Mapping Strategies for the Augmented Tango Shoe

by Laura Sinnott

Submitted in partial fulfillment of the requirements for the Master of Music in Music Technology in the Department of Music and Performing Arts Professions in The Steinhardt School New York University
Advisor: Dr. Mary Farbood
January 1, 2008
# Table of Contents

Abstract 3

Chapter 1: Introduction 4 - 7

Chapter 2: Related Work 8 - 28

Chapter 3: Approach & Methodology 29 - 39

Chapter 4: Description of the Augmented Tango Shoe System 40 - 46

Chapter 5: Design, Implementation, Results, & Discussion 47 - 58

Chapter 6: Conclusion & Future Work 59 - 60

Appendix A: Acronyms 61

Bibliography 62 - 68
Abstract

This thesis proposes a mapping strategy based on exploiting perceptual relationships between movement and sound. In the context of computer music, mapping is a process that links a musician or dancer’s physical gestures with musical sound via computer programs and algorithms. This thesis reviews previous work in both interactive dance-music systems and mapping strategies, and also discusses the importance of music and movement relationships. This thesis describes a specific mapping implementation with a novel computer music controller, the Augmented Tango Shoe. It was found that utilizing perceptual parameters in the mapping design helps exploit natural parallels between certain music and movements; leading to a general audience’s increased understanding and appreciation of live computer music performance.
Chapter 1: Introduction

1.1 Human Disconnect in Digital Music Instrument Design

Imagine sitting in a concert hall listening to music so sweet your eyes begin watering, and then you see the passionate expression on the musician’s face and in their body language and it triggers actual tears. What instrument is that musician playing? Most likely, you are imagining a pianist, violinist, or other traditional acoustic instrumentalist, not someone playing a Digital Music Instrument (DMI), Hyperinstrument [74], or other novel computer music controller. Computer musicians should also be capable of bringing an audience to tears, however many computer music concerts leave an audience in confusion, or worse, asleep, partly because the human interaction with the physical instrument is detached from the resulting sound.

Digital music instrument controllers (figure 1.1) are not directly connected to the sound-producing object, unlike acoustic instruments (figure 1.2). For example, a human plucks the string of a violin, and this vibrating string, coupled with the physical properties of the wooden body of the violin, produces the resulting musical sound. On the other hand, when a human depresses a key on a keyboard, a sensor outputs a voltage value that represents the speed and force with which the human used to depress the key. The voltage value is converted to digital data and sent to a synthesizer, which outputs a digital signal that represents sound. The
digital signal is converted back into a voltage signal, and then output as sound by a loudspeaker. The keyboard controller and synthesizer are entirely separate entities than the sound-producing loudspeaker, where again, the violin is both the controller and the sound-producing object.

This disconnect between the physical controller and the sound producing object can be viewed as a freedom, but if a musician hopes to elicit emotional reactions on a similar level as more traditional concerts, it is becomes a problem\(^1\). Human actions provide a bridge between the controller and the sound in acoustic instruments; but in DMIs these actions physically connect with only sensors (or in the case of computer vision systems, connecting with nothing physical at all).

A solution to this problem begins with an in-depth understanding of the relationship between human gestures or movement and musical sound\(^2\). Then, this movement-music connection must be programmed in the form of software (the mapping process). Finally, if the performer understands then exploits the connection (assuming there is a connection), there is a higher chance that they will elicit an emotional reaction from the audience.

Ultimately, whether the human disconnect in DMIs is a problem or advantage is a question of artistic preference. Just the other day, I was involved in a discussion about how literal a musician’s gestures should be when performing music with DMIs. One person loved the idea that you could make music with gestures that were not traditionally related to the final sound. I argued that performances like these are artistically interesting, but more akin to performance art rather than a musical performance. Because we naturally and intuitively associate certain gestures with certain sounds, we have visual expectations associated with certain sounds; a dancer moving smoothly and asynchronously with a strong, pulsing musical accompaniment may appear strange.

There is definitely a place for more abstract music and dance performances, especially because as humans eventually associate new gestures with new sounds, performances that feel abstract now may not be so abstract in the future. Nevertheless, in

---

\(^1\) Calling the disconnect a problem is based on the assumption that an emotional experience is desired, and that the audience is more likely to experience emotions if they perceive familiar sounds with related actions and gestures. This assumption is explored in Chapter 3.

\(^2\) An in-depth understanding would include an intuitive, empirical, sociological, and cultural understanding.
order to elicit emotions from an audience, some kind of connection between the senses and perceptual modes should be considered during composition, choreography, mapping, and controller design.

1.2 Thesis Goals

This thesis implements mapping techniques that consider perceptual relationships between physical movement (gestures) and sound, thereby addressing the human disconnect problem in DMIs.

Mapping is the step that bridges input data with the output sound: to be specific, mapping is a mathematical process that translates digital representations of gestural information into data that controls sound producing algorithms; to be general, it links our senses. In the case of the Augmented Tango Shoe (ATS), a general computer music controller, the mapping process connects what we see (the dancer) to the music that we hear.

Gesture data is not the only data that can be mapped to sound control of DMIs. Features of sound itself can also be mapped to digital audio effects (DAFx); for example, the amplitude of the sound can control the reverb time. This thesis, however, only concerns gestural control of sound.

Potentially, any gestural controller\(^3\) could have been used with the mapping techniques I present, but the example in this thesis used the ATS. The ATS had already been performed with numerous mapping designs, but they were unstable and unreliable. Ultimately, my goal was to design a structured mapping strategy that implemented perceptual parameters, allowing the performer to connect more with an audience during a live performance. In order to achieve this goal, numerous smaller goals must be addressed first. The following describes what I hope to accomplish for this thesis:

---

\(^3\) Gestural controller refers to just the tangible interface, or actual physical instrument part of a DMI that the musician holds or is attached to. They usually have sensors embedded into the instrument to capture movement. In this thesis, interface, controller, and gesture transducer to refer to the same thing.
- Obtain a thorough understanding of what, exactly, the mapping process consists of in the context of DMIs.
- Create a mapping model and set of guidelines that will provide structure for the new mapping system.
- Implement the mapping model for gestural control of DAFx control and sound triggering, not sound synthesis.
- Re-design the ATS hardware so that it is capable of capturing all the gestures that I need for the new mapping system.
- Compose and choreograph a piece using the new mapping design.

Though mapping strategies are the focus of my thesis, an overview of interactive dance and music systems (IDMS) is important to provide a context for my mapping strategies, which are specific for the ATS. Chapter 2 provides this overview, along with descriptions of related work in mapping. Chapter 3 describes a conceptual approach to mapping and methodology for designing a new mapping strategy. Chapter 4 describes details of the ATS system and some of its historical development. Chapter 5 describes the implementation, results, and a discussion, and Chapter 6 offers suggestions for future work and a conclusion.

In closing the introduction, I would like to reiterate the advice of Filatriau and Arfib [46], which is to always think musically while mapping, both logically and emotionally:

Finally, the evaluation of such links between gesture and [sound] must not let us forget an important thing: music is an intense process, not limited to algorithms and mappings; there is an intense implication of emotion, and one always has to remember that textures are not merely “soup music” but can be an awakening of the senses. This means that a musical point of view always must be the guardian of computer sonic research.
Chapter 2: Related Work

2.1 Interactive Dance/Music Systems: Dance Controls Music
   2.1.1 Sensor-Based Systems
      2.1.1.a Shoe Oriented Music Controllers
      2.1.1.b Body Oriented Music Controllers
   2.1.2 Computer Vision (Video Tracking) Based Systems

2.2 Mapping Gesture to Sound
   2.2.1 Gesture Recognition Techniques
   2.2.2 Mapping Strategies, Frameworks, and More
      2.2.2.a Mapping Strategies
      2.2.2.b Controlling Audio Effects rather than Sound Synthesis
      2.2.2.c Other Mapping Frameworks

Numerous systems exist that use dance as a music controller. A great deal of literature provides overviews and details of these systems; some even discuss gesture recognition techniques and mapping strategies. To dig deeper into gesture recognition or mapping strategies, however, one must obtain literature that narrows the scope and focuses solely on these topics. This exemplifies the truly interdisciplinary and collaborative nature of IDMSs, and suggests that in order to design a DMI that compares with traditional acoustic instruments, a team of experts is required. The first half of this chapter describes IDMSs that are relevant to the ATS, and the second half focuses on mapping strategies and frameworks.

2.1 Interactive Dance/Music Systems: Dance Controls Music

These systems can be based on wearable (embedded) sensors or computer vision (video tracking).

2.1.1 Wearable or Embedded Sensor Based Systems

Although the majority of interactive dance/music systems use computer vision [8, 19, 36, 38-40], numerous sensor-based systems do exist, and have existed since the early 1970’s, when Gordon Mumma composed pieces requiring dancers to wear accelerometers to control synthesizers [24]. Though some systems have been consistently performed with since the 1990s [16, 25], new systems are continuously being developed with improved hardware and software to take advantage of the rapidly
improving technology [15, 17, 18, 20]. Most systems focus on the upper body and legs, commonly fixing sensors on joints, wrists, and ankles. Some focus on foot-oriented gestures with sensor-laden floors [4, 9 - 13]. However, this project is concerned mostly with wearable systems employing sensors inside or on a shoe.

2.1.1.a Shoe Oriented Music Controllers

The medical field has been working with in-shoe sensor technology and gait recognition for a while [65], and Nike and Apple together offer a consumer product that uses a piezoelectric accelerometer in the sneaker that sends data to an iPod [66]. The music technology industry has explored in-shoe sensor technology as well, but unfortunately, none of these industries have collaborated with each other. The following is a description of three music technology systems incorporating shoe controllers, followed by a few more projects that have not moved past the prototyping stage.

Miburi

Yamaha’s Miburi system [8, 32] was released commercially in Japan in 1994 (figure 1). Users wore a vest with flex sensors placed near joints, gripped hand controllers with velocity-sensitive push buttons, and placed inserts with piezoelectric sensors into their shoes (figure 2). The system was wireless only in Japan; otherwise, a cable routed data to a MIDI converter [3]. The system was useful for triggering notes but weak in its mapping abilities [3], which were limited to processing gestures in direct relationships to the musical
output. In fact, twelve of the eighteen sensors could only trigger MIDI notes. The other six controlled pitch bend and program changes. L. Vickery [3] collaborated with STEIM designer T. Demeyer and used the Miburi system to control video elements via Max/MSP in many performances. No literature could be found on people adapting Miburi to control music, however. Miburi is not manufactured anymore.

