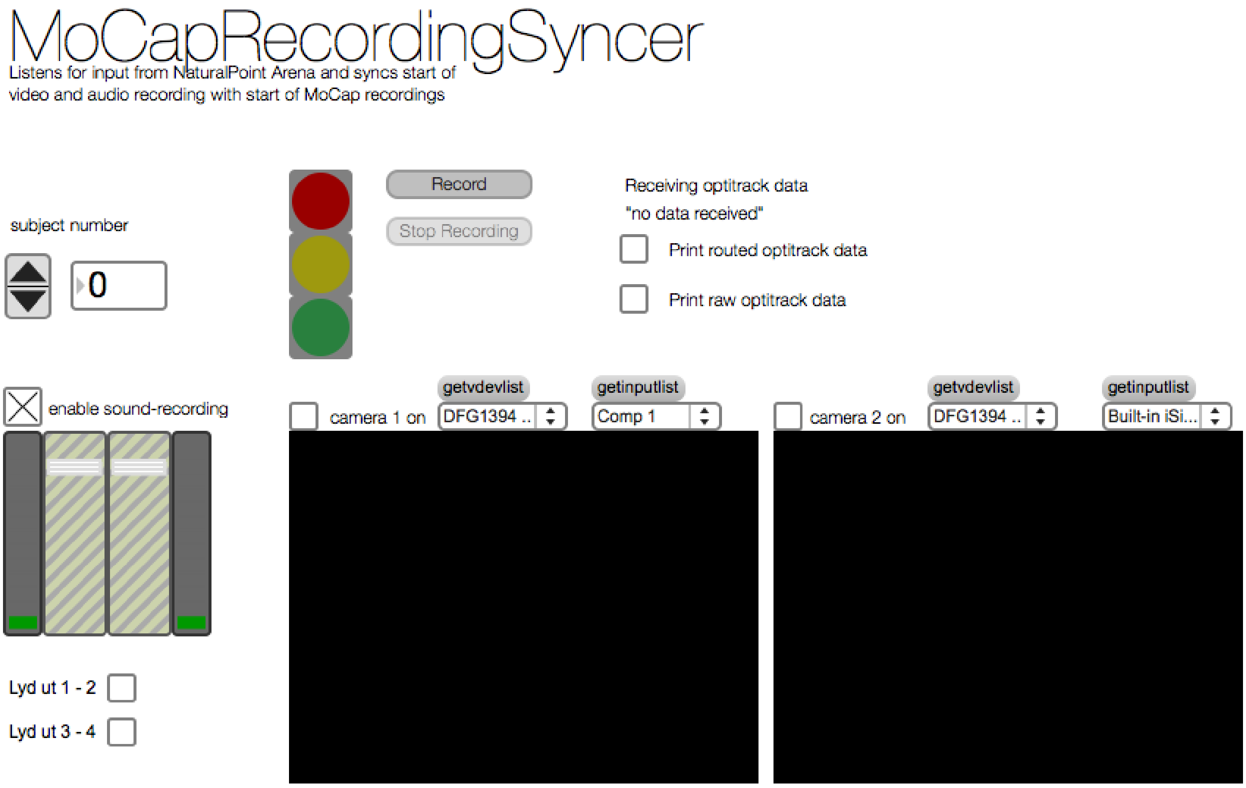
