A Gestural Media Framework: Tools for Expressive Gesture Recognition and Mapping in Rehearsal and Performance

by
Elena Naomi Jessop

Bachelor of Arts in Computer Science and Theater & Dance
Amherst College, 2008

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author____

Elena Naomi Jessop
Program in Media Arts and Sciences
May 7, 2010

Certified by____

Tod Machover
Professor of Music and Media
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by____

Pattie Maes
Associate Professor of Media Technology
Associate Academic Head
Program in Media Arts and Sciences

A Gestural Media Framework: Tools for Expressive Gesture Recognition and Mapping in Rehearsal and Performance

by

Elena Naomi Jessop

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
on May 7, 2010 in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Media Arts and Sciences

Abstract

As human movement is an incredibly rich mode of communication and expression, performance artists working with digital media often use performers' movement and gestures to control and shape that digital media as part of a theatrical, choreographic, or musical performance. In my own work, I have found that strong, semantically-meaningful mappings between gesture and sound or visuals are necessary to create compelling performance interactions. However, the existing systems for developing mappings between incoming data streams and output media have extremely low-level concepts of "gesture." The actual programming process focuses on low-level sensor data, such as the voltage values of a particular sensor, which limits the user in his or her thinking process, requires users to have significant programming experience, and loses the expressive, meaningful, and metaphor-rich content of the movement. To remedy these difficulties, I have created a new framework and development environment for gestural control of media in rehearsal and performance, allowing users to create clear and intuitive mappings in a simple and flexible manner by using high-level descriptions of gestures and of gestural qualities. This approach, the Gestural Media Framework, recognizes continuous gesture and translates Laban Effort Notation into the realm of technological gesture analysis, allowing for the abstraction and encapsulation of sensor data into movement descriptions. As part of the evaluation of this system, I choreographed four performance pieces that use this system throughout the performance and rehearsal process to map dancers' movements to manipulation of sound and visual elements. This work has been supported by the MIT Media Laboratory.

Thesis Supervisor: Tod Machover
Title: Professor of Music and Media

A Gestural Media Framework: Tools for Expressive Gesture Recognition and Mapping in Rehearsal and Performance

by

Elena Naomi Jessop

The following person served as a reader for this thesis:

Thesis Reader:____

Joseph Paradiso
Associate Professor of Media Arts & Sciences
MIT Media Laboratory

A Gestural Media Framework: Tools for Expressive Gesture Recognition and Mapping in Rehearsal and Performance

by

Elena Naomi Jessop

The following person served as a reader for this thesis:

Thesis Reader:____

Leah Buechley
Assistant Professor of Media Arts & Sciences
MIT Media Laboratory

Acknowledgements

TOD MACHOVER
Thank you for your continual guidance, inspiration, and support over the past two years as my advisor. I have had an amazing experience as part of the Opera of the Future group, and I look forward to continuing to learn from and work with you.

LEAH BUECHLEY, JOE PARADISO
Thank you for being part of my thesis committee, and for your invaluable feedback and support during this process. I've learned quite a lot from you.

KEVIN BURCHBY, NOAH JESSOP, DANBEE KIM, LISA M. SMITH, XIAO XIAO
Thank you for realizing *Four Asynchronicities*. Your talent and energy made this performance happen.

PETER TORPEY
Thank you for your boundless collaboration, encouragement, and friendship.

ADAM BOULANGER, ANDY CAVATORTA, WEI DONG, NOAH FEEHAN, BOB HSIUNG
Thank you for a terrific two years of challenge, inspiration, hard work, and fun.

TAYA LEARY
Thank you for your support in so many ways.

MEDIA LAB FACULTY, STAFF, STUDENTS
Thank you for your constant inspiration, excitement, and feedback, and for making this place exactly where I want to be.

ELIZ, JAMES, ROSS, DONA
Thank you for your long-distance friendship, care, and constant enthusiasm.

MOM, DAD, NOAH, KEVIN
For your unflagging love and support, thank you. Always.

Table of Contents

Illustration Index

Chapter 1: Introduction and Motivation

Human movement is one of the most powerful tools available to performing artists for creating and communicating emotions, moods, and experiences. The body, its gestures, and its ways of moving are rich in communication, metaphor, and expressivity. Particularly when we see a person on a stage, every turn of his head, reach of his arm, and step of his feet is full of expressive content. The way a pianist brings his hands down on the keys or the way a dancer curves his body through space sends a tremendous amount of meaningful, metaphorical, and emotional information to the audience. In the context of theatrical, choreographic, or musical performances that incorporate digital media, artists have taken advantage of technology to augment the body's expressivity, using human movement and gestures can be used to drive, control, and shape that media. For these interactions between a performer's body and digital movement to be compelling, the relationships and connections between movement and media should be expressive and the performer's agency should be clear. In my own performance work, I have found that strong, semantically-meaningful mappings between gesture and sound or visuals help create compelling performance interactions, especially when there is no tangible instrument for a performer to manipulate, as is the case in a dance performance. Too often one sees a performer making complex movement and hears complex sounds, but cannot tell if the movement is causing changes to the sound or if the sound is shaping the way the performer is moving. When a specific gesture and the way it shapes a sound seem meaningfully related, especially because of common, context-driven, or metaphorical associations with that gesture, it is much easier to accept that the performance of that gesture actually shapes the media in the performance experience.

A number of performance-makers have explored the process of making connections between movement and digital media, and some tools have been developed to assist users in this mapping process. However, those existing mapping tools are fundamentally lacking; they may help the users form connections between movement and media at a technical level, but they rarely support performance-makers' needs to develop mappings at a high level of artistry. This limitation occurs because preexisting systems for developing connections and mappings between incoming data streams and output media only incorporate low-level and constrained concepts of "gesture." Instead of promoting mappings that incorporate semantically meaningful gestures or qualities of movement, the programming process typically focuses on low-level input data, such as the voltage values of a particular sensor or colored blocks of a video image. Working with computations on such low-level sensor data is not conducive to an artist's goal of flexibly developing compelling material during the rehearsal process of a performance piece, nor does this process support an artist's desire to think at a creative and symbolic level about the associations between movement and media. Much of the expressive content of a performer's movement is lost when we only focus on sensor readings rather than meaningful gestures or qualities of gesture. Additionally, to implement these interactive systems, the artist is forced to be a fairly skilled programmer and to think about intersections of gesture and media as a programmer would.

I came across some of these issues while developing expressive gestural performance instruments in the Opera of the Future group. One instrument is the Vocal Augmentation and Manipulation Prosthesis (VAMP), a gesture-based wearable controller for live-time vocal performance [35]. This glove-shaped controller allows a singer to capture and manipulate single notes that he sings, using a gestural vocabulary developed from that of choral conducting. In the development process for VAMP, I began by envisioning a desired set of mappings of gestures to sound manipulations, selecting sensors for the glove that could detect the necessary gestural vocabulary, and then hard-coding the gestural recognition and mappings. I found that strong semantic and metaphorical mappings, such as pinching fingers together by the mouth to "capture" and hold a note, were crucial in making the glove a compelling performance instrument. In particular, the association of the grabbing gesture to "holding" a note created a metaphorical environment where the voice suddenly became tangible, sculptable, and located at a specific point in space. In the programming process, however, this rich gesture is identified prosaically as a set of threshold values on the fingertip pressure sensor, a definition of gesture not conducive to creative thought and mapping exploration. Due to the low-level nature of the gesture recognition and mapping implementation, it was challenging to later reconfigure or expand the mappings created with this glove, or to experiment with those mappings in a rehearsal setting. A higher-level gesture mapping system was necessary to address these issues.

For this thesis, I have created a new approach and development environment for gestural control of media in rehearsal and performance, a Gestural Media Framework. This approach serves as a toolkit for users who may have little experience with programming, allowing them to create clear and intuitive mappings in a simple and flexible manner by using high-level descriptions of gestures and of gestural qualities. Numerous mapping systems for digital media have been previously developed (such as Max/MSP[19] and Isadora[76]), but these systems do not incorporate gesture representations, or even much conception of gesture. My system centers on the ability to work with a vocabulary of abstracted and encapsulated gesture and gesture quality objects, allowing for higher-level control and creation of mappings between movement and media. Such ease in creating mappings is necessary when exploring relationships between gesture and media during an artist's rehearsal process, rather than as a thought experiment before the rehearsal process begins. As a choreographer developing movement on performers throughout a rehearsal process, I am aware of the need for flexible systems and modes of thought about gesture recognition that could be easily integrated into rehearsal. The Gestural Media Framework draws from related work on gesture recognition in the field of HCI, as well as prior work on expressive qualities of gestures for music, theater, and dance performance, especially Laban's Effort theory for describing qualities of motion. I additionally bring to this project my experience with cross-disciplinary work that combines technology and performance, as well as my background in a variety of areas including computer science, choreography, choral conducting, and theatrical design.

As part of the evaluation of the Gestural Media Framework, I choreographed a piece for public performance that used this system to map dancers' movements to control sound and visual elements, including music, video projection, and stage lighting. This performance piece, titled *Four Asynchronicities on the Theme of Contact,* consisted of four separate but interconnected movements, each of which explored different relationships between the performers' motion and media elements. I incorporated the use of my system throughout the rehearsal process for this performance piece, creating and exploring interactions between performers and media at the same time as I was developing movement on the performers and discovering the story of each piece. I thus examine in this document to what level this system satisfies necessary requirements for the choreographic process, for use in rehearsal situations, and for creating compelling interactions. While I speak of choreography and dance here, the Gestural Media Framework is applicable to a wide variety of performance forms.

In this document, I will first review prior work in some related subject areas, such as technology in dance and musical performance, notation systems and movement analysis in dance, and gesture recognition in Human-Computer Interaction, so as to define the field I am working in and the inspirations I am drawing from that field, while also delineating the ways in which the Gestural Media Framework is a significant theoretical and practical step forward. In Chapter 3, I discuss my own prior work with gesture and performance capture technologies in the Opera of the Future group, and how those explorations were relevant and inspirational in the development of this thesis work. In Chapter 4, I describe the development and structure of the framework design and hardware/software implementation of the Gestural Media Framework, as well as the artistic and practical requirements demanded of this system. In Chapter 5, I discuss the rehearsal process and the four pieces I choreographed for *Four Asynchronicities on the Theme of* Contact, and evaluate the use of the system in this particular performance context. Finally, in Chapter 6, I examine the Gestural Media Framework as a whole, review the successes and challenges of the system, and look ahead to how this technology and theory might be applicable in a variety of contexts.