Expressive Footwear and the Gait Shoe

The Miburi system inspired Joe Paradiso and his team to develop expressive footwear in the late 1990s at the MIT Media lab [1, 8]. Their motivation was to design a wireless shoe controller contained entirely on the foot, and with enough sensors to describe all possible foot gestures including: twists, jumps, kicks, weight distribution, distance of foot from floor (using electric field transmitters) and more. Their “perfected” controller had sixteen sensors including force sensing resistors (FSRs) built into the insole, and accelerometers, gyroscopes, sonar receivers among others built into the circuit (figure 3). The circuit was mounted on a sneaker and transmitted data via FM. The sensor data controlled external synthesizers via C++ based libraries. Dancers performed with expressive footwear numerous times in the United States and Paris. However, while expressive footwear’s hardware design was robust, reliable, and sophisticated, the mappings focused on demonstrating the technology with “simple sound-action mappings” [8].

Expressive footwear evolved into “The Gait Shoe” [31, 32], which is currently in development for use with physical and music therapy. This shoe uses clear Type 1 PVC
Mapping Strategies for the Augmented Tango Shoe

(polyvinyl chloride) insoles (figure 4), a circuit that fits onto the back of a shoe (figure 5), and will be integrating sonar sensors to measure the distance between each foot. This last parameter would be meaningful for any shoe controller-based IDMS.

Fig. 5 Clip-on circuit for the Gait Shoe

Pikapika

Pikapika is a performance system designed by T. Hahn and C. Bahn [2]. The system was designed exclusively for Hahn, and with the intention of portraying a “composed character” inspired by Japanese art forms anime and manga. Pikapika is unique in that Hahn herself controls the sound that emanates from a system of small speakers mounted on her body (developed by Bahn and called SSpeaPer). She holds a box with an accelerometer and button in each hand, and sometimes an accelerometer on each foot. The sensor data is sent wirelessly to a computer running Max/MSP.

Pikapika’s design uses single, simple gestures such as raising her wrist past a threshold to trigger one of hundreds of looped soundfiles. Hahn can communicate these simple mappings, or achieve complex sonic textures and with less obvious movement and sound relationships, by “initiating the performance of sonic elements and textures in any order and combination.”

Contrary to the above-mentioned systems, which focus mainly on hardware design, Pikapika takes another step further by placing emphasis on sound design and musical results as well as innovative hardware. The results were successful enough to garner major national attention (figure 6), and demonstrate that solid hardware design must be integrated with carefully designed software and mapping strategies in order for an IDMS to be meaningful for live

Fig. 6 Pikapika NYTimes article.
performance. Unfortunately, not much literature discusses the foot controller part of Pikapika.

**Prototypes**

A. Schneider [7] developed simple but elegant “perform-o-shoes” (figure 7) for a course called “Networked Objects” at the Interactive Telecommunications Program (ITP) at New York University. He embedded a single photo-resistor into the heel of each sneaker and connected it to an Arduino board via a ¼ inch stereo jack drilled into back of the sneaker. For a 2006 demonstration, he controlled the playback speed of a Michael Jackson song with a Max/MSP patch.

The Shoe-Mouse prototype was designed by students at the Chinese University of Hong Kong. This is not a musical controller, but it does use sensors in the shoe and wireless communication to control computer navigation in placement of the traditional mouse.

2.1.1.b Body Oriented Music Controllers

**MIDI Dancer and DIEM Digital Dance System**

MIDI Dancer [16] and the DIEM Digital Dance System [14] are two wearable, body-oriented interactive dance/music systems. Developed in 1994 and 1997 respectively, both systems utilize flex sensors that are positioned on the performers’ joints. Wires from the sensors are connected to a box that is strapped to the performers’ mid-body and transmits information to a receiver via RF. Both systems output MIDI data, however MIDI Dancer uses the *Isadora* software to map data to the output medium (not necessarily sound), whereas the Digital Dance System uses Max/MSP. Though both systems have been used in multiple performances over the past decade, they are cumbersome to wear, and gestures are limited to angular movement of the elbows, knees, wrists, etc.
SICIB

In 2000, Manzanares et al. [21] developed Sistema Interactivo de Composición e Improvisación para Bailarines (interactive system of composition and improvisation for dancers), or SICIB. Its main function is to utilize dance to compose music. However, it is also capable of real-time performance. It uses a magnetic tracker comprised of sensors on a dancer’s body to track three-dimensional position and orientation “within a ten-foot radius around a central transmitter.” Choreographic regions are defined by analyzing sensor positions with respect to a reference point on the main transmitter. Rule sets are then written with Prolog, as well as Aura, Csound, and Escamol. SICIB’s disadvantage is its complex software architecture. Even with a friendly user-interface, fewer software components will likely result in higher performance reliability. Performers without a programming background may not be computer savvy enough to debug the system, whereas a program like Max/MSP is easier to manipulate. However, as useful as Max/MSP is for prototypes and experiments, a commercially available IDMS could use its own software to perform all gesture recognition and mapping directly on the microcontroller.

PAIR and WISEAR

The PAIR and WISEAR system was developed in 2005 by [15]. They are actually two separate systems that can be interchanged with other systems. WISEAR is a Linux based computer that is comprised of a power regulator, the single board computer, and a daughter board. PAIR is a collection of sensors that (in this example) are wired to the daughter board of WISEAR. They employ Max/MSP for audio processing and Isadora for video processing, though the system is designed to work with other software as well.

WISEAR is the strength of this system. Unlike many sensor-based IDMSs,
which transmit mostly RF signals, WISEAR is an 802.11 a/b/g device that sends data wirelessly using Transmission Control Protocol (TCP) reads and writes. Since each box has its own IP address, an unlimited number can communicate simultaneously with audio and video applications [25]. This offers interesting possibilities for interactive performances with groups of dancers. The other strength to WISEAR is its potential to become standardized hardware specifically geared towards interactive dance/music systems, as it moves “beyond custom-made … projects that are often created on a per-performance basis and rely heavily on MIDI.” An enormous downfall to WISEAR, however, is its size (figure 8). Though lightweight, at less than one pound, a large, corset-like elastic wrap would have to be positioned on the dancer’s lower and middle back; this must be incredibly inhibiting for a dancer to wear, especially when back bending!

DanSense and the Sensor Music Project

DanSense [20] is a software package created for the PhD thesis of German student U. Enke in 2006. It is based on the conceptual Sensor Music Project [33]. His focus is on extracting rhythmic data from a person’s natural movements so that he or she can intuitively control music, thereby becoming a musician without formal music training. The motion data is captured with accelerometers, and the system is not wireless. A wired sensor system using only accelerometers was not adequate for his desired results. He did prove that using Fourier transforms and auto-correlation improved the results, but as his conclusion admits, his goals were vaguely met. Wireless technology would vastly improve the system, which could then be used in collaboration with choreographers, composers, or physical therapists. Enke’s thesis is mainly about gesture recognition algorithms, and does not actually discuss musical mappings.

Eco Platform

The Eco platform [17] achieves true wearability with sensor nodes smaller than a dime. The most recent version (2006) consists of an Eco sensor node, a “data aggregator”, and a “base-station board”. The sensor node is smaller than the tip of an
index finger (figure 9) and is packed with a tri-axis accelerometer, temperature and light sensor, power switch, battery, expansion port, and more (figure 10). The node has an RF transmitter that sends data to the Data Aggregator, worn on the dancer’s waist. This is considerably smaller than the WISEAR hardware device discussed earlier.

Like WISEAR, the data aggregator sends data via TCP with an 802.11b card to a wireless access point (the base station board) that connects with a computer. Applications can vary, though they have successfully used the system (in conjunction with Max/MSP and Jitter) in interactive dance/music/video performances. One of Eco’s only disadvantages compared to larger systems is its short battery life of 1.5 hours; communication via TCP is also questioned by some [19]. Overall, the über compact hardware of Eco is impressive and provides an alternative to video-based interactive dance/music systems, often preferred by dancers because they do not require cumbersome devices to be worn on their body.

_Celeritas_

Celeritas is another recent (2006, 2007) IDMS that advertises its use for solo or group performances [18, 19]. It employs the Tyndall Mote [34], a wireless inertial measurement unit (WIMU) comprised of accelerometers, gyroscopes, and magnetometers. Each Mote transmits data over a 2.4 GHz ISM radio band, which offers ‘Use Without Permission’ worldwide, a great advantage compared to previous systems. The base station identifies each node with an “address driven master/slave protocol,” and developers currently use a “Mote” object in Max/MSP and Pure Data (PD) to create
software platforms. Current literature only discusses potential mapping strategies and does not mention any actual performances using Celeritas.

2.1.2 Computer Vision or Spatial Sensing Based Systems

Three computer vision based systems are often mentioned when discussing IDMS [14, 15, 17, 18, 21, 35]. One is David Rokeby’s *Very Nervous System* in 1988 [36] as a set of externals used in Max/MSP that compares changes in successive video frames to extract motion and gestures. Another is *Eyesweb*, a research project that has been in consistent development at the University of Genoa since the late 1990s. Headed by A. Camurri, this system can use wireless body sensors but mainly uses black and white, infrared, and color video cameras to capture three-dimensional movement [26]. Each frame is filtered and quantized to distinguish a dancer’s figure from the background, and parameters based on *Laban’s Theory of Movement* are computed. The third is STEIM’s dated *Bigeye* software system [37], which compares changes in video frames from a reference frame in order to extract motion, and then converts the information to MIDI messages. *Bigeye* is not supported by STEIM anymore, but they suggest using *Isadora*, software by M. Corniglio [39] that includes real-time video processing modules (also used in the MIDI Dancer system [16]). Rising in popularity is Cycling ‘74’s *Jitter* [40], a set of video, matrix, and three-dimensional processing objects for Max/MSP.

Computer vision systems offer an interactive environment where the dancer can move freely in space, without the burden of wearing transmitters, wires, and sensors. Though this is a great benefit for capturing uninhibited motion, there are many disadvantages to computer vision systems. First, lighting is an issue, and therefore most performances must be in spaces with controlled lighting. Second, it is difficult to extract micro movements without expensive cameras and high-resolution video frames, whereas small movements can be easily recognized with wearable sensors. This is true especially for foot-oriented control of sound. It could be difficult for computer vision system to “see” a dancer transfer weight from the inside of the ball of the foot to the outside if they don’t exaggerate the weight transfer with leg or upper body movement as well. A clearer example is to imagine a dancer standing still with his feet together. If he shifts his weight
from his left foot to his right, the change represented by FSRs inside the sole of the shoe would be extreme, where as the change in video frame data would be small, as the body does not move a great deal.

Eyesweb has the right idea - utilizing both computer vision and wearable sensors - but this requires lots of human power, processing power, and equipment. Employing GPS for a spatial sensing IDMS is interesting; though used in augmented reality audio systems such as *Hear&There* [41], no popular IDMS uses it.

### 2.2 Mapping Gesture to Sound

Sonification of physical gestures and movement is based on two principles: gesture recognition and mapping.