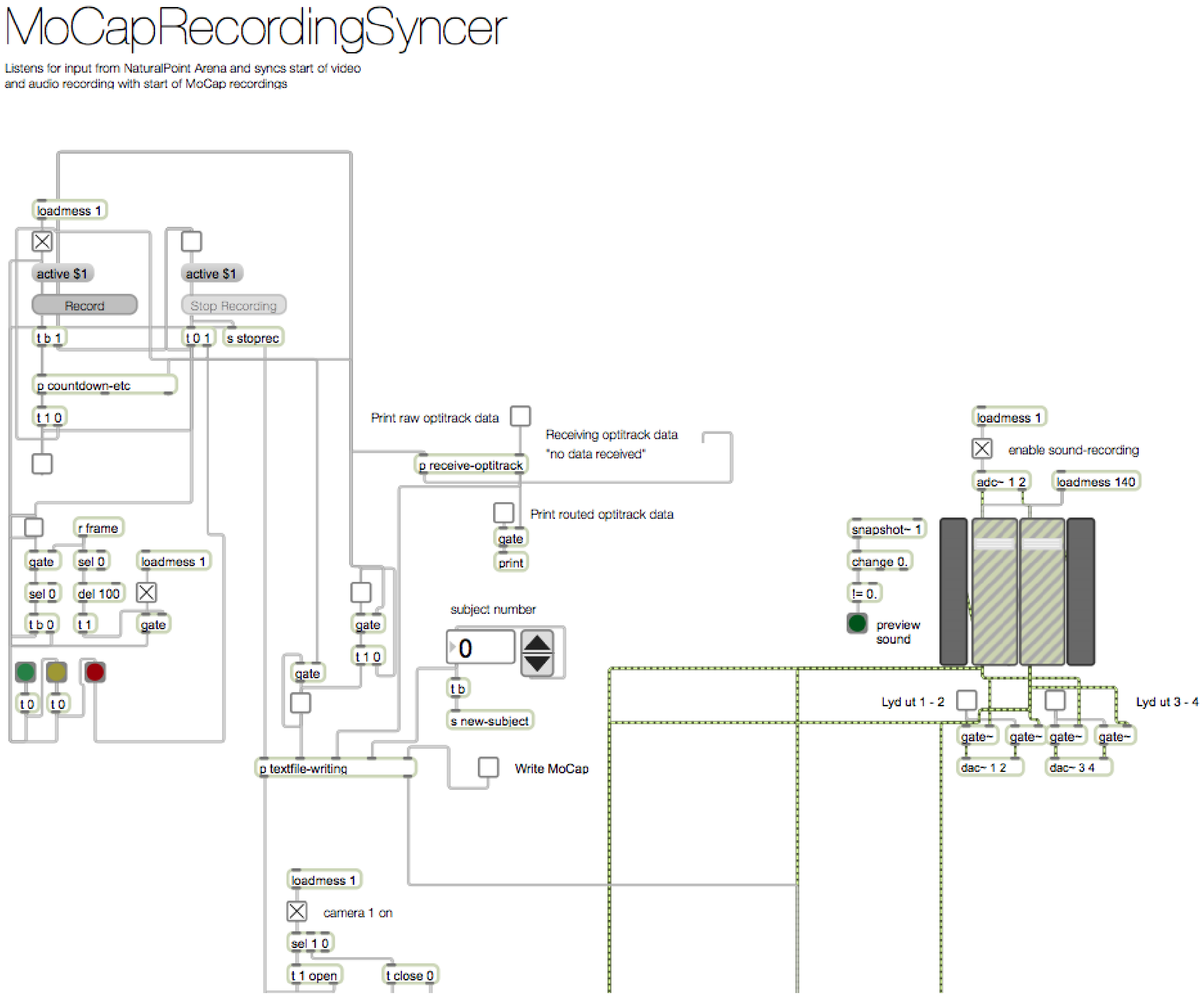