We are now in an era in the intersection of technology and performance where the technology no longer needs to be the primary focus of a piece. The performance is not about the use of a particular technology; instead, the performance has its own content that is supported and explored through the use of that technology. This is a particularly exciting time to be doing work at this junction, when many sensing, visualization, and networking technologies have already been developed for performance and when the primary question is how to create compelling performance experiences that draw from and are greatly enhanced by, but do not center on, these technologies. In particular, technologies that help enhance and expand a performer's expressive physical gestures take advantage of a key aspect of live performance art: the capability for change from performance to performance, for variation in the performer's moment-to-moment expressiveness. Technology is best integrated into performance when it can support this variety and liveness, instead of fighting against it with pre-determined, pre-timed events. Thus, I aim to create ways that technological media elements can be intimately linked to the expressivity and nuance of a performer's live movement.

There are currently many ways that aspects of a performer's movement can be detected technically, from on-the-body sensors to computer vision systems, to capacitive field sensing. There are also increasingly many tools for gesture recognition, at least in the context of human-computer interaction. But how do we make sense of a performer's movement, and how do we explore the significance of those gestures? How do we make specific movements and qualities of movement augment a performance experience by how they affect other media in the performance? How can we make even greater use of the wide affective and expressive channel of a performer's body, physical presence, and gesture? How can we effectively and captivatingly extend the power of the human body into the realm of digital media? How can we create tools that encourage metaphorical, meaningful, and rich associations between movement and media, rather than naïve and linear mappings? With these questions at the forefront, this thesis presents my work developing tools and methodologies that make it easier to step away from the details of the gesture and movement analysis technology to design compelling experiences.

Chapter 2: Background and Related Work

Before addressing the questions posed in the previous section, it is important to first put my work in context, as there are several areas that have impact on the direction of new expressive gesture recognition technologies for performance. The first of these areas is the prior use of technology in performance work, particularly in dance performance and musical performance. This is a large field and is quite varied, so I will confine the scope of this review primarily to previous work where the expressiveness of a live performer, particularly that performer's movement, directly or indirectly shapes the form of the technology in the performance. While I organize these performances into the loose categories of theater, dance, music, and opera, the boundaries of these performance areas are quite fluid and there is significant overlap. The second significantly related area of study is the history and analysis of movement in dance performance, examined from a less technologically-centered perspective through such methods as traditional dance notation systems. The third relevant and influential area of study is the use of gesture recognition in the field of Human-Computer Interaction, and the technical implementation of gesture and movement recognition technologies. Following the brief review of these areas, I will address how my methodology and system connect to and draw from, but are significantly unique from, this prior work.

It is also necessary to quickly define what I mean by "gesture" and by "quality of movement," as these terms are quite significant in the background review and in my own work. By "gesture," I mean a specific complete motion of a performer's body, with particular body parts moving through space over a period of time. Frequently, these movements carry semantic, emotional, and/or communicative content. This is distinct from a "pose," a static position of the body. By "quality of movement," I mean the dynamic content of the movement, the manner in which a movement is performed, as distinct from the specific changes in body shape and position that occur during that movement.

Additionally, for purposes of this discussion, it is necessary to define what I mean by a "mapping system." In this thesis, a mapping system is a software framework that allows a user to define functional relationships between a set of input parameters and a set of output parameters: output = f(input). Most of the mapping systems we will discuss have input parameters such as sensor readings, MIDI values, Open Sound Control messages, etc., and allow the user to create relationships between these inputs and control parameters for some media outputs.

2.1: Technology and Performance

As computer-based technology has become a major part of living in the world today, it is not as unusual as it might seem to combine the fields of technology and performance. In fact, theater and performance artists have often been interested in exploring cutting edge technology. Steve Dixon says in *Digital Performance*,

> Digital performance is an extension of a continuing history of the adoption and adaptation of technologies to increase performance and visual art's aesthetic effect and sense of spectacle, its emotional and sensorial impact, its play of meanings and symbolic associations, and its intellectual power."[21, pp 10]

In fact, the development of technology and the exploration of that technology in the theater have often gone hand-in-hand. For example, early theatrical experiments in the early 20th century with the new technology of electric lighting popularized lighting technology and resulted in the widespread creation of power grids across America [21]. Other new technologies, from the Internet to capacitive sensing to digital video, have been quickly explored for their potential impact on performance experiences. The range of performances that incorporate technology is therefore quite wide and varied. I will primarily limit the range of discussion to those technologically-enhanced performances where the technology depends upon and is affected by the behavior of a human performer. This category stands in contrast to the vast majority of "technological" performances where the technology takes the form of video projections or computer-generated sound that may share the space with live performers, but remains essentially disconnected from those performers.

2.1.1: Dance and Theater Augmentation Through Technology

Illustration 1: Loie Fuller in one of her specialized dance costumes. (Photo: Frederick Glasier)

Technology has frequently been a component of modern dance performances, even in the earliest foundations of the form. For example, the modern dancer Loie Fuller was an early adopter of technological performance techniques. For her solo dance performances, she would wear long, flowing costumes (of a design she patented) and used strategically placed electric lights to create a range of visual effects, transforming the shape and movement of her body by the way the costumes were hit by the lighting. She even created an performance piece with a costume that glowed due to the use of radium [47]. Importantly, the effects and experiences that Fuller created were dependent not simply on the technological elements, but on the interplay between those elements and her dance movement.

One of the founders of modern dance, Merce Cunningham, was an early adopter of technology in dance, and put technological components into many of the dance pieces he choreographed throughout his long career. As early as 1965, Cunningham's *Variations V* incorporated photoelectric sensors and antennae to mark the positions of dancers; the data gathered by these sensors and antennae then controlled electronic musical devices [47]. This fit into Cunningham's aesthetic of creating a dance and then putting it in the same space with music, rather than the procedure typical in his time of creating a dance *to* a specific piece of music. Additionally, Cunningham incorporated electronic music into his work, typically through his collaborations with the composer John Cage. Cunningham also brought computer technology into his work as part of both the structure and the content of his pieces. Between 1991 and his death in 2009, Cunningham choreographed all of his dances with the help of a computer program called DanceForms, which allows a choreographer to record and manipulate sequences of movement in a three-dimensional computer environment [42]. Since the computer program allows users to quickly reassemble and restructure sequences of movement, it provided Cunningham with tools suited to his aleatoric style of choreography. Additionally, using advanced animation and motion-capture software, Cunningham could digitally record movement sequences performed by a live dancer and then manipulate that movement on the computer. This technology was used perhaps most notably in Cunningham's 1999 work *Biped*, a collaboration with Paul Kaiser and Shelley Eshkar of the Open-Ended Group. Kaiser and Eshkar recorded movement sequences choreographed by Cunningham, then transformed those sequences into animated three-dimensional figures that performed on scrims along with live dancers [21]. In *Biped* the images were not directly affected by the live performers, but instead served as a counterpoint.

Illustration 2: Cunningham's performance piece, Biped.

Visuals by the Open-Ended Group (Photo by Stephanie Berger)

Many other performance artists incorporate movement-capture technology into their dance pieces, using a variety of sensors to track a performer's movement and thus control such elements as sound, video, lights, and costumes. The work of the dance company Troika Ranch is dedicated to integrating technology into dance performance, including multimedia and movement sensing technology. Mark Coniglio and Dawn Stoppiello, the creative directors of Troika Ranch, developed the mapping software Isadora (discussed later in this chapter) to make it easier to control live video mixes and effects in performance. Frequently, the input to the Isadora system includes information about the dancer's movements, detected by bend sensors on the performers' bodies or external sensing systems. Other performance works done by Troika Ranch have involved movement sensors such as laser beams crisscrossing the stage and impact sensors on the floor [71]. In their piece "16 [R]evolutions," Troika Ranch also worked with the EyesWeb computer vision system, developed by Camurri et al. and discussed later in this chapter, using the system to track the trajectories of sixteen points on a dancer's body and use aspects of those points to shape visual and sonic elements of the piece in Isadora [75]. While the resulting performance work does have media elements that are obviously related to the performer's movement, the resulting interactions appear to be simply paired with the performer's position in space and amount of movement, without the impression of the performer having rich, instrumental-like control over the media.

Illustration 3: Troika Ranch's "16 [R]evolutions."

(Photo: Troika Ranch)

Yamaha's Miburi system [77], Paradiso and Aylward's Sensemble [4], and the Danish Institute of Electronic Music's Digital Dance Interface [65] are other wearable sensor systems for movement tracking in performance, all of which have been used for the real-time generation and adaptation of music to accompany performers onstage. One thing that these sensor-based performance systems have in common is their focus on having technology that reacts in real time to the specific details of an individual performance, rather than being programmed to run particular sequences identically every performance. However, all of these systems limit in the kinds of associations that can be made between movement and performance media due to the descriptions of the movement inherent in the systems, such as the amount of bend in particular joints (the Miburi system) or the amount of activity detected among a number of moving performers (Sensemble).

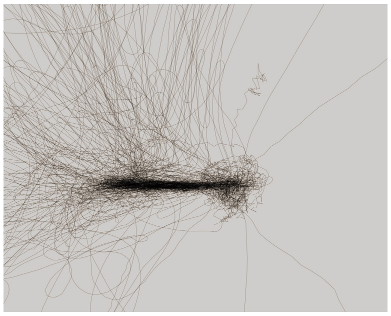


Illustration 4: Screenshot from Levin's audiovisual environment "Floccus"

(Image by Golan Levin).

Other performance systems that are driven by movement track the performer's gestural touch on a surface. For example, Golan Levin created a series of systems that allowed a performer to simultaneously shape synthetic sound and abstracted animation through gestures with a pen, mouse or other pointing device [41]. Some of these systems drew shapes inspired by the user's gestures, while others had animations that were shaped by the movement of the cursor. This gestural input, primarily defined as the movement of the cursor through a two-dimensional space over time, along with the resulting animations, were simultaneously sonified in a variety of ways, creating an audiovisual performance interface. While Levin's model is limited to mouse gestures, the idea of creating models for intuitive gestural control of multimedia elements remains very relevant to my work in the development of the Gestural Media Framework.