#### 2.2.1 Gesture Recognition Techniques

Gesture recognition may be as simple as a switch that turns on when you tap a foot pedal [4] or as complex as recognizing the emotion in a one minute piece of modern dance choreography via motion cues extracted from frame-by-frame video processing [26].

A large quantity of research has been conducted specifically on gesture recognition techniques and algorithms. Most apply complex algorithms utilizing statistical models, machine learning, neural networks and information retrieval techniques.

The Smart Design Studio design environment at the Italdesign-Giugiaro Virtual Reality Center, is one of many systems that use hidden Markov Models (HMMs) in their CAD-based software to recognize gestures [27]. The user presses a button on a physical sensing device called SoapBox, performs a gesture, and releases the button when the gesture is finished. SoapBox sends data from accelerometers, temperature sensors etc. to the software as input to first train (in this case with two repetitions), then recognize gestures.

MnM, a mapping toolbox created with the FTM library for Max/MSP by IRCAM [28], bases their gesture mapping on HMMs and Principle Componant Analysis (PCA) [29].
Kahol et al. [30] proposes an algorithm called “Hierarchical Activity Segmentation,” which uses a naïve Bayesian classifier to eventually “…parse a given motion sequence into gestures.”

Some of the EyesWeb patches use neural networks to recognize static postures from video frame analysis [26].

Paradiso et al. [31] designed a comparatively simpler algorithm that interprets the area under accelerometer data curves from sequential peaks to determine if a gesture is present. This goal of this “inertial measurement framework” is to generalize the design process for sensor-based applications. Attractive in its simplicity, the framework currently only allows gesture recognition in two dimensions.

The focus of this thesis is mapping between pre-recognized gestures and the musical (sound) output. Detailed exploration of gesture recognition is beyond the scope of this thesis. However, the topic must be discussed, and a particular method employed. At the time of writing, I employ elementary gesture recognition techniques I developed in Max/MSP for the ATS.

2.2.2 Mapping: Strategies, Frameworks, and More

Mapping is the bridge between the gesture transducer and sound output; a number crunching process as well as an artistic vision. Mapping design can be a precarious journey due to the many paths one can take from clear, direct approaches with demonstrative intentions, to abstract mappings for performance art. No matter the intended application or audience, it is necessary to define terms and create a general framework from which a composer, choreographer, or programmer can start the design of the mapping strategy.

This section describes related work in mapping strategies, outlines commonly used terms and models, and provides support for my own mapping framework.

Parameters

The key to a successful mapping design is an understanding of all parameters involved. Parameters are characteristics and features of each part of the system that can be quantified. They can be perceptual or physical (loudness or amplitude), micro or
macro (attack time of a note or time length of a song), and can pertain to multiple modalities (height of a dancer’s leap, brightness of a spotlight). Afanador [49] believes that defining parameters “is of the utmost importance because this is the information used by the computer to generate new material.” Furthermore, the number of parameters the gestural transducer transmits will affect the musical capabilities of the system [35], for example, most keyboards that send only one parameter to identify which key was pressed will not have as much musical variation as keyboards that send the additional parameter of velocity or after-touch.

Haptic Feedback

M. Leman points out that many computer music controllers employ haptic feedback. Therefore, physical gestures are “badly reflected in the microstructure of the sound energy,” which can make it challenging for the audience to understand or enjoy. [57 p.163]. D. Arfib et al. points out that once parameter values have been defined, then boundaries are introduced; haptic feedback is necessary to sense these boundaries. This thesis does not focus on haptic feedback, though natural, muscular feedback can be utilized when designing future mapping strategies for the ATS.

Early Work in Mapping

In-depth studies, experiments, and implementations on mapping human gesture to sound have been conducted for over twenty years4. Specific models and frameworks with defined typologies have been presented since at least the early 90s. In 1990, I. Bowler et al. [cited in 72] presented a mapping method using an interpolation scheme that provided continuity between performance parameters and synthesis control parameters. Sidney Fels’s 1994 dissertation used neural networks to map hand gestures to speech [48]. Rovan et al. [cited from 72] designed a performance system in 1997 that implemented new, complex mappings in the Yamaha WX7 wind controller, which demonstrated the importance of mapping and not only “input device design” and “new

4 Some examples are papers written about the Radio Baton (by Max Matthews in 1989) and the MIDI controller called The Hands (by Michel Waisvis in 1985).
Mapping Strategies for the Augmented Tango Shoe

synthesis algorithms.” Hunt et al. [52, 72], mapping researchers since at least 1997, took Rovan’s study further and implemented three mapping strategies ranging from simple, one-to-one mappings to complex, cross-coupled mappings that more accurately represented the behavior of a real wind controller. Results showed that musicians preferred the complex mappings where beginners or non-musicians chose the simple mapping, and that overall:

altering the mapping layer in a digital musical instrument and keeping the interface (an off-the-shelf MIDI controller) and sound source unchanged, the essential quality of the instrument is changed regarding its control and expressive capabilities. [52]

Some authors explored mapping more conceptually, which is necessary for anyone designing mappings from scratch. In a 1995 paper [35], Winkler commented on the importance of the physical and structural relationships between music and dance: “An examination of the physical parameters of movement, their limitations, and the methods used for their measurement, will yield clues to an appropriate musical response” [35]. If the participants are aware of the laws of movement, they can use the computer to break these laws, resulting in, “provocative and intriguing artistic effects.”

Today, plethora of papers and publications exist about general mapping strategies, frameworks and guidelines. Some discuss applications of these strategies in music and dance performances [44, 49], for alternate music controllers, (a general mapping model for human actions to sound synthesis [70]), and for sound effects control as opposed to sound synthesis control [42, 71]. Participants in the New Interfaces for Musical Expression conference, started in 2001, frequently address mapping issues in papers, poster sessions, and demos every year in their international conference.

The next section describes some heavily cited frameworks, typologies, and strategies related to mapping.

2.2.2.a Mapping Strategies

Implicit vs. Explicit

Mapping strategies can fit into one of two categories, implicit or explicit. Implicit mappings are non-linear in nature, and utilize a black box method where the designer
defines the rules but not the exact values [70]. This mapping style, like neural networks or pattern recognition, allows for a desirable degree of unpredictability. Implicit methods use “internal adaptations of the system through training or [selecting the] most important features among the set of signals” [50]. However, to strengthen “human-ness” by means of complete control of the system, it may not be the best strategy.

For successful detailed control, explicit mappings describe each mapping connection by defining input and output parameters and linking them with mathematical expressions [71]. This thesis deals with explicit mappings only.

**Simple vs. Complex Mapping Strategies**

Once parameters have been defined, the next step is to define the relationships between them [50]. There are three general approaches to mapping design: 1) compose sounds first and figure out what gestures works with the sound, 2) design the gestures (choreograph a dance) and then play with sounds match the movement, or 3) alternate between the two approaches. All explicit approaches, however, must define and then assign relationships to parameters. Todoroff [75] describes four assignation strategies, which numerous authors have described as well [44, 70 - 72]:

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Also known as</th>
<th>Example [75]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 1</td>
<td>one-to-one, direct, simple</td>
<td>Pitch mapped to key position on keyboard</td>
</tr>
<tr>
<td>1 – M</td>
<td>one-to-many, divergent, complex</td>
<td>MIDI velocity of key depression mapped to attack time, volume, brightness</td>
</tr>
<tr>
<td>N – 1</td>
<td>many-to-one, convergent, complex</td>
<td>Variety of gestures control volume</td>
</tr>
<tr>
<td>N – M</td>
<td>many-to-many, complex</td>
<td>A combination of the above.</td>
</tr>
</tbody>
</table>

Table 2.1 Parameter relationships.

Complex mappings can improve the expressive power of DMIs [35, 44, 72] because they offer more control over nuance and detail; leading to virtuosic possibilities. On the other hand, direct, or one-to-one mappings can be best for mapping perceptual parameters to each other for “intuitive navigation in perceptual spaces” [71]. They can also result in easy-to-use instruments and for educating the audience. In *Coat of Invisible Notes*, a performance at the University of Leeds [51], direct mappings and simple motions were used at first in order to help the audience understand the relationship.
between the dancers and the music. As the piece progressed, multi-layered\(^5\), complex mapping strategies were introduced, which allowed the audience “to follow the performance with increasing understanding and appreciation.” While not the only approach, I think it is a fine one, especially for a general audience.

Palindrome, a twenty-five year old interactive music and dance company, has learned that “even when only part of an interactive piece is clear and convincing, audiences will attune to and accept more complex relationships” [44].

**Layers: Single Layer**

The next step in mapping design pertains to layers. Layers could be thought of as a mathematical function that translates one (set of) parameter(s) to another. The simplest mapping strategy will employ a single layer (figure 2.11), which could use simple or complex parameter relationships (figure 2.12), but basically requires scaling the input values to output values.

![](image1.png)

*Fig. 2.11* A simple, single-layer mapping model. One slider position determines amplitude of an oscillator, the other determines the frequency. [52]

![](image2.png)

*Fig. 2.12* A single-layer mapping model with many-to-many mapping techniques. The sensor input doesn’t necessarily need to be from a gestural controller, it can be parameters extracted from sound itself (such as fundamental frequency).

**Layers: Two-Layer**

Hunt et al. proposed a two-layer mapping approach [52] that stressed using independent layers:

The intrinsic advantage of this model is its flexibility. Indeed, for the same set of intermediate parameters and synthesis variables, *the second mapping layer is independent of the choice of controller being used*. The same would be true in the other sense: for the same controller and the same set of parameters, *multiple synthesis techniques could be used by just adapting the second mapping layer, the first being held constant*.

---

\(^5\) Multi-layered mapping strategies are discussed in the next section
Multi-layered mappings require the designer to define a new set of parameters between the sensor input data and the synthesis or DAFx control input data. Some authors suggest basing this intermediate layer on perceptual parameters (loudness or brightness) [43, 70, 71], but it can be parameterized in any way. Figure 2.13 represents this intermediate parameter layer by the box with the dashed line.

**Layers: Three-Layer**

Hunt and Wanderley expanded this to three layers in [72] and describe the layers as such:

Layer one: Extraction of meaningful performance parameters (an optional extra layer…[that could be used]…for deriving performance-relevant parameters, such as the player’s energy input to the system).

Layer two: Connection of performer’s (meaningful) parameters to some intermediate representation set of parameters (for instance, perceptual or abstract).

Layer three: Decoding of intermediate parameters into system-specific controls.

This optional first layer could be what they call “interface specific,” and it would extract parameters obtained solely from the sensors attached to the gesture transducer.

Figure 2.14 is a three-layer mapping model with at least one perceptual parameter layer.
Layers: Three-Layer Perceptual

Arfib et al. [71] describes a more detailed three-layer mapping strategy that uses perceptual gesture and sound spaces. Their idea is that pure, quantitative measurements cannot provide as effective and intuitive human-computer interactions compared to qualitative, perceptual measurements. I describe their strategy using physical gestures as an example (the paper also discusses musical gestures and a combination of both).