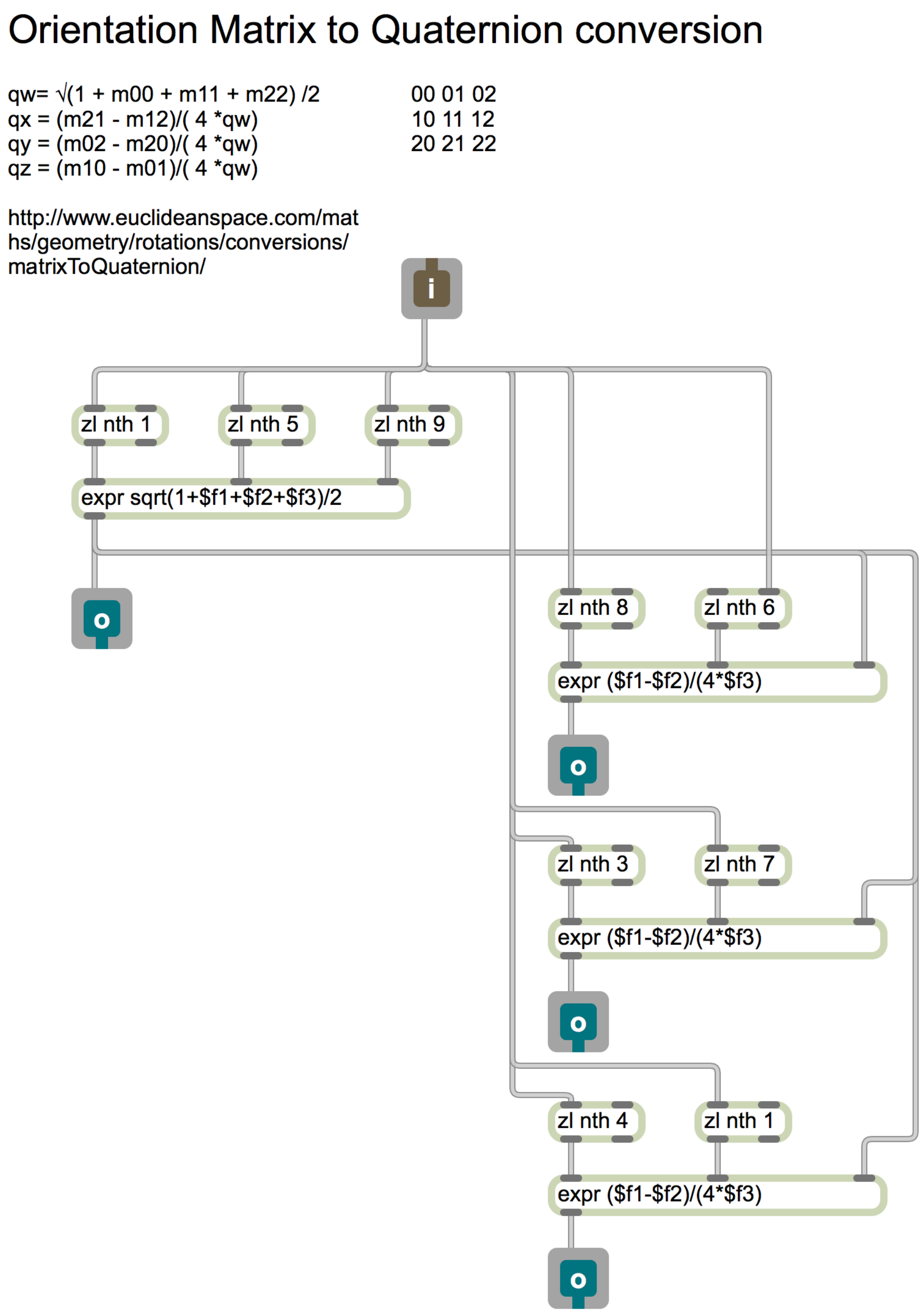
Figure A.14: MoCapRecordingSyncer GUI



MoCapRecordingSyncer

Listens for input from NaturalPoint Arena and syncs start of video and audio recording with start of MoCap recordings

Figure A.15: MoCapRecordingSyncer code (extraction)



Orientation Matrix to Quaternion conversion

qw= √(1 + m00 + m11 + m22) /2
qx = (m21 - m12)/(4 *qw)
qy = (m02 - m20)/(4 *qw)
qz = (m10 - m01)/(4 *qw)

00 01 02
10 11 12
20 21 22

http://www.euclideanspace.com/maths/geometry/rotations/conversions/matrixToQuaternion/

Figure A.16: matrix2quat.maxpat