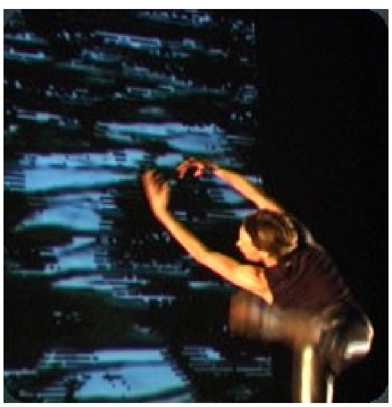


Illustration 5: A piece by Palindrome using the EyeCon camera tracking system.

(Photo by Palindrome).

Camera systems for tracking motion are also particularly popular in interactive dance and performance. *Falling Up*, a performance piece by Todd Winkler, uses one such camera system, the Very Nervous System designed by David Rokeby. In this performance, live video is processed to determine the location and speed of an onstage performer; these data streams are then mapped to manipulations of the sound and the live-captured, projected image of the performer [83]. The Very Nervous System has also been used by Rokeby in a variety of installation scenarios, where different areas of the camera screen is mapped to different instrumental controls. A user's activity and movement in those areas shapes aspects of a music-generating program [59]. Stichting Eleckro-Instrumentale Muzeik (STEIM) has developed another camera-based performer tracking system called BigEye [70], often used for performances where performers trigger sound or music events by moving into particular areas of the stage [64]. The German dance company Palindrome uses their own camera-tracking system EyeCon to detect contact between dancers or differences in the amount that dancers are moving and use that information to shape musical phrases [21].

Most of these camera systems for interactive performance do not have any knowledge of human movement and do not attempt to model the performer's body digitally, instead examining movement as activity or changes in particular pixels or regions of the input. Thus, most associations that can be drawn between the input movement data and output media rely on using particular spacial regions as triggers for events or processes, or on using the amount of change in different areas of the camera input to shape those processes. In contrast, Antonio Camurri and his collaborators in the EyesWeb project attempt to create higher levels of abstraction when using computer vision to examine movement [13]. The EyesWeb system has some knowledge of the human body and the body's capabilities of movement, and uses artificially intelligent agents to process that movement and draw meaning from it. Additionally, EyesWeb stands apart from other systems in that it attempts to integrate some notion of expressive movement, calculating twelve "quality of movement" parameters: "Quantity of Motion (Motor Activation) computed on overall body movement and on translational movement only, Impulsiveness, vertical and horizontal components of velocity of peripheral upper parts of the body, speed of the barycentre, variation of the Contraction Index, Space Occupation Area, Directness Index, Space Allure, Amount of Periodic Movement, Symmetry Index" [15]. Camurri et al. have also explored the concept of KANSEI (emotional) information in dance, developing vector descriptions of emotional dance information inspired by Rudolf Laban's studies of dance movement through space [14]. While the EyesWeb system takes an important step forward in descriptions of movement for mapping purposes, it still has no gesture recognition capabilities and fairly low-level quality measurements.

Flavia Sparacino and her collaborators at the MIT Media Lab have also done extensive work with the augmentation of dance and theater performances by video-based examination of the performers' movement [67]. In particular, they created DanceSpace, an "interactive stage" that incorporates computer vision and image processing systems to track and recognize the body, motion, and some specific gestures of a performer who enters the "stage" space. DanceSpace allows the creators of a performance work to connect movements of specific parts of a dancer's body (such as the hands, head, feet, and torso), to the control of different musical instruments, with volume or pitch of an instrument tied to the spatial location of the associated body part. The DanceSpace system also could incorporate visual elements generated from movement, such as lines drawn in a projected space following the movement of the performer's limbs.

Additionally, Sparacino, in her work with interactive technologies for the theater, developed the concept of Media Actors, software agents that could recognize to the gestures of a live performer and react appropriately and expressively through media objects such as text, video, and images [68]. These programmed agents have their own intentions and behaviors, take in sensory input from the outside world (such as movement data gathered by computer vision systems, or audio information about the performer's voice), and then react to a combination of the performer's behavior inferred from these data sets and the agents' own internal motivations. In this situation, there are no direct mappings between a live performer's movements and the resulting media, since the media changes are driven by the individually-acting, non-scripted software agent.

Artificially intelligent software agents that take in data from a live performance but have their own methods and motivations for reacting to what they perceive is also at the core of Marc Downie's work at the Media Lab [22]. Rather than focusing on issues of mapping, examining what function of input movement data would produce interesting results in the output media, Downie proposes an entirely different approach to the intersection of live performance and digital media. His interactive agent paradigm uses biologically-inspired, artificial intelligences to algorithmically generate visual or sonic elements of a performance; these agents perceive live

Illustration 6: Trisha Brown's dance "How long..." with artificially intelligent software controlling projected imagery.

(Photo: Pat Shannahan for the New York Times).

performance information, but have their own autonomous goals and behavior (which can be shaped, but not precisely determined, by an artist). Downie and the Open-Ended Group (Downie, Kaiser, and Eshkar) have created a variety of performance works with interactive media controlled by these intelligent software agents, including *Ghostcatching* with Bill T. Jones and *How long does the subject linger on the edge of the volume* with Trisha Brown. The autonomous behavior of Downie and Sparachino's agents avoids naïve interactions between the live performance and the media elements, but does not work for the instrument paradigm, only for the player paradigm. The resulting interactions are generally not reproducible and do not give the performer a sense of control.

2.1.2: Music and Technological Augmentation

As in the field of dance, technology has also been given a significant role in musical performance. From the early days of computer music, artists have been exploring the ways in which digital technology can enhance and expand musical experiences. The increasing popularity of digital music-making, particularly with the rise of computer music, led to an interesting disconnect between music and skilled performer. When a musician plays a traditional musical instrument, we in the audience can see a clear correlation between the movements of the performer and the sound that results. When we see the performer striking piano keys, bowing a violin, beating a drum, we receive significant physical information related to the sound generation process and the expressivity of the performer. By adding computer technology in the mix, the connection between the actions of the performer and the sounds that result is no longer one-to-one. If a performer is standing at a laptop clicking with a mouse, it is not intuitive that this physical movement could be generating and performing the incredibly complex textures that we hear.

This disconnect between performer and digital music performance has been addressed in many ways, and many of these explorations have incorporated some aspect of the movement of a performer or an audience member in the creation of an interactive piece of music. Robert Rowe lays out some interesting classification categories for interactive musical systems in [61], though these categories are not unique to musical performance. In particular, Rowe distinguishes between two paradigms of interactive systems, the instrument paradigm and the player paradigm. In the instrument paradigm, the system serves as an "extended musical instrument," where aspects of a human performance are analyzed and control a rich output that goes beyond the traditional response of an instrument but still feels like a solo performance. This paradigm has been used in models such as Tod Machover's Hyperinstruments, discussed in the next section, where a system observes elements of a live musician's performance and uses those elements to shape its musical behavior in ways that are learnable, repeatable, and perfectible by the performer. In the player paradigm, the system serves as an "artificial player," with its own musical behavior affected to various extents by the human performer. This is the case in interactive performance systems like the work of George Lewis, whose generative music systems observe Lewis's live performance on the trombone, but may or may not use that information in determining what it is going to play for its part in the duet [61].

Focusing more on interactive systems that are shaped by a performer's movement, Marcelo Wanderly characterizes three different modes of physical and gestural interaction with music: digital instruments, sound installations, and dance-music interfaces. These interactions take place with varying levels of intentionality: digital instruments are played by performers specifically to produce music, sound installations are played by people who also serve as the audience members, and in dance-music interfaces dancers do not dance explicitly for the purpose of generating sound, but dance movements are interpreted to generate sound [79].

2.1.2.1: Hyperinstruments

A key question in technologically-enhanced musical performance is how to combine the expression and physicality of a traditional musician with the rich sonic vocabulary accessible through computer music. Tod Machover's musical paradigm of Hyperinstruments, digitally enhanced musical instruments that allow an expert performer additional levels of control and expressivity, creates an expressive instrumental relationship between a live performer and sophisticated digital sound. The goal of the traditional Hyperinstrument is to give virtuoso musicians a way to digitally expand and amplify their musical gestures through sophisticated extensions of their traditional playing technique. These amplifications in the digital realm are is still significantly shaped by an instrumental model: the computerized aspects of the music are deterministically controllable by the performer, and the performance behavior necessary to generate these digital extensions is able to be notated as part of a performance score [61]. This is very different from the intelligent agents of Downie or Sparacino's work, where the behavior of the accompanying system can be affected by the performer, but is not controllable by the performer or repeatable with the same performance input.

Illustration 7: Yo-Yo Ma performs with the Hypercello.

(Photo from Tod Machover)

In the Hyperinstruments paradigm, aspects of a musician's performance are measured, those performance measurements are analyzed by computer programs, and musical output is generated based on the live performance [43]. Performance information can come from audio or music data (such as MIDI), as well as from the movement of the performer. Examples include the Hypercello [43] and the Hyperbow [84], both of which captured data about the musician's physical movement and used that to generate and control layers of digital music as well as to affect the sound of the analog instrument. The Hypercello measured such gestural parameters as the angle of the performer's wrist, the pressure of his fingers on the strings and the bow, and the bow position. Additionally, this system interpreted certain playing styles from this information and provided those abstractions as additional mapping parameters. In Machover's piece *Begin Again Again...,* commissioned by Yo-Yo Ma, the Hypercello tracked this variety of continuous gestural parameters and could use those parameters to control aspects such effects applied to the audio sound. The Hyperbow provided information about the position, acceleration, and strains on a carbon fiber bow, and this data could be used in real time to modify the sound of an electric violin.

Additionally important to the Hyperinstruments model is the ability to give amateur performers the ability to control and create sophisticated musical content through their expressive performances, as in Machover's Brain Opera [82] and Toy Symphony [44]. As digital technology allows the sound generation to be separate from the control instrument, simple interfaces can be used for complex and rich sounds. The form of these amateur performances can be quite varied, from novel percussion instruments [82, 80]; expressive free-gestures, as in the Brain Opera's Gesture Wall [82]; tangible interactions with physical objects, such as the Shapers in Toy Symphony [44]; or even singing, such as the Brain Opera's Singing Tree [52].

2.1.2.2: Gestural Control of Interactive Music

Illustration 8: Laetitia Sonami and the Lady's Glove.