Static or dynamic parameters are detected from data provided by the sensors on the gesture transducer or controller and are mapped to a perceptual layer with qualitative, “related-to-gesture-perception parameters.” They describe perceptual parameters as abstractions of “raw control inputs” translated into concepts that we can perceive sonically, or in the case of gestures, concepts that we can perceive visually or kinesthetically. These parameters are applied as axis on gesture perceptual space. Next, the gesture-perception parameters are mapped directly to another perceptual layer describing “related-to-sound-perception-parameters,” thereby creating a sound perceptual space (psychoacoustic space). Finally, these parameters are mapped to “synthesis-model parameters.”

One of their examples describes “The Voicer,” a DIM using a Wacom tablet and joystick as the gestural controller (interface) and a source-filter algorithm that synthesizes a singing voice. Numerous parameters are used according to the three-layer perceptual mapping strategy, for example: pressure of the stylus on the tablet (quantitative gesture data) is converted by means of a lookup table to “strength and expressiveness of the gesture” (perceptual gesture parameters), and strength is mapped to loudness (perceptual sound parameters) and level (synthesis parameters). The scaled position of the joystick represented tilt, which was mapped to “vowel space” (see figure 2.15) with one dimension of ‘acuteness,’ another as ‘openness’, and the third axis a linear combination labeled ‘smallness’, and the position in vowel space controlled filter coefficients and gain correction.
Working with perceptual parameters and multi-layered techniques may seem complex on the surface, but it can actually simplify the mapping design [42, 43], because once you are working with perceptual parameters, simple one-to-one mapping strategies provide effective results: “What is direct from the perceptual modification point of view may be indirect from the signal processing point of view, and vice versa” [42]. In can be looked at like a trade-off between “simplifying the mappings and making the layer structure more complex” [43].

Other Mapping Types

Numerous additional mapping techniques have been discussed. When controlling vibrato, for example, high level control would include rate and depth mapping, whereas low level control would include pitch mapping [71]. Adaptive mapping combines dynamic and static mapping and refers to one that morphs over time, for instance when one timbre morphs into another timbre [71]. Mapping can also determines a performer’s local or global control over digital audio effects (DAFx), such as modifying bounds or changing entire presets [42].

2.2.2.b Controlling Audio Effects rather than Sound Synthesis

Physical and musical gestures in DMIs can be sonified in the following ways: 1) playback a prerecorded sound (triggering), 2) synthesis (creating), and 3) signal processing (modifying). The study of mapping gestures to DAFx has been neglected [42], perhaps because it is considered less complicated, but also because people do not think of effects as being performable. However, “audio effects can also be performed” [42 citing Todoroff]. This notion of ‘performing’ audio effects is a similar outlook author Blesser [47] takes when pointing out that a sound mixer exercises musical and creative control of a song (by manipulating reverbs to change chord tonality, for example). Though controlling a DAFx indirectly modifies the sound, one can still indeed ‘perform’ an effect, especially if the mapping has been designed around that intention. In fact, those “indirect” modifications can have direct results, such as when increasing the reverb time can cause an increase in volume level.
Verfaille et al. [42] present a paper that investigates gestural and adaptive control of DAFx, which is extremely relevant to the ATS since it mainly controls global events and DAFx. They present “an explicit multi-layer mapping strategy that clearly separates adaptive and gestural control,” and give numerous examples using the model. Though the work on adaptive control comprises much of their paper, the focuses of this thesis is on gestural control, so will not give adaptive control much attention.

Their strategy consists of two levels: 1) a gestural control level with two mapping layers and 2) an adaptive control level with two mapping layers and sublayers. Each level requires feature extraction (a form of mapping) that is computed mathematically (figure 2.16). The adaptive control’s sublayers apply warping functions (usually producing non-linear results), normalization, and combination of multiple sound feature parameters into one. The gestural control layers can modify the adaptive control’s parameters. Both levels map their extracted features into DAFx control parameters. Since the control levels are independent of each other, the user should be able to disable one and still use the instrument.

Low-level gestural control can be used for direct mapping to basic effects parameters such as a fader position mapping to volume. High-level control of adaptive effects (not the same as adaptive control) can be used at a global level, such as gestural interpolation between different reverb parameter presets, or a local level, by modifying adaptive control parameters.

One example described in [42] is adaptive spatialization. Verfaille designed a Max/MSP patch that allowed a dancer to control granular synthesis and spatialization

---

6 Adaptive control uses sound feature extraction to define parameters for DAFx control.
among eight loudspeakers. They used ultra sound beam sensors to pick up the dancer’s horizontal and vertical position, speed, and acceleration. First, the dancer’s vertical position determined grain position. Next, speed and acceleration were mapped to comb filters, delay effects, and grain pitch-shifting. After the sound was synthesized, features were extracted from it (RMS and spectral centroid), then combined with the gesture parameters, then mapped to parameters such as sound motion (among a particular trajectory), source width, and more. The dancer’s position could also directly control sound position gesturally by manipulating “…using the horizontal ultra-beam sensor and with rules such as point or axis symmetry, shifting and scaling.” A more detailed account can be found in [42].

Another example is electronic musician L. Crawford’s mapping design for his MIDI Air Guitar gestural controller [67]. In one performance, he controlled DAFx applied to sounds from a live drummer [68] with two layered, one-to-one mappings. He applied a warping function to the gravitational tilt reading from an accelerometer, then directly controlled transposition values of a pitch-shifter.

### 2.2.2.c Other Mapping Frameworks

Fenza et al. presents a mapping multilevel and four-layer model (based on a conceptual framework by Antonio Camurri) designed to explore the relationships between sound and “movement spaces” [45]. Their model implements two three-dimensional expressive spaces. The first, an “expressive space” representing physical movement, is defined by cues such as directness and fluency (inspired by Laban’s Theory of Effort). The second, an “expressive space” representing music synthesis control, is comprised of axis correlated to tempo (as related to kinematics), energy (as related to intensity), and brightness. This framework seems to be a successful mapping strategy for IMDSs using computer vision techniques.

Filatriau and Arfib [46] explore mapping by thinking of a sound texture first, and then asking what kind of gesture would best produce the sound. Their framework suggests connecting gesture sensing parameters with sound algorithmic parameters by

---

7 Though similar to the perceptual spaces presented in [map3], these authors point out that their 3D expressive space and framework significantly differ from Arfib’s.
recovering gestural *intention* and mapping it to a musical *expression*. Though an interesting model, it is still hypothetical in nature and seems to focus on definition of terms. They suggest that more work in gesture and sonic textures is needed.

*Mapping Toolkits: Applications of Frameworks*

The numerous publications about mapping techniques and frameworks have prompted people to design mapping toolkits and software libraries, including *MnM* [28], a set of externals designed for Max/MSP users, *Mapping Library* for PD, a “fledgling library of mapping primitives with the aim of cataloging existing mapping methods” [73], and Jamoma, a system of modules and components designed to help build high level Max patches [53].
Chapter 3: Approach & Methodology

3.1 Conceptual Approach: Gestural Coherence
3.1.1 Why Gestural Coherence?
3.1.2 Relationships between Music and Dance: Intuitive vs. Empirical
3.1.3 Embodied Music Cognition and Cross-Modal Perception
3.2 Guidelines for a "Good Mapping"
3.3 Methodology
3.3.1 Design a Mapping Model
3.3.2 Implement the Model

I adopted a conceptual approach to mapping that keeps the perceptual relationship between music and dance in mind. This approach provides support for my mapping model; it also gives direction for making artistic choices while implementing the model. From this approach and other previous research (section 2.2 & 3.1), I derived a set of guidelines and criteria the final mapping design should adhere to.

3.1 Conceptual Approach: Gestural Coherence

There are numerous conceptual approaches to mapping, or translating, physical gestures to sound. The two extremes are direct and abstract. Direct mapping offers an explicit and obvious relationship between music and dance to an audience, whereas an abstract mapping offers relationships that may only be known to the performer or by examining the mapping algorithm itself\(^8\) [44].

I took a partly abstract but mostly direct, and more specifically, gesturally coherent, approach to mapping. The term gestural coherence\(^9\) (also discussed in [55]), is adopted from Palindrome Intermedia Performance Group’s [44] approach to mapping for IDMS. It states that IDMSs should acknowledge the perceptual similarities and parallels between a physical gesture and the resultant sound, or more specifically, the perceptual relationship between music and dance. They support believable relationships between the dance and music in interactive systems, stating that:

\(^8\) This is a different notion than the idea of invisibility, which author S. Dixon [58] explains is when the audience is not necessarily aware of the computer–performer relationship. He says that a “less-is-more” approach results in more sensitive and sophisticated interactions compared to the “rough and reactive” interactions of older IDMSs. An invisible system can still either use direct or abstract mapping approaches. The ATS does not strive to for invisibility (at this time).

\(^9\) Multiple sources uses the word “congruence” in reference to the perceptual similarities between vision or other modalities and the resultant sound [49, 60, 62].
“Believability depends upon gestural coherence…”
Gestural coherence happens through a mapping system that “mediates the two parallel structural systems (musical and choreographic).”
The music structure emerges from dance gestures through a schema that mixes simple and complex mapping strategies.

[44]

These believable relationships are often based on assumptions, intuitions, and expectations we develop according to our culture and environment (see section 3.1.1). Recently, interdisciplinary research comprised of cognitive science, neuroscience, psychology, and musicology [49, 54, 56, 57, 59, 60] has investigated this topic more empirically; some of which has been conducted specifically in the context of mapping [49, 57]. This is helpful for mapping designers because although mapping strategies have significantly evolved, “…such mapping is rarely based on perceptual or cognitive data, but rather on personal intuitions or unexamined shared suppositions…may help to establish musical motional mapping algorithms on a more solid empirical base” [56].

The rest of this section discusses support for the gestural coherence approach and why it is effective.

3.1.1 Why Gestural Coherence?

The gestural coherence approach to mapping will be effective and satisfying for a general audience, based on numerous conversations I have had with computer music audience members, along with my desire to easily connect with the audience.

Familiarity with more traditional music, instruments, and dance creates expectations for gestures and their associated sound. Why? According to some authors [56, 57], people tend to make associations between familiar sounds and actions with their corresponding sound-producing object, even when we hear the sound without seeing the object. Therefore, it makes sense that we associate musical sounds with the instruments that produce them, even without seeing the instrument. For example, if we hear a loud, inharmonic bang, seeing a sharp, kicking motion will be gesturally coherent with the corresponding sound; as this gesture is similar to the action performed (striking a drum) when producing the sound. However, if a gesture is completely unrelated to the
accompanying sound (such as a slow head roll with a bang on a drum), the audience will most likely not perceive a relationship between the two; and if there is one, may have to learn what the relationship is. This is not desired in performances with the ATS. The audience should understand the movement-sound relationship without having to deeply analyze the situation, and exploiting principles they are already familiar with should help achieve this.

Another reason why gesturally coherent mappings elicit more satisfaction from an audience compared to abstract mappings is due to the intertwined nature of music and movement; they can connect our senses. Stimulating multiple senses in a coherent manner can enhance expression in each domain, carrying the audience closer to what Leman calls a “peak experience”\(^{10}\)."

The next section discusses intuitive relationships between music and dance, and why there is a need for empirical and scientific studies about these relationships.

### 3.1.2 Relationships between Music and Dance: Intuitive vs. Empirical

Many assumptions can be made about relationships between music and dance. For example, we can assume that music with an obvious beat will make people move. If you look around at a nightclub playing music with a steady pulse, people usually react with parallel, physical motions we call ‘dance’ (i.e. head bobbing, toe-tapping, or foot shuffling). Other assumptions are: “the effect of a physical gestures appears in the musical gesture” [71], and, a musical gesture “is an action and a perception of movement” [46]. Finally, we assume that when watching a dance performance, certain physical gestures will ‘match’ the music more than others [54].

These types of intuitive assumptions are probably used more often than scientific and objective evidence while choreographing dances. Many choreographers are no doubt knowledgeable about psychology, cognitive science, neuroscience, etc., but from my

\(^{10}\) Leman suggests that people seek experiences with sound energy (and other energies) in order to get moved and absorbed by it, resulting in a feeling of connecting with reality. This ultimately contributes to a feeling of self-reward, with effects of happiness and wellbeing [57, p4].
experience\textsuperscript{11}, the majorities do not apply this knowledge explicitly to dance. Music and dance are, after all, subjective art forms, so this is no surprise. Furthermore, dancers and choreographers from around the world and throughout time have had successful results so far, so what benefit would empirical evidence about the music-dance relationship offer? For one thing, IDMSs requires extensive knowledge about technology, computer programming, and mathematical formulae, things that are, at the core, not as intuitive or subjective to work with. IDMS designers could try to exploit these empirical findings as mathematical representations in order to inject more consistent human qualities into the technological side of IDMSs. Here is a familiar example: we know from psychoacoustics that we do not perceive sound intensity linearly. Over the years, researchers created a general perceptual loudness curve, and mapping designers can now use this curve in a function in order to control amplitude in a manner that is relevant to human perception. To my knowledge, no generally accepted perceptual curves like this exist that relate movement/sound parallels, but if researchers did create such curves, it would surely be helpful for many DMI and IDMS designers.

Empirical and scientific evidence can also be useful for choreographing dance for atonal music, where the intuitive relationship between music and dance may not be so apparent:

C. Casciato et al.’s [61] abstract entitled “Studying Free Dance Movement to Music” (2005) is an observational study of how modern dancers intuitively moved to atonal music. Results showed that dancers were likely to make movements that resembled corresponding sound-producing gestures (with acoustic instruments). They also showed that correlations of low-level features imply a “correspondence between the amplitudes of the spectral centroid and gestural features.” Though not an empirical study, it is an example of a study that could turn into an empirical one (and the authors express desire for this) that shows that “dance movements might encode information about the neuro-cognitive structures of music perception.”

The next section describes empirically based studies about the relationship between sound, movement, and vision.

\textsuperscript{11} “…my experience…” meaning discussions with dance choreographers, classes in choreography, choreographing dances, and personal observations.
3.1.3 Embodied Music Cognition & Cross-Modal Perception

Embodied music cognition offers an approach to music research, and specifically to the design of IMDSs, that takes into consideration relationships between music, the brain, and the physical body. In the book presenting the idea [57], M. Leman presents a music communication model that states: “musical communication is based on the sharing of neural structures that pertain to movement” [p.161]. He says that the theory of action-perception couplings can potentially explain the connections between music and movement, and that the “multisensory basis of listening to music.” Though just a framework, it is based on significant neuroscientific evidence such as the finding that action and perception share brain pathways [21, 27], and even imagined actions and perceptions [63]. This could explain why sometimes people watch a dancer and feel as if they are experiencing the movement, or why motor areas in the brain are sometimes activated when imagining a sound. Even just the perception of tempo can cause an internal sense of motion [64].

An attempt to explain embodied music cognition pathway in the context of DMIs could be [adapted from 57, p.160-61]:

- The performer thinks of a musical intention, and then performs physical gestures in order to realize this intention.
- These gestures are transferred to a gestural controller (transducer) which “encodes” the gesture in binary data form.
- The mapping section of the DMI translates this “biomechanical energy” representation into “sound energy” representations.
- A D/A converter and loudspeaker turns the representations into real sound energy.
- Various forms of feedback (haptic, visual, aural) may or may not be given back to the performer.
- Finally, the listener (audience) perceives the sound energy (music) and the visual information (performer’s gestures), decodes the physical gestures encoded in the music, and imitates, physically and/or mentally, “the music’s underlying articulations.”
Leman points out that these imitations are not necessarily literal imitations, but some kind of reaction that allows the audience to experience a common, neurally encoded framework they share with the performer.

If this pathway is valid, even occasionally, it provides an explanation to why an audience-performer connection can be exciting and enjoyable: the audience not only understands, but feels the gesture-music relationship. For example, if a dancer performs a boleo, a circular kick executed with a whipping action, the music should reflect this gesture with something like a fast crescendo or an arpeggiated chord. This could heighten the audience’s experience of the music and dance with a sensation of themselves “imitating” the boleo.

Along with embodied music cognition, studies in cross-modal perception have shown perceptual coherences between temporally similar audio, visual, and kinetic experiences. Afanador [49] discusses some of these studies in her thesis, for example latency effects: Research supports the idea of a temporal window of debatable length (but within 400 ms) where an auditory and visual event will be simultaneously perceived, sometimes as a single event (pp. 23-24). She also discusses empirical studies about auditory capture that show that “the combination of two media produces unintended but complementary audiovisual experiences for the viewer” [p. 34]. One example by R. Mitchell and M. Gallaher [60] demonstrated that even when music and dance are not meant to “go together,” the music will often influence the observer to perceive matches or “congruence” between the music and dance. Another audio capture study by L. Shams et. al [59] showed that “when a single flash of light is accompanied by multiple auditory beeps, the single flash is perceived as multiple flashes.”

These types of findings can be helpful when designing mapping strategies for IDMs if the performer hopes the listener will make sense of the performance like they do with more traditional musical performances.

### 3.2 General Guidelines

Embedding sound-movement relationships into the mapping strategy for IDMSs should help the audience make sense of the instrument (controller) and connect with the
performer. This section offers guidelines I used to create the new mappings for the ATS. To re-iterate, this is neither the only, nor the absolute best route to take, but I believe it will be effective for live performance with the ATS.

3.2.1 A “Good Mapping”

What makes a “good mapping?” Some authors have expressed their opinion:

• When we ‘feel’ the sound with easy movements [71].
• When they are intuitive, and when they afford the maximum degree of expression with minimal cognitive load [69].
• When the gesture or movement in the physical domain is tightly coupled … with the intention of the musician” [69].
• When the audience understands to a certain degree the relationship between the sound and movement [interpretation of 44, 57].
• “Mapping strategies should focus and harness the decisive qualities or parameters of the movement and sound, while acknowledging the perceptual dimensions of dance and music” [44].
• Mappings should use natural gestures (what comes to mind when you hear a sound) and educated gestures (one you can learn and reproduce) [46].

In addition to the above opinions and the previous work described in Chapter 2, the mapping design of the ATS should also adhere to the following:

• Mappings should include complex parameter relationships.
• Mappings should be intuitive and exploit cognitive and environmental principles.
• Mappings should utilize independent layers.
• Mappings should utilize perceptual layers and parameters.
• Haptic feedback should be considered in the mapping design.
• Physical and perceptual relationships between movement and sound gestures should be considered in the mapping design.
3.2.2 Table of Sound and Movement Analogies

Table 3.1 (continued on the next page) lists examples presented by various authors that describe gestural coherence. The first section lists parallels between music and movement, the second lists shared perceptual qualities of music and movement, and the third does not mention movement but lists examples of perceptual and physical sound analogies. This table is by no means exhaustive, and subject to debate. These analogies have been explored and tested by psychologists [49], choreographers [49], music technologists [56], musicologists [57], etc. D. Levitin et. al [69] speculate that many of these analogies became valid during evolution, throughout which our brains incorporated specific (environmental) principles of the world around us.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Music example(s)</th>
<th>Dance examples(s)</th>
<th>Cited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulse</td>
<td>Increase tempo</td>
<td>Increase number of gestures (faster movements)</td>
<td>[56]</td>
</tr>
<tr>
<td></td>
<td>Crescendi</td>
<td>Speed up gestures (faster movements)</td>
<td>[56]</td>
</tr>
<tr>
<td>Dynamics</td>
<td>Decrescendo</td>
<td>Movements become smaller (body travels through smaller amounts of space)</td>
<td>[56]</td>
</tr>
<tr>
<td></td>
<td>Louder/softer</td>
<td>Faster/slower</td>
<td>[64]</td>
</tr>
<tr>
<td></td>
<td>Louder</td>
<td>Harder (stomping feet)</td>
<td>[69]</td>
</tr>
<tr>
<td>Pitch</td>
<td>Pitch wiggle/vibrato</td>
<td>Gestural wiggling</td>
<td>[69]</td>
</tr>
<tr>
<td></td>
<td>Decreases</td>
<td>Falling (not other other way around)</td>
<td>[56]</td>
</tr>
<tr>
<td></td>
<td>Higher in pitch</td>
<td>tighter</td>
<td>[69]</td>
</tr>
<tr>
<td>Energy</td>
<td>Amplitude/intensity</td>
<td>Motion/quantity of body movement</td>
<td>[72,57]</td>
</tr>
<tr>
<td>Time</td>
<td>Both evolve on comparable…</td>
<td>…time scales</td>
<td>[45]</td>
</tr>
<tr>
<td></td>
<td>Staccato/legato</td>
<td>Kinematic tempo</td>
<td>[45]</td>
</tr>
</tbody>
</table>

Shared Perceptual Qualities of Music and Movement

| Rhythmic  | Accent, meter   | 49 citing Hodglin |
| Dynamic   | Volume of music gesture | Volume of choreographic gesture |
| Textural  | Number of instruments/performers, counterpoint, |
| Structural| Corresponding motives, phrases, structures |
| Qualitative| Parallels of timbre, articulation, dissonance/consonance |
Mimetic | Choreography imitating a particular sound in the music  
---|---  
Physical and Perceptual Sound Analogies  
Physical | Perceptual  
spectrum centroid | brightness  
harmonic jitter | shimmer  
loudness | boldness, restless, powerful  

Table 3.1 Perceptual and physical analogies between music, movement, and sound in general.