(Image from Laetitia Sonami)

Other artists and researchers have similarly been inspired by the Hyperinstruments model in their work, exploring the ways in which the expressive performance of a musician can be captured through digital technology and used to augment the musical experience of that performance. A category of these performance capture instruments that is particularly relevant to this work is those instruments that use the performer's gestures and movement directly to control music and sound, rather than shaping the music through the interaction with a tangible, physical instrument. As the human body is incredibly capable of expression and communication through motion, movement is likely to be a particularly rich channel of expressive and affective information for musical performance. To record this movement information, many gesture-based instruments incorporate sensor systems worn on the body of the performer. Others incorporate vision systems to track the movements of a performer playing a traditional instrument, as in Overholt et al.'s system to recognize gestures of a flute player for cuing a computer-generated instrument [55]. Other interfaces use the movement of the performer's body in space or in relationship to objects in order to create and shape the music. The Theremin, patented by Leon Theremin in 1928, is an early free-gesture interface, an analog sound generator whose pitch and amplitude are shaped by capacitive detection of the performer's hands in relationship to two control antennae.

One well-developed gestural instrument is Michel Waisvisz's "The Hands," which incorporates small keyboards on the player's hands, pressure sensors manipulated by the player's thumbs, and sensors to detect the tilt of the hands and the distance between them [9]. Waisvisz used this instrument to manipulate a variety of parameters to trigger MIDI instruments, change the sound of his voice, and manipulate sonic sources in a variety of ways. "The Hands" was developed and refined over a number of years, beginning in 1984, incorporating more sensing parameters and additional layers of control and musical shaping.

Illustration 9: Michel Waisvisz and "The Hands."

(Photo from STEIM)

Another such instrument is Laetitia Sonami's "Lady's Glove," developed by Sonami and Bert Bongers [10]. This glove utilizes flex sensors on each finger, a Hall Effect sensor on the thumb and magnets on the other four fingers, switches on top of the fingers, and ultrasonic receivers. Data from these sensors is used to control sound, lighting, and even motors. Additionally, Sonami has used the glove as an instrument to manipulate the sound of her own voice and other audio streams that are created live, rather than simply prerecorded [66]. Similarly, the French singer Émilie Simon performs with an arm-mounted controller, developed by Cyrille Brissot of IRCAM, that allows her to sample and manipulate her voice and the sound of other accompanying instruments [37].

Illustration 10: The Bodycoder system used in a performance of Etch.

(Photo from Wilson and Bokowiec).

A full-body gestural controller that has been used in musical performances is the Bodycoder System created by Marc Bokowiec and Julie Wilson-Bokowiec [12]. In early forms, this system employed resistive sensors on knee and elbow joints and keypad-like switches in a glove. Switches triggered pre-recorded samples and selected particular audio and visual interactions. In the authors' more recent work with the Bodycoder System in vocal performances such as "The Suicided Voice" and "Etch," the glove switches trigger particular MSP patches and video events, and all sound manipulation is performed live [7]. This system, like the Lady's Glove and The Hands, is a gestural instrument that can be adapted for a variety of different performance pieces. The technology was designed to be mapped in a variety of ways to output media. The functions of each switch and the sensitivity and range of each resistive bend sensor can be adjusted for each performance work and even within the course of an individual performance [8].

There also exist wearable performance interfaces for music that use just the movements of the lower body for sound control and generation. One notable example of these instruments is the Expressive Footwear shoes developed by Joe Paradiso [56, 57]. These shoes measured such movement aspects as the pressure in the toes and heels, the bend in the sole of the shoe, the twist of the foot, and the orientation and position of the foot [56]. With the wide array of different movement parameters coming from these shoes, mappings of data to music generation were done at a variety of levels of complexity, with some specific parameter thresholds triggering specific musical events (such as cymbal crashes or glissandi) and other values serving to control the volume, octave, or voicing of some of these events [57].

There has also been significant previous work on capturing the expressive movement vocabulary of a live conductor for the purposes of digital or digitally enhanced music performance, using on-the-body sensors and/or visual processing techniques. An early example

Illustration 11: Keith Lockhart, conductor of the Boston Pops, wearing Marrin's Conductor's Jacket.

(Photo: Associated Press).

of this work is Tod Machover's use of a dataglove to capture hand gestures and give a conductor another channel of precise control over the audio mixing and timbre in Machover's composition *Bug-Mudra* [43]. Another notable example in this category is Teresa Marrin's “Conductor's Jacket” [45]. This system used EMG sensors on the conductor's biceps and triceps to detect muscle tension, along with a chest strap that collected physiological data such as heart rate and galvanic skin response. Marrin's extended work with the system, as described in [46], showed that the muscular tension of the arms provided the most data about expressive dynamic intensity from pianissimo to fortissimo. Interestingly, Marrin also found that the elements of physiological data outside the performer's direct control (such as galvanic skin response) did not have a strong connection to the performer's intended performance of emotional expression. For example, Marrin found that the strongest peaks of the GSR were correlated with the conductor reacting to mistakes in the performance, rather than correlated with the conductor indicating strong emotions in the most expressive sections of the music.

All of these wearable systems attempt to take some information from the performer's movement and use that as input to shape a musical performance. Of prime importance, however, is the association between movement and the resulting sonic manipulations. Too often, this connection is not clear. One sees a performer doing some complex movement and hears some complex transforming sound, but the intention is not present. Is the performer shaping the music through his movement, or is the music predetermined and affecting the performer's movement? If the vocabulary of movement, the sound, and especially the mappings between the two are not very carefully constructed, the connection can be weak. The audience's view may also be influenced by the “traditional” relationship of dance-like movement and music, where a dance is performed to preset music, rather than the music being created by the dance. Once a performer starts executing significant free-gesture movement, they run the risk of appearing to be using this dance-from-music model, unless they make careful associations between that movement and the sound that is created by it.

2.1.2.3: Opera

Illustration 12: Robert Lepage's La Damnation de Faust

(Photo: Metropolitan Opera)

Technology has also found a place in the relatively new performance form of opera. Compared to the millennia for which people have practiced music, dance, and theater, opera, with its roots in Italy in the end of the 16^{th} century, might be seen as a fairly new model of performance, still developing, still ripe for new explorations. In addition to its relative novelty as a performance form, opera draws on elements from a variety of other performance traditions, combining musical performances, narrative storylines, theatrical design elements such as costume and scenic design, even the occasional dance number. Correspondingly, a variety of opera productions and new operas also incorporate technological performance elements into the medium's rich palette. For example, Tod Machover's *Valis* used two early hyperinstruments to create the musical score, with computer-generated music extending the live performance of a digital piano and a percussion instrument. *Lost Highway*, an opera based on the film of the same name by David Lynch, uses intricate live and prerecorded video streams and a rich synthesized soundscape to translate a complex movie into a live musical performance. This production was directed by Diane Paulus with video design by Philip Bussman [32]. *StarChild* (1996) [53] is an example of a "multimedia opera," incorporating surround-sound technology, planetary data sonification, and precise synchronization between a number of audio and video streams. The Canadian director Robert Lepage has also brought interactive performance technologies into the world of opera. Lepage's 2008 staging of Hector Berlioz' *La Damnation de Faust* for the Metropolitan Opera used microphones to capture pitch and amplitude of the performers' voices and the orchestra's music, as well as infrared lights and cameras to capture motion. The data from these sensors was used to shape projected images in real time, such as projected curtains waving behind dancers, or giant projected flames that varied based on the singer's voice [78]. However, most of the mappings used in this production were fairly simplistic, not powerful instrumental extensions of a performer's technique.

Another opera that draws on highly sophisticated technology as an integral part of the performance experience is Tod Machover's upcoming *Death and the Powers* [49]. This opera, directed by Diane Paulus with production design by Alex McDowell, tells the story of a wealthy and powerful businessman, Simon Powers, who finds himself near the end of his life. Powers seeks to keep his mind, his influence and emotions and ability to interact, in the world a little longer. To achieve this goal, he uploads his consciousness into a computer system that he has constructed, which extends throughout his house. Theatrically, this means that for the majority of the performance the set – the lighting, the scenic pieces, the furniture – must perform as a main character. A variety of sophisticated new technologies are used to tell this story, from robotic performers to ambisonic sound manipulation to the Disembodied Performance system, used to translate a live performer's presence and emotional state into a robotic stage. Disembodied Performance and my role in developing performance capture technologies for the singer playing Simon Powers are discussed in detail in Chapter 3.

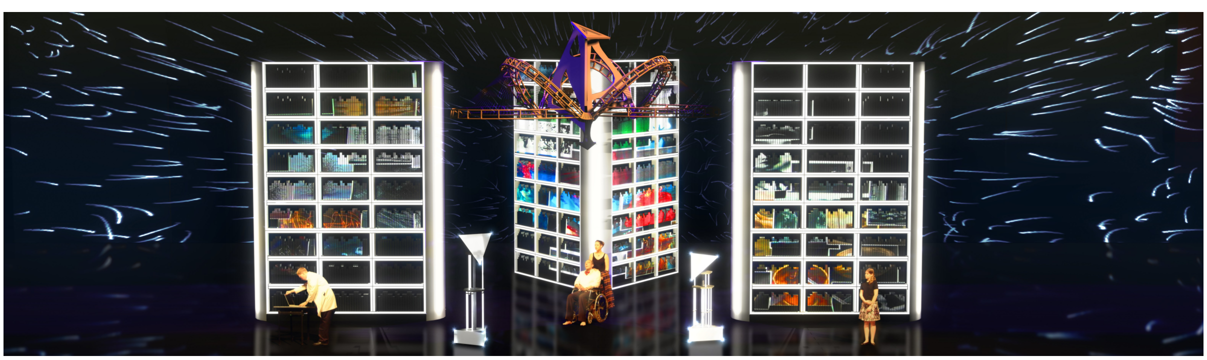


Illustration 13: Stage Rendering for Death and the Powers.

(Image from Peter Torpey)

2.1.3: Mapping Systems for Technological Performances

Due to the many artists manipulating digital media in a variety of ways, as discussed in the previous sections, numerous mapping systems for digital media have been developed to make it easier and more intuitive to create control programs for media, especially programs that can be modified during a performance by input parameters from external sources or from a computer operator. Max/MSP [19] is one of the most popular of these mapping systems, and is popularly used for sonic and musical performances. Max/MSP, an interactive, graphical programming environment, incorporates a visual programming language that allows users to develop music, audio, and media pieces and adjust them in real-time. In Max, visual objects represent such entities as sound sources, mathematical operations, and sound manipulations. Each object has inputs that it performs operations on and outputs the resulting information. By drawing connections between these objects, the user determines the flow of information through the program. These connections can be adjusted and manipulated live; similarly, objects can be created and modified while a program (referred to as a "patch") is running.