### 3.3 Methodology

I now present a methodology for designing a mapping strategy for the ATS. There are two main steps: 1) create a mapping model, and 2) implement the model. The details of the actual implementation are described in Chapter 5.

#### 3.3.1 Design a Mapping Model

As we know by now, there is no single, correct way to approach mapping. In fact, numerous techniques exist (sections 2.2.2, 3.2), and different strategies may be more appropriate in the future as technology evolves and more research is conducted. A common method, however, is to design a model that describes that the mapping process and provides a frame of reference. The previous models and frameworks described in Chapter 2 do not exactly fit the ATS. The DAFx model [figure 2.16] is partly appropriate since the ATS mainly controls effects, but it places significant focus on effects control via sound feature extraction, something the ATS does not do. The model in [72] comes closest, but is not quite detailed enough (figure 2.14 is very similar to the model in [72]). Therefore, I devised a mapping model specifically for the ATS, which is described in detail in Chapter 5.

#### 3.3.2 Implement the Model

The two sections of the mapping model described in Chapter 5 should be separately implemented (as software) in order to create independent sections. This will enable any gesture transducer to be used with the sound mapping section, and vice versa.

The following objectives (in any order) should be defined before implementing the sound mapping section:

- **Create the musical structure of a song:** Will it be a solo or will there be other performers executing the song? Will the performer generally focus on controlling melodies, rhythms or both?
- **Compose/design sound files/loops:** These could be drum loops, melodic phrases, recordings of acoustic instruments, etc.
- **Describe perceptual and qualitative characteristics of the (desired) sounds:** dark, bright, legato, staccato, harmonic, inharmonic, loud, soft, etc.
- **Define perceptual and qualitative parameters you want to manipulate and control:** Loudness, pitch, harmonicity, darkness, spaciousness, clarity, etc.
- **Decide what specific effects to apply to the sounds/loops:** Reverb, delays, pitch-shifting, time-warping, spectral convolution, etc.
- **Relate perceptual and qualitative parameters with quantitative parameters:** Loudness to amplitude, pitch to pitch scaling factor, darkness to low-pass filter cut-off, clarity to reverb time, etc.

The following objectives (in any order) should be defined before implementing the gesture mapping section:

- **Create general structure of the dance:** Will the dance consist of a few gestures with many pauses? Will it focus on the upper body, lower body, or both? Will it cover a lot of space or remain in one place?
- **Choreograph educated gestures:** These could be static gestures that occur in a moment, like stepping onto the left foot, or dynamic gestures that occur over time, like slowly raising an arm. There could be a single gesture (a kick) or a pattern of gestures (a series of kicks that get faster and higher each time).
- **Describe perceptual and qualitative characteristics of the movements:** Fluid, strong, graceful, sharp, energetic, etc.
- **Relate perceptual and qualitative parameters with quantitative parameters:** Fluidity and gracefulness to steady, non-fluctuating parameter readings, strength to acceleration, sharpness to the slope of a peak, energy to amount of motion.

---

12 An educated gesture is one that can be learned and consistently performed [45].
After sketching out these objectives, they must be organized and converted into a computer program via software like Max/MSP, Csound, or Eyesweb. Finally, in order to bring the mapping design to life, one or more “songs,” or performances, need to be composed, or choreographed.
Chapter 4: Description of the Augmented Tango Shoe System

4.1 The Controller
   4.1.1 Sensors
   4.1.2 Hardware
   4.1.3 Communications Protocol
4.2 Software & Mapping
4.3 Sound Design & Choreography

Each component of an IDMS provides limits that affect its mapping design; an understanding of these components will result in a more efficient and stable mapping design. Thus, before describing the actual mapping implementation process, a description of the ATS system is provided.

The ATS is a general computer music controller designed to trigger sounds and control effects via gestures and movement created with the legs and feet. Combined with a Max/MSP patch and pre-composed sounds, it becomes an instrument that gives musical control to ‘open air’ gestures [77]. It also allows the performer to, as J. Rovan enticingly describes, “magically conjure music from ‘thin air’ [77]. Figure 4.1 shows the general

Fig. 4.1 Overview of components in the ATS system’s path.
path the ATS system follows. It begins with controller, which is a tango shoe outfitted with sensors, a microcontroller, and a Bluetooth module, and then involves a computer running a Max/MSP patch that translates (maps) sensor data into musical sound.

## 4.1 The Controller

The controller is a pair of 2.5-inch high-heeled shoes designed and made specifically for dancing Argentine Tango. Both shoes are augmented with a variety of sensors that are wired to a 15-pin serial connector. Once the performer puts on the shoes, they strap a 3.5 x 2.75 x 0.75 inch circuit board plus a single AA battery around each ankle. A switch lets the battery power the circuit board and sensors.

### 4.1.1 Sensors

The first ATS prototype had one dual-axis accelerometer and two force sensing resistors (FSRs) on each shoe for a total of eight sensor output readings. For the second prototype, the dual-axis accelerometer was replaced with a tri-axis accelerometer on one foot, and a dual-axis gyroscope was added to the other foot, increasing the number of sensor output readings to eleven. This third prototype employs three FSRs and a tri-axi accelerometer on each foot, plus a dual-axis accelerometer with a higher g-force rating on the right shoe and a dual-axis gyroscope on the other shoe, bringing the total degrees of freedom to sixteen.

The FSRs are resistors that change value depending on the amount of force applied on to the sensor. The FSRs’ sensitivity range is $< 100 \text{g}$ to $> 10 \text{kg}$, and the output voltage and voltage swing can be altered depending on the measuring resistor value in a voltage divider. Figure 4.2 describes the output voltage equation. I used smaller resistor values for the toe and ball FSRs compared to the heel FSR in order to out the output voltage values. I wanted to use the heel FSR is more like a switch, and thus used a larger measuring resistor, which provided a higher voltage output when a minimal amount of pressure is applied to the FSR.
All three FSRs lay between two layers of a Dr. Scholl’s foam insole: one beneath the big toe, one under the ball of the foot, and one under the center of the heel. Compared to previous prototypes, this configuration provides more information to perform gait tracking, which is useful since there are different types of tango walking. For example, when walking backwards the toe will strike first, then the ball of the foot; the heel rarely meets the floor. With only two FSRs, you would miss the toe strike.

The ATS uses Analog Device’s ADXL330 tri-axis accelerometer with breakout boards designed by Sparkfun and Leah Buechley (figure 4.3) [79]. They measure tilt (or ‘static acceleration’ with respect to the earth’s gravity) and dynamic acceleration within a range of +/- 3g in three dimensions. The dual-axis accelerometer is Analog Device’s ADXL210AE with a Sparkfun breakout board.

It measures tilt and dynamic acceleration as well, but between a range of +/- 10g. This only one axis of this sensor is used (due to the number of inputs on the microcontroller) and is used for recognizing kicks and jump landings.

Analog Device’s IDG-300 dual-axis gyroscope (figure 4.4) can measure the rate of rotation up to 300 degrees per second. The ATS uses only one axis, but
this is sufficient for measuring speed of spins where one of the feet is the pivot center of the spin.

It would be useful to employ an infrared or sonar sensor to measure the distance between each foot, but the current prototype does not support those sensors.

### 4.1.2 Ankle Hardware

The sensors output analog signals to a middle-level microcontroller\textsuperscript{13} that performs A-D conversion and outputs a serial data stream. Like the first two prototypes, the microcontroller is not actually affixed to the shoe itself. The first prototype used a single Arduino NG developer board with an ATMEGA8 microcontroller. This was strapped to a waist belt (figure 4.5), and wires from the belt were run down around the legs and into sensors on the shoe. The second prototype used two Arduino Mini developer boards with the ATMEGA168 microcontroller, which provided in total more than twice the number of analog inputs. The Minis were small enough to strap on to the ankle along with a 9V battery (figure 4.6). Prototype three still uses the Arduino Mini boards, but uses all analog inputs and a 1.5 battery (AA) instead of the 9V. Though a high-heeled shoe offers space beneath the arch of the foot, the microcontroller circuit is not

\textsuperscript{13} Tom Igoe describes different levels of microcontrollers. High level microcontrollers have many hardware components built into the system, making it easy to plug into other devices without much programming or circuit-building. They also work with easier programming languages such as Max/MSP. Low level microcontrollers require the user to build their own interfaces and program in a language like C. of components already built into http://tigoe.net/pcomp/microcontrollers.html
small enough yet to put there. The ATMEGA168 runs at a clock speed of 16 MHz and has a 10-bit A/D converter.

![Prototype 2 ankle hardware](image)

**Prototype 2: A smaller circuit was strapped onto each ankle.**

### 4.1.3 Communications Protocol

It is crucial for IDMS to provide wireless communicate from the dancer to the computer, since the confines of wires inhibit movement tremendously. Most systems described in Chapter 2 use RF transmitters and receivers with a base station that converts the RF signal into serial data [1, 14, 16]. More recent systems use TCP packets sent over the 802.11b wireless communications protocol [15, 17]. The ATS system uses the Bluetooth wireless communications protocol. The ATMEGA168 sends serial data to the Parani-ESD200, a Bluetooth serial module, which transmits the serial signal to a computer wirelessly. The Parani-ESD200 is designed to replace the RS-232 cable and operates within a 30m range [79]. Any computer with Bluetooth capability can receive the signal.

### 4.2 Software & Mapping

The ATS uses the Max/MSP programming environment to map the signals from each shoe into sound or sound effects parameters. It uses a patch called “Arduino2Max” which reads the Arduino pins via the serial object. Once the pins are read, the user can program gesture recognition algorithms, then map the gestures to any synthesis or sound
control desired. A new Max patch was written for each ATS prototype; prototype one and two included three different modes with different mapping designs.

One example of a mode in Prototype one was the loop trigger mode. A quick, backwards spin on the right foot triggered a two-measure drum loop that repeated until dancer performed either a right-foot gancho (a fast, “hook” kick led by the heel) or another backwards spin. A gancho would stop the loop, and another backwards spin would stop the original loop but immediately start playing a different loop. FSR readings from the ball of the foot would trigger individual piano and violin notes, and in addition, one-to-two mapping techniques were used to add reverb and delays to the violin and piano notes. The mapping strategies for the third prototype are described in the next chapter.

Digidesign’s ProTools, Properllerhead’s Reason, and a graphical interface to Csound called Cecilia are also key components in designing the ATS’s sounds (see section 4.2).