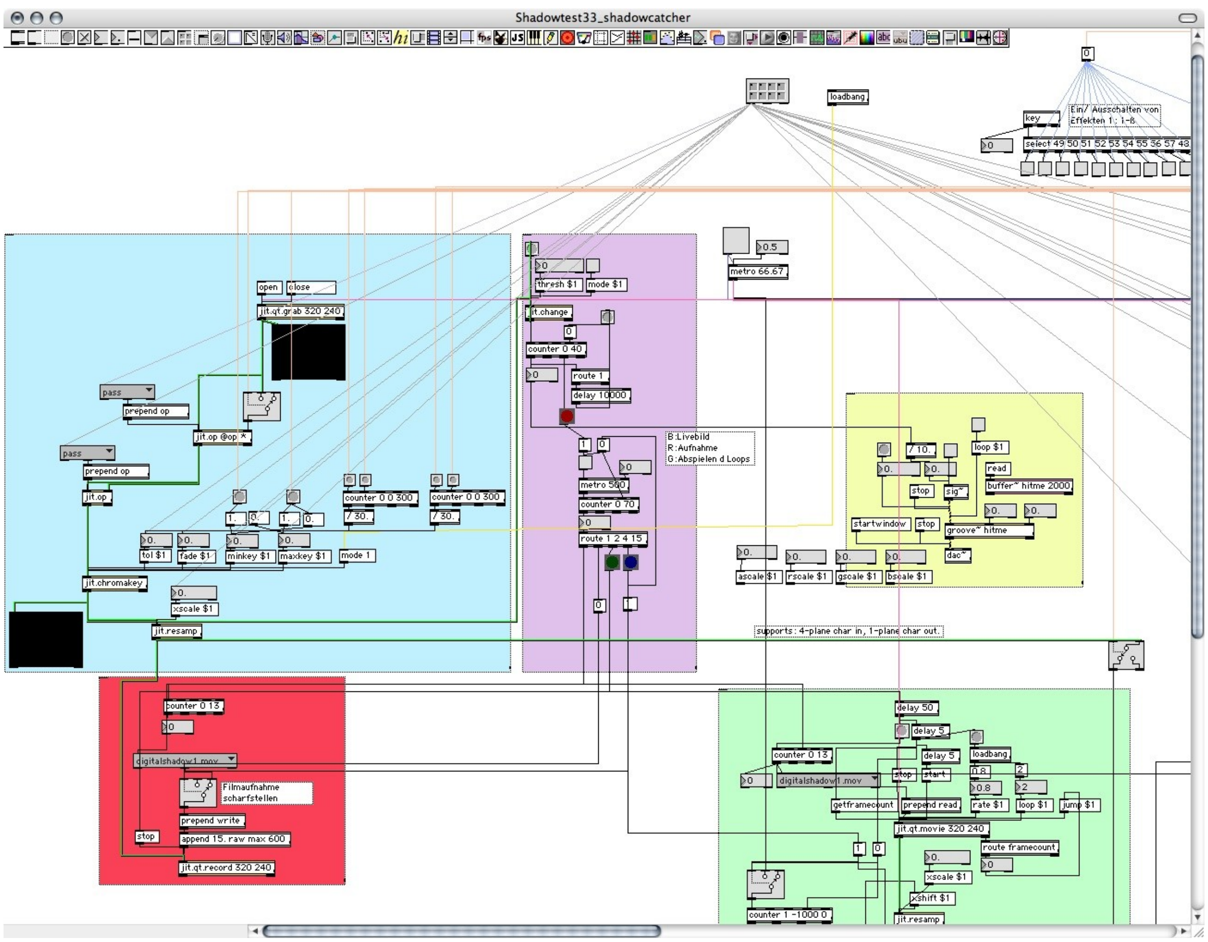


Illustration 14: Example screenshot of a Max/MSP patch.

The dance company Troika Ranch, seeking to create a tool that would allow them to more easily control visual elements and video projection during live dance performances, has developed the software system Isadora [71, 76]. Similar to Max/MSP, Isadora is a graphical programming environment with graphical object representations of different media elements and transformations, connected by lines delineating the flow of information. However, while Max/MSP's primary focus is control of audio and musical data, the majority of the objects in Isadora are designed for sophisticated manipulation and mixing of video streams, either live or prerecorded. Isadora also provides tools for MIDI and serial input into the resulting patches, as well as keyboard and mouse input. Troika Ranch has developed sensor systems to integrate with Isadora, such as the MidiDancer system, a wearable bodysuit that outputs MIDI data about the amount of bend in major joints on a dancer's body.

However, neither of these systems, nor other existing mapping systems, incorporate higher-level gesture representation, or even much conception of gesture. Isadora can receive "gesture" data from Troika Ranch's MidiDancer system using a built-in object, but this gesture data is simply MIDI values corresponding to the amount of bend in various joints. The system does not have the capabilities to work with any higher-level encapsulation of movement qualities or gesture. The Open-Ended Group's Field [23], a coding platform for digital artists developing work for live performances, seeks to allow easy experimentation during a rehearsal process. However, Field similarly has no gestural abstractions, or even any built-in concept of gesture or movement (though it has been used in works incorporating motion capture data [22]). Even EyesWeb, mentioned previously, [13], which explicitly attempts to capture "affective" and "gestural" qualities of movement from webcam footage of dancers, has fairly low-level abstractions of movement, looking at aspects such as a dancer's "Space Occupation Area," "Directness Index," "Amount of Periodic Movement," and "Symmetry Index" [14].

These systems all affect the user's thought process about creating mappings through the abstractions they incorporate or leave out. By having little concept of gesture or quality of movement, and by only taking in sensor data or slightly-processed computer video footage, they limit the ways in which a user can easily think about designing mappings. This limitation encourages naïve mappings, such as mapping the amount of bend in the elbow to the pitch of a note or the amount of reverb on a sound, and does not inspire more sophisticated thought about the emotional, expressive, or semantic aspects of movement. Thus, if users want to create more sophisticated mappings with gesture and media, they generally have to reinvent their own recognition and mapping systems.

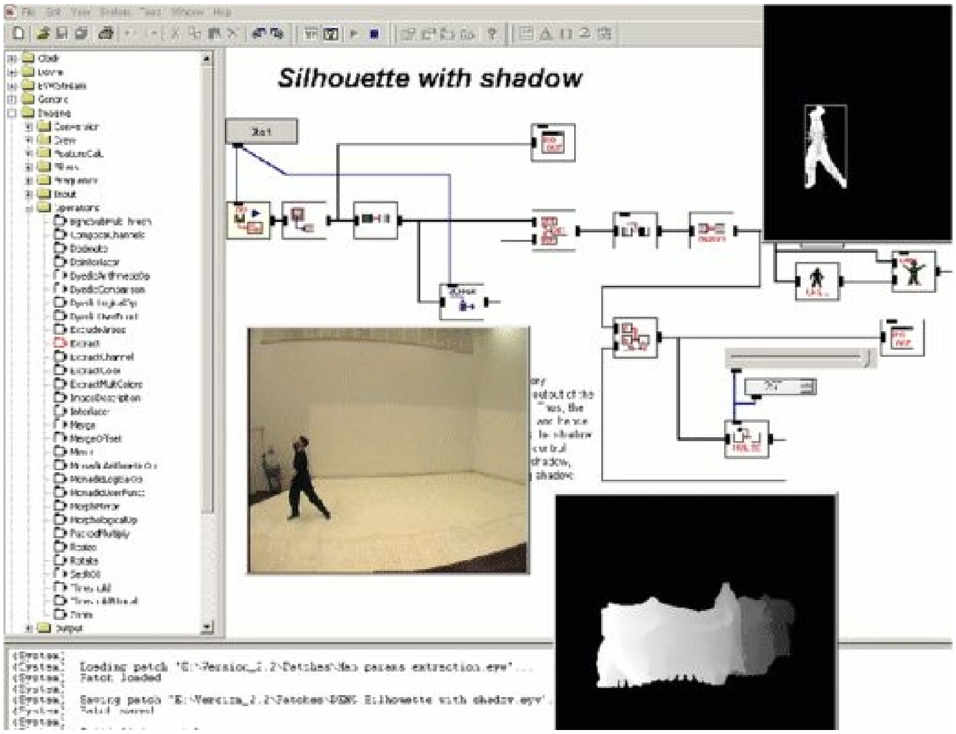


Illustration 16: Example screenshot of an EyesWeb patch, showing figure detection and overall movement of the dancer's body over time.

David Merrill's FlexiGesture project [48] explores the need of meaningful mappings between gesture and musical sound, and gives an example of a hand-held digital instrument that allows users to create their own associations between a given sonic vocabulary, the discrete gestures they could make with the instrument, and the continuous control parameters of the instrument. Merrill found that such a trainable instrument could be a compelling performance tool, as well as provide a platform for exploration of gesture/sound relationships. While Merrill's work is a beneficial step in developing flexible mappings between movement and sound, the field of performance is still lacking a multipurpose mapping system or abstraction framework that could encourage higher-level thought about and exploration of full-body gesture and movement mappings, especially focusing on the movements of the body through space rather than the movements of the body in relationship to other objects..

2.2: Gesture and Dance

It is clear that technologies for interacting with movement in performance are varied and popular, though still lacking in some important aspects. For the development of new gesture recognition and movement quality recognition technologies for dance and musical performance, it is important to consider the role of choreographic movement and gesture in non-technologically enhanced performance, particularly for the wide range of movement vocabularies encompassed by the label of "modern dance." Additionally, it is useful to draw from prior attempts to codify gesture and gestural qualities in music and dance. This section will first briefly examine the development of modern dance as an art form, and discuss a few modern choreographers whose work has been influential to me as a performance-maker/choreographer and the way that these choreographers use gesture and movement in their work. Following this, I will review a few historical and popular systems of dance notation and the ways that those systems attempt to describe movement and qualities of movement. Additionally, I will briefly examine the use of gesture in Delsarte's System of Oratory, the first and only attempt at a formal description of theatrical gesture. These areas and their ways of thinking about and describing movement all informed the development of my work on the Gestural Media Framework.

2.2.1: A Brief History of Modern Dance

Modern dance began early in the 20th century as a reaction against the strict forms of ballet; in fact, the term "modern dance" was originally used as a catch-all phrase to refer to "serious-theatrical-dance-that-is-not-ballet."[47] Modern dance pioneers such as Loie Fuller, Isadora Duncan, and Ruth St. Denis were some of the first to step out of the box of ballet and create serious, expressive dance. The American Loie Fuller, mentioned earlier in this chapter, began her dance career around 1889, with performances that focused on the visual effects that she could create by moving with layers of flowing clothing under electric stage lights, or even with glowing radium attached to her costumes. Fuller's performances were also seen by Isadora Duncan and Ruth St. Denis, who both created their own forms of modern dance. All three performers were instrumental in transforming dance into a vehicle for expressing emotion through a variety of styles of movement, free from the confining techniques of ballet [47]. In their work, these women explored unrestrained and flowing movement vocabularies, inspired by classical mythology.

From these early creators, modern dance spun in a variety of different directions, with choreographers creating movement to express emotions and tell stories, or to explore the capabilities and intricacies of the human body, without purposefully conveying narrative content. A complete survey of these artists is outside the scope of this thesis, but I will mention the work and choreographic process of a few choreographers who have been influential to my working process as a choreographer. The first of these is Martha Graham, hailed as the "mother of modern dance," who began her dance studies at Ruth St. Denis' dance school, Denishawn, in 1916 when she was twenty-two years old [47]. Graham then went on to develop her own technique of modern dance and to become one of the primary figures in the history of the form. Graham's work focused on movement and dance as a key to expressing deep emotions and emotional truths, stemming from her long-held belief that "movement never lies" [29]. Much of Graham's work is deeply psychoanalytic and dramatic, telling stories (often with mythic components) and exploring the motivations and passions of the characters. To control the strong emotional core of her choreography, Graham developed a movement vocabulary that was just as structured and technically precise as ballet, centered around the opposing movements of contraction and release. Graham's work was also significant in its focus on bringing together movement, music, costumes, lighting, and scenery to create a full theatrical experience. However, she would not put anything on the stage that was not essential, a focus that is worth remembering when one begins putting a variety of technologies on the stage with performers.

Another major figure in the development of modern dance was Merce Cunningham (mentioned earlier in this chapter), one of Graham's dancers from 1939 to 1945. Cunningham rejected Graham's belief that every movement had a specific emotional meaning; instead, he believed that the most important element of movement was what it was, not what it "meant." Cunningham did not believe that dance should be devoid of emotion; however, he believed this emotional content was inherent in pure motion, not an external layer of symbolic meaning imposed on top of movement. Cunningham's choreography incorporated some pedestrian movements (such as walking and running) combined with highly physical and precise leaps, jumps, and turns. The use of chance scores to develop sequences of movement is also a key element of his choreographic work, letting him abandon traditional narrative and linear structures for a form where many equally important events occur simultaneously [47].

Another modern dance choreographer whose work is particularly significant to my own is the choreographer and performer Trisha Brown, who was born in 1936 and has been choreographing since the 1960s. Many of Brown's early movement experiments took place during her association with the Judson Dance Theater, an umbrella for a number of choreographers that worked with the belief that any movement could be "dance" (including purely pedestrian and non-technical movement) and any person (trained or not) could be a "dancer." Brown's works have focused on pure movement and physical control, including a number of pieces that are based primarily on mathematical, spatial, and geometric structures. She also developed structural forms based on the accumulation of gestural vocabularies. Trisha Brown's detailed and precise structures serve as a key element of her choreographic process [GOLDBERG]. Brown, like Cunningham, has also occasionally integrated technological aspects into her work, such as in her collaboration with the Open-Ended Group on *How long does the subject linger at the edge of the volume*, a piece incorporating intelligent software agents that attempt to connect their projected graphics to the performers' movement [22].

Finally, the choreographer who has perhaps been the most influential for my own movement and performance work is the late German choreographer Pina Bausch. Bausch was born in 1940 and trained under the modern dancers Kurt Jooss and José Limón. Bausch, the founder of the Wuppertal Tanztheater and its director from 1973-2009, works with a mix of movement and text that bears little resemblance to the highly technical modern dance systems that were prevalent in the first half of the 20th century [28]. Her non-linear collage structures and use of spoken word and song, as well as her focus on telling stories rather than on the exploration of pure movement, set her apart from even her immediate predecessors [64]. Much of her movement vocabulary explores common body language and the artificiality of typical gestures. She combines pedestrian movements and everyday gestures (such as walking, running, or caressing) with her own stylized dance vocabulary to create sequences of intense and physically demanding movement. A major characteristic of Bausch's work is repetition; as the performers repeat a pedestrian gesture or interaction over and over again, the significance and meaning of the gesture develops and transforms [26].

With this rich range of choreographic working processes and uses of movement and gesture, it is clear that any gesture and quality recognition system, if it is to be usable by a variety of different choreographers, must be highly flexible both in the vocabulary it can incorporate and in what aspects of the movement are determined to be important. It is necessary for a system to handle a wide range of qualities of movement, variety of movement, and movement content. The process of enhancing a Graham piece through mapping visuals or sound to the movement would necessarily be quite different than the process of working with a piece by Pina

Bausch.

2.2.2: Dance Notation Systems

Given the range of dance movement briefly mentioned above, it has been a longstanding and varied effort to develop formalized ways of describing and notating dance movement. Ann Hutchinson-Guest breaks down the description of movement into several aspects: timing, parts of the body, spatial variation, and quality. She describes dance notation as "the translation of four-dimensional movements (time being the fourth dimension) into signs written on two-dimensional paper. (Note: a fifth 'dimension' – dynamics – should also be considered as an integral part, though usually it is not.)"[30, pp xiv]

The earliest systems of dance notation arrived in the fifteenth century, and were used to write down the social dances that were common in that time period. These were simple letter notation systems: each dance step pattern had an individual name that could be notated by the first letter of the name, with a dance sequence being described by a sequence of these step patterns [30, pp 42-46]. As social dances evolved and included more complex patterns of movement around the floor, notation systems began to include birds-eye views of these patterns, including systems by Playford (1651) and Feuillet (1700). Feuillet's system took floor patterns to a greater level of sophistication, as this system notated the particular steps taken (with limited notation of arm positions), the dancer's spatial path, and the timing in relationship to the music (see figure). However, as theatrical dance developed a greater vocabulary in the end of the eighteenth century, Feuillet's system proved insufficient to capture the range of gestures [30, pp 62-67]. New notation systems similarly attempted to link specific movement steps to their proper timing in relationship to a musical score, such as the system of Saint-Leon (1852), which transformed musical notes into stick-figure-like annotations about movement. Similarly, Stepanov's anatomically-based system (1892) used note values and a staff as in music notation, but used the location of notes on the staff and marks across the note stems to indicate body part and direction and level of movement [30, pp 72-74].

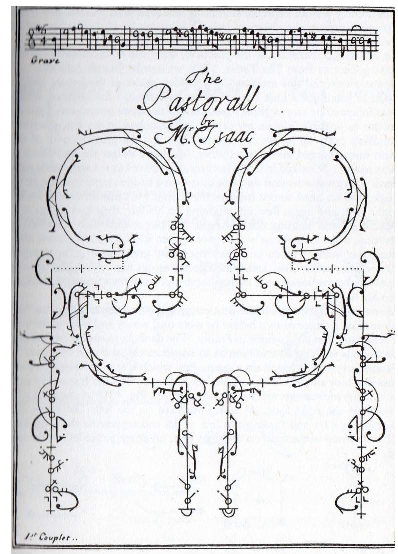


Illustration 17: An example of Feuillet Notation

(Image from [30]).

As dance vocabularies continued to develop and expand, particularly with the introduction of modern dance in the 20th century, the existing notation systems proved inadequate to capture the full range of movements appearing in dance. If the choreography of a piece was not to be limited to a specific, already-named set of motions that were often to be performed in a specific manner (as in ballet), the task of preserving some description of the movement became increasingly challenging. (In fact, most notation systems are still primarily used for recording ballet choreography.) Morris (1928) developed an anatomically-based system similar to Stepanov's that included detail about breathing, facial expression, and muscular tension [30, pp 79]. Other systems such as Zadra's (1935) and Ruskaja's (1940) used abstract symbols to represent particular movements. However, the notation systems that have become most popular are those of Rudolf Laban, Joan and Rudolf Benesh, and Noa Eshkol/Abraham Wachmann.

Labanotation, likely the most popular dance notation system in the present day, is the only dance notation system that combines into a single notation symbol the direction of the movement (represented by the shape of the symbol), the timing (the symbol's length), the level (the symbol's shading), and the body part used (the symbol's location on a staff) [30, pp 84]. Laban saw movement as a form of communication where the details of a gesture could convey the inner state of the performer to those observing the movement. Labanotation also, uniquely in dance notation systems, puts emphasis on how the performer's movement is related to other performers or to objects in the space, and, most importantly for our purposes, on the dynamics of a performer's movement [30, pp 87]. Laban's Effort System, describing the type of muscular energy used in a movement, divided movement dynamics into four axes reflecting *Time*, *Weight*, *Space*, and *Flow*. Laban's studies of Effort and the ways that I incorporated aspects of descriptions into the Gestural Media Framework's definitions of movement quality, are described in detail in Chapter 4.

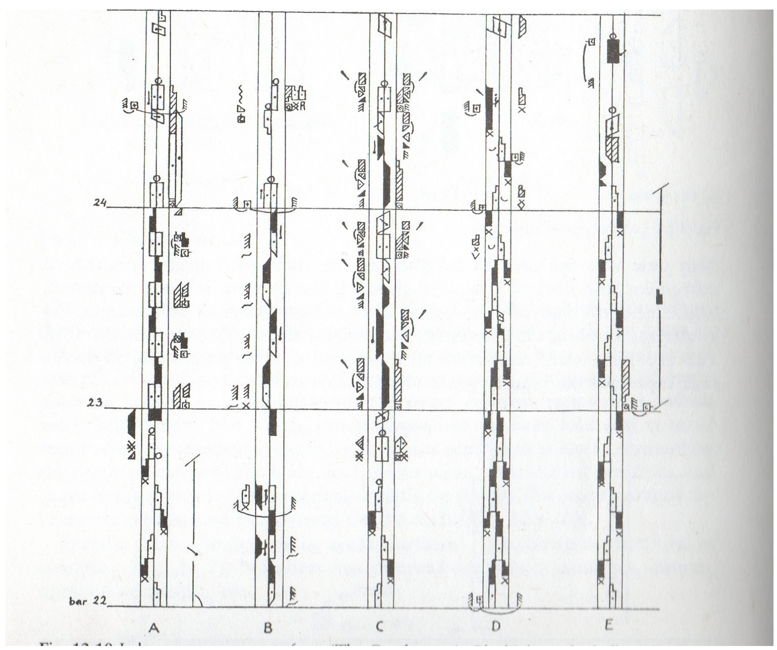


Illustration 18: Example of Labanotation from The Green Table *by Kurt Jooss*

(Example from [30])

Benesh Notation plots movement left to right in a staff, as in musical notation. Patterns of movement on the floor are written beneath the staff, the positions and movements of key points on the body and limbs are marked with simple lines on the staff, and information about rhythm and phrasing is marked above the staff [58]. This notation system is concerned with creating a simple and flexible language structure for dance notation, shaped by linguistic principles. The Benesh notation framework assumes a basic "alphabet" of physical movement that can combine in a variety of different ways for various forms of dance. The specific details of how movements are performed in a particular dance form are assumed to be known to the person reading and writing the notation system [30, pp 103-104]. Thus, while this system can capture some basic elements of movement, it does not communicate how the movement ought to be performed.