4.2 Sound Design & Choreography

The ATS can potentially work with any kind of sound imaginable, as we can just as easily store a rock-n-roll drum loop in the Max/MSP buffer as an electro-tango drum loop. However, the controller is a high-heeled tango shoe, and its associated sound is intended to fit in the genre of “electro-tango,” a mix of traditional Argentine tango music with electronic sounds, beats, samples and audio processing techniques. Subtractive synthesis was also used in the second prototype for a mode that produced filtered noise. Like the first two, the third prototype implements acoustic violin, piano, bandoneon, and guitar sounds, as well as synthesized sounds and loops. Electronic beats and loops were designed in ProTools, using sounds from the Reason sample library and software synthesis modules. Musical phrases were composed with the acoustic instrument sounds, and then processed using delays, filters, harmonizers, reverbs, time warpers, convolution techniques, and pitch shifters. Many sounds were designed using the DAFx processors in Cecilia, a graphical interface for Csound modules. More effects processors were created in the Max/MSP patch, providing real-time audio processing.
The ATS can potentially work with any style of gestures, however just like electro-tango is the intended musical genre, the ATS is intended to work with tango moves and gestures. A large tango dance vocabulary already exists from which many gestures are extracted from and exploited in the gesture recognition section of the mapping design.

The ATS is truly a system, and each component is necessary for the system to behave as a musical instrument. However, the performer’s expressive capabilities will heavily depend on the mapping design. Since the intention is for the audience to perceive a relationship between the performer’s gestures and the resulting music, the mapping design should enable a cognitive connection between the gestures and sounds.
Chapter 5: Design, Implementation, Results, & Discussion

5.1 Design a Mapping Model

Models are important for designing DMIs [72], as they help define and clarify each layer. I derived a mapping model (figure 5.1) from models described in section 2.2. It is divided into two sections, one for the gesture transducer (controller) and the other for the sound playback\[14\] and processing. These sections should ideally be independent from each other, except for the direct connections between gesture feature extraction parameters and the triggered sound sources. The intention, like in [42] and [71], is to separate the control parameters from the gestural transducer so that any gestural transducer can be used with the sound mapping section without needing to

---

\[14\] I write “sound playback” as the ATS currently does not control synthesis, rather it triggers pre-recorded audio files and controls the effects processors.
change it. In other words, the sound mapping section should stand on its own. This could be beneficial since the ATS is still a prototype, and its hardware will most likely change. Also, it could encourage collaboration: if other musicians wanted to execute the same music with their own controller, modularity would simplify this process, requiring the musician to redesign only the gesture mapping section.

Each section includes a mapping layer that translates physical parameters into perceptual parameters (layer one and three). These layers can include simple and complex parameter relationships. An event from the gesture mapping section will trigger a sound, such as pushing a button on a hand-held controller, or stepping from one foot to the other on a shoe controller. Potentially, an action within any layer can trigger a sound, though it will most likely be an event from the gesture feature extraction layer.

The second mapping layer connects the two sections using perceptual parameters with simple, one-to-one parameter relationships. Perceptual sound parameters will be translated into DAFx control parameters with simple or complex relationships, and these control parameters will be sent to the effects processors. There is an option of directly controlling the DAFx modules via the gesture feature extraction parameters, though this option disables the independence of the sections. Finally, the effects processors will output the modified sound, which may be mixed with the original sound (the audio files can bypass the DAFx modules if no processing is desired). Any number of DAFx modules can be used with any number of controllable parameters.

### 5.2 Implementation

The mapping model can be implemented in numerous ways. For this example, I used the objectives described in section 3.3.2 as a foundation to build the Max patch. First, I created drum loops and composed some music. Next, I established the sound section, and finally, established the gesture section. This section describes how I planned out the mapping design via the objectives, and then goes into implementation details with a Max patch.
5.2.1 Objectives

**Compose**

*Create the musical structure of a song:* I wanted a single performer to be able to “play” accented chords and notes, either over rhythmic drum loops or solo. I also wanted to give them control of the sounds’ musical qualities and control over a reverb module to help manipulate intensity levels. The performer needed enough reverb control to quickly cut off the dry sound and let the reverb tail decrescendo or crescendo, fading out only at their desire.

*Compose/design sound files/loops:* I prepared about thirty samples of notes, chords, and cadences that included an acoustic violin, bandoneon, piano, and guitar. Next, I composed two sets of rhythmic loops: one that had strongly accented beats comprised of noisy, inharmonic sounds from the ‘glitch’ sample library in Reason, and another comprised of softer, more harmonic sounds that were rhythmic but with subtler beats.

**Develop Sound Mapping Section**

*Describe perceptual and qualitative characteristics of the sounds:* This step was purely subjective, using words that would eventually describe the perceptual parameters mapped in layer two. I described some sounds as smooth, playful, sharp, and frisky, and others as dark, serious, and intense.

*Define perceptual and qualitative parameters to control:* One perceptual parameter would be the “mood” of the pre-composed sounds, and the other, “clarity,” which would pertain to reverb parameters.

*Decide what specific effects to apply to the sounds/loops:* I only used one DAFx module in this example: *Gigaverb*, a Max object developed by J. Sadeharju and O. Matthes. It outputs stereo reverb from a mono sound file, with over seven controllable parameters.
Relate perceptual parameters with quantitative parameters: The audio samples and loops will be grouped into multiple sound banks, with each bank representing a particular perceptual characteristic (figure 5.4). A simple, linear scale will describe the “mood,” with as many scale numbers as there are sound banks. Ultimately, the number of the mood scale will determine the bank from which the sounds are selected. Another linear scale will represent “clarity,” and these numbers will be mapped (linearly and non-linearly) to reverb parameters including reverb time (sec), wet/dry amount (dB), room size (meters), and tail reflections (dB).

Choreograph

Create general structure of the dance: Musically, this example is based on strong accents, heavy pauses, and busy drum loops. Choreographically, the dance is based on accented movements, heavy pauses, and smaller movements in between the accented ones. The performer will take slower, more fluid steps as time in between steps increases, and faster and sharper steps as time in between steps decreases. Large, exciting gestures such as kicks and spins will decorate the simple stepping. Ideally, the performer should be able to build the music by executing faster steps and more dynamic gestures, with the option of cutting off all sounds except the reverb tail with a jump, and then using a slow, fluid gesture to fade out the reverb tail.

Choreograph educated gestures: This part was less challenging, as a large tango vocabulary already exists [80]. I wanted to keep the gestures on the simple side so that the gesture recognition and feature extraction sections could work consistently (gesture recognition is the weakest part of my mapping design). Thus, I used the most basic tango move as the base of the dance: the walk, or a step. These include straight forwards and backwards walking, and side steps. These are considered static gestures. More static gestures for this example include the boleo, gancho, and a forward kick. Dynamic gestures include forwards and backwards ochos (a step that follows a pivot on one foot);

\[ s(t) = F(e(t)) \]

15 J.M Courtier’s [81] definition of static and dynamic gestures is mathematical: static- (can be compared to viewing a movie frame-by-frame [82]) \( s(t) = F(e(t)) \); dynamic- (a sequence of static gestures [82]) \( s(t) = F(e(t), e(t-1), ...) \)
fast and slow spins with the free leg either out to the side or tucked behind the spinning leg; slow, side, leg extensions; and circles “drawn” on the floor with the toe.

Develop Gesture Mapping Section

Describe perceptual and qualitative characteristics of the movements: This was another purely subjective step, choosing words that would eventually describe the perceptual parameters used in layer two. The gestures I intended to work could be described as: slow, light, languid, and sweeping (fluid); short, quick, and flick-like; sharp, strong, dynamic, and intense.

Relate perceptual and qualitative parameters with quantitative parameters: The overall choreographic style will be quantified with the “mood” scale. The parameters that will determine the mood scale level include amount of motion and the number of sharp, static gestures (fast spins and kicks) during a specific time period (1 – 4 seconds). In order to control the “clarity” of the music, I created a perceptual scale that quantifies the “sharpness” of the gestures, which is determined by both the number of steps and static gestures over a specific time period.

5.2.3 Mapping Design: The Max Patch

The following descriptions follow the mapping model in figure 5.1.

Gesture Recognition

First, I route the scaled and smoothed sensor data input to subpatches that perform gesture recognition. The gesture recognition techniques currently distinguish:

- Steps (forward and backward steps; diagonal steps confuse the patch)
- Kicks (fast, high, forward kick, side kick, gancho, ‘flying’ boleo)
- Spins (fast, slow, forwards, backwards)
- Foot tilt with respect to gravity (from accelerometer)
- Acceleration of legs/feet

Gesture Feature Extraction

I obtain the following parameters from the gestures above:
- Time between steps (ms)
- Number of kicks per unit of time
- Number of fast spins per unit of time
- Estimated amount of motion
- Overall speed of motion per unit time (fast/slow)

**Mapping Layer 1: Calculate perceptual gesture parameters**

The gesture feature parameters are used in mathematical expressions to obtain perceptual gesture parameters for the mood scale and sharpness scale.

**Mood Scale**

The mood scale is an example of a many-to-one mapping. You can see the actual mapping in the expression in figure 5.2. There are four gesture feature parameters (\(f_1 - f_4\)) and a variable the user sets in advance (the ‘no. of peaks’ multiplier, \(f_5\)).

First, the amount of motion is estimated by adding the values of each axis of the tri-axis accelerometers (zero is at rest; positive and negative acceleration is all greater than zero). Higher numbers represent more motion. These numbers are added to each other with the ‘accum’ object (the first box under “gesture feature extraction” in figure 5.3), and the total sent to \(f_1\) and \(f_3\) in the mood scale expression (figure 5.2). Sharp peaks are found by analyzing the tri-axis accelerometer values. If there is a change in direction past a threshold, a value captured after a user-defined amount of time after the peak (75ms in figure 5.3) determines whether this peak represents a quick, sharp gesture such as a kick. The number of peaks per unit time are sent to the mood scale expression (\(f_2\) and \(f_4\)). The ‘no. of peaks’ multiplier determines how much influence the number of peaks has on the mood scale value.
Using this many-to-one mapping design, the performer controls the sample bank selection mainly by how much she is moving her legs and feet. The specific gestures are arbitrary, however, she can quickly change the mood scale value by throwing in gestures that will cause peaks. Keep in mind that there is a delay from when the mood scale value is being calculated to when it actually selects the sample banks. I have found that four seconds is the largest time window that allows performers to still feel in control.

**Sharpness scale**

The sharpness scale is another example of many-to-one mapping. You can see the actual mapping in the expression in figure 5.4 (top left box). Three parameters are used to calculate the sharpness scale values: the time ($i1$), the number of *ganchos* and *boleos* ($i2$), and the number of fast spins ($i3$) between each step. The user can adjust how much influence the *ganchos*, *boleos*, and spins have on the sharpness value by adding a multiplier. In this example, spins will have a greater effect on the sharpness
value than kicks ($i_3$ is multiplied by 1.5). Generally, the slower and more fluid the movement, the lower the sharpness scale value. Summing up the perceptual gesture parameters, we have:

*Mood* scale parameters: smooth – frisky – strong – intense. Smooth movements will have less motion and fewer kicks and spins, where as intense movements will have many kicks, steps, and more movement in general.

*Sharpness* scale parameters: fluid – sharp – very sharp. The more steps (walking), *ganchos*, and fast spins, the higher the sharpness scale values. The more fluid the movement, the lower the sharpness scale value.

---

**Fig. 5.4** Three-layered mapping strategy to relate the “sharpness” of the gestures to “clarity” of the sound.
Mapping Strategies for the Augmented Tango Shoe

Mapping Layer 2: Calculate perceptual sound parameters

This is the simplest mapping layer, consisting of one-to-one mappings. We have, so far, values from the mood scale and sharpness scale. In this particular example, the “mood” parameter represents the mood of the dance and the music; perceptual qualities of movement and sound are shared (see table 3.1 for examples). No calculations are necessary to obtain the mood scale for sound (though there could be if the user wanted to use the mood scale values to control another DAFx).

The sharpness scale values are scaled non-linearly to “clarity” scale values (see figure 5.4, “perceptual sound parameters” box on the right side). Summing up perceptual sound parameters, we have:

Mood scale parameters: smooth – frisky – strong – intense. Smooth sounds will trigger sustained chords with longer attacks, whereas intense sounds will trigger accented, orchestral chords and cadences.

Clarity scale parameters: fluid – sharp – very sharp. Low values represent blurry sounds (lots of reverb, longer reverb time etc), and high values represent clear sounds.

Mapping Layer 3: Calculate sound control (including DAFx) parameters

In layer three I apply the two perceptual sound parameter values to one-to-many mappings.

Mood Scale

A one-to-two mapping is used, where the mood scale values will determine which two sample banks the performer will trigger sounds from. In this example, there are four mood scale values and eight sample banks.

Clarity Scale

A one-to-many mapping is used where the clarity scale value is mapped via the ‘scale’ object to four parameters that control the ‘gigaverb’ object: reverb time, room size, wet/dry amount, and reverb tail.
Direct Control

The part that requires direct control from gesture feature parameters to sound control is actually a crucial part of this mapping example. The sounds are triggered each time the performer takes a step, and also when she performs a very specific gesture (in this case, a slow spin on the right foot with the left foot extended behind the right triggers a slow, melodic loop, a fast backwards spin on the right foot with the left foot wrapped in front triggers a faster drum loop, and a sharp, fast, striking motion where the left leg juts out to the side and stops turns off any loop). Essentially, every time the dancer takes a step, which is essentially a shift of weight over the feet, she “plays” a corresponding note, chord, or cadence.

When the performed executes one of these gestures, a bang travels to one of the ‘route’ objects (top of fig 5.5), and the mood scale value determines where it is sent from there.

Fig. 5.5 Mapping layer 3: map mood scale values to sound bank selection.
5.3 Results

My goal with the new mapping design was for the music to directly reflect the dance, and in that case the project was a success. Though not yet performed in front of a live audience, I watched videoed practices of the ATS. It felt successful for a three – four minute piece, however it was difficult to tell how much of the “success” was due to the mapping design, the dance, or the music. When I performed slower, more languid gestures the music would be less rhythmic, smoother, and with long, reverberant decays. At a moment of long decays into silence, I could perform a strong, high kick that would start a loud, surprising drum loop.

5.4 Discussion

It is important to re-iterate that this mapping design is only one example of how you can implement the mapping model in figure 5.1. Also important to note is the difference between mapping for a sound-specific controllers and more general controllers. For example, the Yamaha WX7 wind controller mentioned in [52, 74] has an interface similar to that of a saxophone, and though could potentially work with a variety of mapping designs, is used mainly with the mapping design that comes with it: one that emulates a real saxophone. Key layout is the same as a saxophone (with additional keys to extend the octave range), so a finger pattern on the WX7 will produce the same notes if executed on a real saxophone. On the other hand, more general controllers such as the Radio Baton [83] typically do not have a pre-conceived associated sound or mapping design. Musicians who use these controllers usually must decide on the sound they want to control as well as the mapping strategy to use.

The motivation for this thesis was to design a mapping strategy for the ATS. Though I was hoping to use the mapping strategy permanently and afterwards focus just on composition and choreography, I realized that only one mapping design will severely limit performances possibilities with the ATS. It functions as a general music controller, and I believe that each new song or piece performed with the ATS will most likely need a

---

16 Other commonly used terms are “alternate controllers” or “novel controllers,” which both refer to gesture transducers used to control sound, light, video, even robots.
new mapping design that conforms to the performer’s dancing abilities and to the music composer’s desires. In fact, there have been seven different mapping designs over the course of the ATS’s development, and I now know there will be many more in the future.

Constantly needing new mapping designs can a negative because either collaboration is required, or else a great deal of work descends upon the person who becomes music composer, choreographer, and mapping designer. Even within mapping design, many people focus solely on gesture recognition techniques and others on taking those “recognized” gestures and mapping them to sound synthesis; so for one person to take on all the demands of an IMDS, the final results may not reach its potential. However, this can also be a positive, as if the mapping design does in fact employ independent layers, once experts in each area refine their work (i.e. engineers perfect a controller, computer musicians create a synthesis algorithm, mapping designers implement a mapping strategy), people could “mix and match” each component with confidence that each part will behave with stability and reliability. Instead of a tangible instrument having a signature sound, like the piano having an associated sound that the hammers make when they strike the strings, a signature sound could exist independently, and a performer can adopt that sound with their general controller for one song, and with push of a button switch to another sound. Or they could keep the same sound, but change the mapping design in an instant, or both. This possibility offers a great deal of potential to a performer, but they may risk trading virtuosity for unlimited combinations of components.

One interesting thought, but a topic for a separate discussion, is whether this component exchange would encourage the democratization of music performance. With one, forty-dollar Nintendo Wii controller [84], a computer, and a couple free downloads from the internet, you could control sound and maybe legitimately call yourself a musician. The implications of these possibilities are fascinating, and only time will tell how it unfolds.
Chapter 6: Conclusions and Future Work

Composing a Performance

The goal of this work was to focus specifically on mapping strategies for the ATS system. I found, however, that completely separating the mapping design from the other components (hardware, gesture recognition, etc.) was very difficult, if not impossible, mainly because none of the other components were perfected or reliable. In essence, what I presented in this thesis was an example of *composing a performance*, or the combination of composing music, choreographing dance, and designing mapping strategies. Since the ATS is still in its prototyping stage, this turned out to be a positive path, as each component of the ATS did improve. However, in order to narrow the focus of my work to mapping strategies, a significant amount of preliminary work is necessary.

Concerning the ATS specifically, future work would begin by improving the controller’s hardware. A printed circuit board would help prevent poor soldering jobs and loose wires. Embedding the accelerometers and gyroscopes inside the shoe rather than attached to the outside would increase robustness. Decreasing the circuit size in general so that it could fit underneath a high-heeled shoe would be less inhibiting for the dancer, as now the circuit fits around the ankle.

The next stage of future development would be improving the gesture recognition techniques. Employing advanced computational techniques like neural networks and the Hidden Markov Model (HMM) could help distinguish between gestures with only subtle differences. Just improving the explicit techniques I have already implemented could improve reliability, as currently, unpredictable gestures sometimes trigger sounds that I do not intend to trigger.

Finally, Argentine tango is a partner dance full of intertwining legs and feet, and so the most effective performance scenario would include two dancers with two pairs of augmented tango shoes. In addition to accelerometers, gyroscopes, and FSRs, these shoes would have distance sensors (i.e. ultrasonic sensors) so the mapping designer could exploit the relationship between the dancing couple.
Adhere to a Set of Objectives

Adhering to a set of objectives helped execute the mapping implementation efficiently. Without the objectives, it was easy to get carried away with endless mapping possibilities; suddenly realizing you have side-stepped your original goals.

Perceptual Parameters

Since I believe that live performers should aim, at least partly, to emotionally connect with their audience, mapping strategies should take the relationship between sound and movement into consideration by employing perceptual parameters. More research in embodied music cognition and neuroscience can offer a deeper understanding of these relationships.

Other Recommendations for Future Work

Other general recommendations for future work in mapping strategies for IDMSs include:

- Continued exploration into mapping strategies for “performing” DAFx with gesture transducers.
- Applying mapping strategies that employ sonic feedback to dance and/or athletic training.
- Collaboration between the medical field and the entertainment industry (for example, advanced gait tracking techniques have been used in the medical field for quite some time, but the technology has yet to be applied to the interactive computer music field).

Ultimately, a mapping designer for an IDMS could function as the liaison between a professional dancer or choreographer, and music composer. Though there is a plethora of work to be tackled in the interactive multimedia field, it is a relatively new performance genre; where traditional dance and music performance has been around for thousands of years. I find it fascinating to experience the connection between human and machine during a live performance, and since mapping design is the key to exploiting this connection, more development in this field will result in performance experiences not yet possible in today’s highly technological world.
Appendix A: Acronyms

ATS       The *Augmented Tango Shoe* refers to a DMI comprised of a gestural controller built into high-heeled shoes that trigger sounds and controls digital audio effects with software written in Max/MSP. This thesis’s goals are to improve the mapping design specifically for the ATS.

DAFx      *Digital audio effects*, but also the name of a research project with an international conference held between 1997 and 2001.

DIEM      The *Danish Institute of Electroacoustic Music*.

DMI       *Digital music instrument* refers to a system that includes a gestural controller, software, and computerized sound. The term has been adopted from Wanderley and Miranda’s book *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard* [50].

FSR       *Force sensing resistor*. These resitors change value depending on the amount of force applied on it.

IDMS      *Interactive dance/music system* refers to a system where dancers’ gestures and movements control music through a computer interface such as gestural controllers or video cameras.

SICIB      *Sistema Interactivo de Composición e Improvisación para Bailarines* (interactive system of composition and improvisation for dancers).

STEIM      The *Studio for Electro-Instrumental Music*, based out of Amsterdam.
Bibliography


http://www.yamaha.co.jp/design/products/1990/miburi/


http://www.andrewjs.com/performoshoes.html


Mapping Strategies for the Augmented Tango Shoe


Mapping Strategies for the Augmented Tango Shoe


http://www.tekscan.com/medical/system-mobile.html


http://www.langdoncrawford.com/videos.htm


http://www.media.mit.edu/hyperins/


Mapping Strategies for the Augmented Tango Shoe


http://www.emfinstitute.emf.org/exhibits/radiobaton.html

http://www.amazon.com/Nintendo-Wii-Remote-Controller/dp/B000IMWK2G/ref=pd_sim_vg_title_1