Finally, the Eshkol-Wachmann system uses a mathematical approach to movement, with standardized angles for movement displacement and units of time. This system's movement analysis is based on the concept that all movement is essentially circular, and can be seen as on conical paths carved by full limbs. This system looks at the body purely as a vehicle for movement, without any analysis of dynamics or expression beyond the timing of movements.

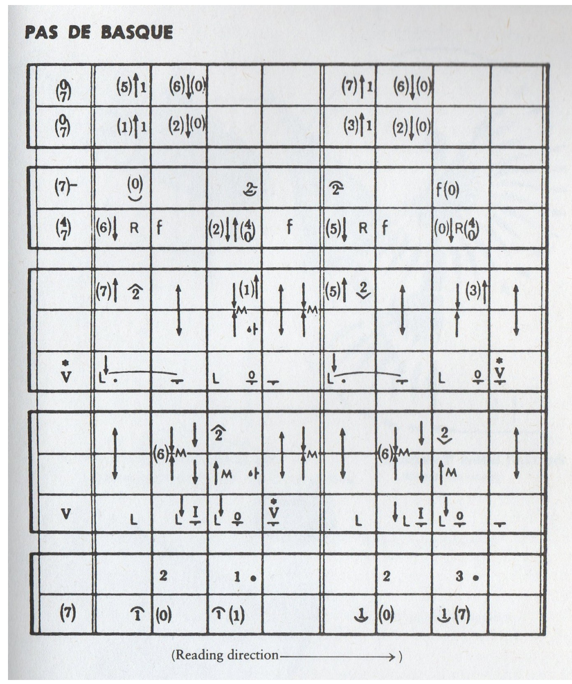


Illustration 19: Eshkol-Wachmann Notation (Example from [30])

It is clear from this review that there is little formalized notion of movement qualities and dynamics in the majority of dance notation systems. And yet, such aspects of movement are a significant component of what we see as "expressive." I theorize that although movement sequences are notated only as information about direction, shape, and time, losing significant information about quality, the expressive content can frequently be regained when this notation is transferred back to a performer's body because of the skill, expressiveness, and physicality of the dancer. The lack of dynamics in the notation can be compensated for by the details the performer adds. However, if we are attempting to capture and work with some part of the expressive content of the performance through technological means, we need ways to formalize exactly what is meant by the quality of movement. My work in this direction has been most inspired and informed by Laban's theories of Effort, as is described in Chapter 4.

2.2.3: Delsarte's System of Oratory

Outside of the framework of dance, there are limited formalized theories of movement in performance. One of the earliest (and one of the only) efforts to create a comprehensive theoretical framework of performed movement was made by François Delsarte in the late 19th century. Delsarte was a former actor who turned his efforts to oratory and to developing a theory of aesthetics based on the ways that the inflection of the voice, the muscular movements of the body, and the content of a speech conveyed the speaker's life, soul, and mind (which Delsarte held to be three separate and interconnected entities) [20]. In Delsarte's framework, gesture served as the conveyor of a person's "soul," that is, their sentiment and emotion, and was the most powerful of these oratorical elements. He stated, "The artist should have three objects: to *move,* to *interest*, to *persuade*. He interests by *language*; he moves by *thought*; he *moves, interests,* and *persuades* by gesture."

Delsarte held that there existed a specific gesture and stance that was the ideal form to convey a desired sentiment to one's observers. Each emotion was connected to a distinctive set of movements and positions of the eyes, arms, hands, and full body. For example, the head can take on nine separate positions, each of which conveys a different emotional state, such as confidence, pride, reflection, or veneration. Similarly, nine different stances of the legs are seen to reveal different states of the speaker's mind, from vehemence to terror. Additionally, Delsarte believed that gestures should be limited, controlled, and focused on one at a time.

Delsarte also noted an important point that gesture goes beyond simple poses of the body and limbs, the static components of gesture. For Delsarte, the "dynamic" of gesture contained the inflictions and rhythms of a movement. This concept of dynamic movement is an important characteristic of Delsarte's analysis of gesture: he saw that movement was communicative and semantically meaningful not only through the performer's stance and pose, but also through the way that the shape of the body changed over time. This concept of gesture as a semantic, time-dependent language was also influential for me in my work on the Gestural Media Framework.

2.3: Gesture Recognition in Human-Computer Interaction

2.3.1: Gesture Recognition Applications and Techniques

From the preceding examples of technology for performance and of movement notation systems, it has clearly been a fairly complex process to categorize and describe gestures in the context of performance, music, and dance. However, in the field of Human-Computer Interaction, gesture recognition has become an increasingly popular tool, and significant research has been done on gesture recognition and gestural vocabularies for interacting with computers. In developing new performance tools that incorporate gesture recognition, it is valuable to draw on the rich body of related research from HCI, including gesture sensing technologies, recognition algorithms and techniques, and movement vocabularies. The majority of gesture recognition systems in this field are used for sending commands to a computer through specific movements, replacing commands from mouse clicks or keypresses. Often, these gestures are made with a mouse movement, a finger (or multiple fingers) on a surface such as a tabletop or touchscreen, or a tablet pencil. I will limit my examination to those gestural systems that require larger-scale body motions, as these are more related to the process of movement recognition in dance and musical performance.

Gesture recognition in HCI has been performed using a variety of input technologies, including computer vision systems [69, 38], handheld devices [72, 1, 63], wearable systems [5], and EMG sensors [83]. Additionally, this research has used and expanded a variety of pattern recognition/machine learning algorithms, such as Hidden Markov Models [69, 86, 24], Principal Component Analysis [6], Dynamic Bayesian Networks [3], and Neural Networks [50]. Many gestures and poses with applications for HCI, as well as a number of gesture recognition technologies, are summarized in [62]. However, there are limitations in the adaptation of HCI technologies for performance contexts. Standard gesture recognition systems work best for applications where there is a preset gestural vocabulary and all movements made by the user fall into that predetermined vocabulary. These systems generally have no concept of the expressiveness of a gesture, and have little ability to pick out important gestures from a variety of movement.

Many gesture recognition systems rely on the user to hold an object that can detect motion (for instance a cell phone). Here, the measured and recognized information is the movement of that device. Example applications include a system where different kinds of information are retrieved when a portable device is moved near particular body parts [1] and a music player controlled in a similar manner [71]. Recognition techniques using Hidden Markov Models have also been developed for identifying gestures using the accelerometers contained in a Wiimote [63]. The limitations of these gesture recognition systems include the need for the user to consistently hold an object while performing gestures, as well as the ability to separate performing a gesture from performing specific body movements. For example, in Nintendo's Wii Bowling game, the system still recognizes a bowling gesture if you just swing your wrist so that the controller moves appropriately, rather than requiring the user to make a full-arm swing and take a step forward as the game instructs the user to do.

A frequent use of gesture recognition in HCI that is particularly relevant in capturing communicative movement is in tools for recognizing sign language. Frequently, these tools use computer vision systems to track the user's hands and Hidden Markov Models to perform the gesture recognition [36, 69, 81]. However, most of these recognition systems are limited to identifying gestures one at a time, with the user having to pause briefly between gestures to signify the start and end of these gestures. Kelly et al. attempt to identify gestures from a continuous stream of sign language [36], an even more challenging task. These researchers use Hidden Markov Models as the machine learning component of their application, with one Hidden Markov Model trained for each gesture they wanted the system to recognize. Additionally, their system is equipped for continuous gesture recognition through the addition of an additional HMM trained for movement epenthesis, the small movements of the hand from one place to another between recognizable gestures. Thus, they could sort out the noise of non-meaningful movement from the important gestures.

A very limited subset of HCI gesture recognition research goes beyond seeing movement as command statements or symbolic gestures and attempts to recognize the emotional content of gestures, examining ways in which affective gestures could lead to better interactions with computer systems. Fagerberg et al. have developed an affective model of gestural input, where the user can explicitly express emotions to the system through gestures [25]. This research is inspired by Laban's notions of Shape and Effort as well as by Valence, a standard parameter used in affective computing. They then used this system in designing gestures for a user to represent emotions to be conveyed with an SMS message. This system recognizes a small set of gestures determined to be emotionally resonant, but cannot recognize emotional content of movement independently.

2.3.2: Gesture Recognition Toolkits

In all of these systems, it is very important to develop fitting connections between the performance of a specific gesture and the resulting computer action. This is often made more difficult when designers do not have in-depth knowledge of pattern recognition and machine learning techniques for gesture recognition. Therefore, some researchers have developed prototyping tools to make it easier to develop and explore gestural interactions. Bjorn Hartmann's Exemplar [31] is one such system, giving a designer tools to quickly program sensor-based interactions through providing and manipulating examples of those interactions. While Exemplar does not explicitly focus on gesture recognition, its basic premise of rapid prototyping through pattern recognition of sensor data can be easily applied in this field. This has been done in the MAGIC system, developed by Ashbrook and Starner, which assists users unskilled in pattern recognition techniques in designing distinct and recognizable gestural vocabularies that do not overlap unduly with gestures made in the course of daily life [2].

As previously mentioned, a major challenge in incorporating gesture recognition into interactive applications has been the complexity of the machine learning algorithms required and the effort and understanding required to program and work with those algorithms. Additionally, most researchers working on systems that involve gesture recognition end up needing to recreate recognition algorithms from scratch. Because of these challenges, it is time-consuming to experiment with gesture recognition or integrate gesture recognition into interaction design. Therefore, researchers at Georgia Tech have designed an open-source toolkit specifically for the purposes of making it easier and quicker to create your own gesture recognition tools through abstracting the programmer's interaction with the necessary algorithms, with the goal of letting researchers spend more time exploring gesture recognition and less time recreating gesture recognition techniques [81]. This Gesture and Activity Recognition Toolkit, or GART, became quite useful for me in the implementation process of my work, as described in Chapter 4.

2.3.3: An Introduction to Hidden Markov Models

The Georgia Tech toolkit bases its recognition algorithms on Hidden Markov Models (HMMs), a pattern recognition technique that is frequently used for speech recognition but can also be generalized to gesture recognition. A standard Markov Model is a probabilistic model representing a non-deterministic process taking place in a domain where all the possible states are observable. The model consists of states and the probability of transitioning from each state to each other state. Each timestep, the model transitions to a new state based on the transition probabilities. For an example (borrowed from [59]), the model's states could represent certain weather conditions on a given day (say, rainy, cloudy, sunny), and the transition probabilities are the chance that, given a particular weather condition A on one day, it will be weather condition B the next day. Using this model, we can calculate the probability of a specific sequence of states (say, that the weather over four days will be rainy, sunny, rainy, cloudy). Hidden Markov Models are an extension of standard Markov Models, where the states in the model do not represent the states in the domain, but are instead probabilistic functions of the states in the domain. The domain states are unknown (and thus, "hidden"). In this case, both the states and the transitions have probability functions. More details of the Hidden Markov Model algorithm can be found in Rabiner's HMM tutorial [59].

There are three basic problems that are associated with Hidden Markov Models [59]. First, given a specific model, what is the probability that a specific sequence of observable symbols would be generated by that model? Second, what is the most likely sequence of internal states in an HMM that would lead to a specific observable sequence? Third, given one or many observable sequences, what is the Hidden Markov Model that best explains those sequences? For gesture recognition, we primarily care about the first and third problems. In this case, we have a number of Hidden Markov Models, each of which represents a specific gesture. For recognition, we can ask: given a sequence of movement data, what is the probability of each model generating that sequence? For training the recognition engine, we can ask: given a set of sequences of movement data labeled as specific gestures, what are the states, transitions, and probabilities of the Hidden Markov Model that explains those sequences?

Hidden Markov Models are useful for both gesture recognition and speech recognition domains because they are able to handle inputs of varying lengths. There has been substantial development of HMM algorithms in the speech recognition community, particularly with Cambridge University's development of HTK, the Hidden Markov Model Toolkit [33]. This open-source scripting software provides a toolkit for speech recognition using HMMs; however, in order to adapt the use of this software from the domain of speech recognition to that of gesture recognition, it is necessary to have broad knowledge of the speech recognition literature and deep knowledge of Hidden Markov Models. Such in-depth knowledge is made unnecessary by the GART toolkit, as it simplifies and abstracts the interactions with HTK so that it can be easily used in gesture recognition contexts. In particular, this toolkit abstracts the HMM training and recognition process. A set of reusable models is automatically created and trained from a set of gesture examples specified by the user, and these models can then be used to recognize a later example. The details of the training and recognition algorithms (such as the system's use of the Viterbi algorithm for calculating the probability of each model producing the specified example) are hidden. This library has been used for a variety of applications including workbench activity recognition, sign language recognition, and blink pattern recognition [81].

2.3.4: Limitations of HCI Gesture Recognition Models

So what are the difficulties when we seek to extend typical gesture recognition models, as used in Human-Computer Interaction, into the domain of performance? Can we simply use these tools onstage? The primary limitation is HCI's focus on binary gesture identification. In most HCI applications, it is only necessary to determine whether or not a gesture is occurring, perhaps adding some additional information about the direction of the gesture (as in a pointing motion), or perhaps even adding a layer of information about particular sequences of gestures. There is no emphasis, or need, to determine *how* a gesture is performed. In performance, however, the same motion may have quite different implications if it is performed carefully, quickly, lightly, staccato, roughly, intensely, etc.

This limitation is particularly clearly seen when looking at the example of G-Speak, the gestural interaction framework developed by Oblong Industries [51]. This system can identify hand positions and locations in a three-dimensional space with millimeter accuracy, using infrared camera systems and retro-reflective markers on the hand and fingers of gloves worn by the user. Gestures can be defined by programmers using a wide variety of finger and hand positions, and that recognition data can then be abstracted and sent to other applications. This

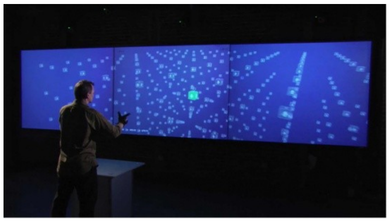


Illustration 20: G-Speak in use for navigating large data spaces.

(Photo: [51])

abstraction process turns gestures into keystroke data, and hand position into mouse data, thus limiting the space of possible gestural interactions back into the keyboard-and-mouse model of interaction. While G-Speak is a powerful gesture recognition system, it does not take advantage of any of the expressive potential of human movement and thus would be highly limiting for performance applications that require continuous control, not simply triggers. Additionally, this system points to another weakness of many HCI "gesture recognition systems": a definition of a static pose as a "gesture." G-Speak primarily recognizes when the user's hand is in a particular position, rather than using information about the hand's time-dependent movement through space. A G-Speak-like system does not use the time information that is so important in the expressive content of a performance piece.

As the G-Speak example points out, despite the significant amount of work already done in the field of gesture recognition for HCI, as well as the significant amount of work on the algorithms and methodology of gesture recognition, the systems that have been designed for those applications cannot be immediately transferred into a performance context. It is necessary to design new tools that explore more of the expressive potential of gesture and movement.

2.4: How is the Gestural Media Framework Different?

With such a rich variety of prior work in many related domains, it is important to explore how my work on a Gestural Media Framework can expand on the successes of this prior work while being a significant step forward in the research. My work focuses on capturing and expanding the rich expressive potential of movement through technological enhancements, while maintaining the dynamism and variety of live performance art. It is inspired by and draws from existing models of technological performance explorations, and incorporates some existing gesture recognition techniques and strategies.

The Gestural Media Framework is significant and unique in the way it addresses the major points of concern brought up in this analysis of related prior work. The first of these concerns is the way that current mapping systems used for relating movement information to media control constrain their users by only using low-level sensor data as input to their systems. Much of the existing work in performance and technology has been done with mapping systems that have these limitations, or by hand-programming connections between the input movement data and the output media. With these tools, it is challenging to develop mappings that are meaningfully related to a performer's movement (and thus are clearly driven and shaped by that movement), or mappings that take advantage of the deep metaphorical and semantic content of that movement. In response to this issue, my Gestural Media Framework seeks to expand on the possibilities for powerful mappings and intuitive exploration of media-movement relationships by providing meaningful high-level encapsulations and descriptions of movement and movement qualities, abstracted away from pure sensor data. The framework separates the recognition of specific gestures and the recognition of particular qualities of movement (rather than only recognizing a gesture if it is performed with a specific movement quality), such that gestures and movement qualities can interact in mapping scenarios, but can also be accessed independently.

Additionally, this framework provides flexible mapping and recognition strategies that are not dependent on a performance-maker or choreographer's particular gestural vocabulary, specific movement sensors, or specific output devices, allowing an individual artist to create mappings that support his vision of a particular performance piece. Traditional mapping systems force the artist to look at questions such as, “What do I want to happen when the sensor on the performer's elbow is bent past a particular threshold?” The Gestural Media Framework allows and inspires the artist to ask (and answer) more intuitive questions such as, “What do I want to happen when the performer reaches his arm forward and curls his hand into a fist? When the performer makes a sudden, jerky movement?” This system seeks to create a space for exploring metaphor in movement, for finding intuitive extensions of the human body, and for creating seemingly magical interactions between a performer and digital media elements.

Second, in HCI gesture recognition systems, performance mapping systems, and even in existing dance notation systems, there is very little focus on movement dynamics and qualities in the description of movement. This lack of awareness of dynamics in gesture recognition systems leads to systems that can replace some keyboard and mouse interaction with gestures, but that cannot be easily used for expressive control of rich media. To address this issue, my Gestural Media Framework incorporates not only specific gesture recognition using standard HCI techniques, but also high-level information about qualities of movement, inspired by Laban's Effort theories. In particular, the Gestural Media Framework implementation lays out a way to translate sensor data about movement into quantitative parameters reflecting Laban's qualities of Weight, Time, Space, and Flow. With this additional functionality, the system can leverage HCI tools in the domain of performance. Due to these differences from prior explorations of gesture recognition not only in the domain of performance but also in the domain of human-computer interaction, the Gestural Media Framework is a step forward in creating new tools for gesture recognition and mapping in performance.

Chapter 3: Toward a Gestural Media Framework

Having discussed related work in the fields of technologically-enhanced performance, dance analysis and notation, and gesture recognition for Human-Computer Interaction, it is also important to set the Gestural Media Framework into context in terms of my own prior work. In particular, over the course of my two years as a member of the Opera of the Future group at the Media Lab, I have worked on several projects where I explored different kinds of relationships between a performer's gesture and the manipulation of various media elements. The methodology and gestural abstractions in these projects were inspirational to me in my work creating a Gestural Media Framework. The first of these significant projects that I will discuss is the Vocal Augmentation and Manipulation Prosthesis, or VAMP, a wearable musical instrument for a singer. The second project I will describe is my work with Peter Torpey on the Disembodied Performance System for the opera *Death and the Powers*. The third project is my development of a Gesture Glove, a glove-shaped controller for computer visuals that served as a proof-of-concept application for the Gestural Media Framework.

3.1: The Vocal Augmentation and Manipulation Prosthesis

In the fall of 2008, I began developing the project that was perhaps most influential in my work on the Gestural Media Framework. This was the Vocal Augmentation and Manipulation Prosthesis, or VAMP, a a gesture-based wearable controller for live-time vocal performance. This controller was created for vocal performers in order to let the performer serve simultaneously as the conductor and the performer of a piece of solo vocal music, extending his or her voice purely through free gesture without touching buttons, dials, or a computer [35].

This instrument was originally inspired by my work on Tod Machover's upcoming opera, Death and the Powers. In the libretto for the opera, it is given that the character of Nicholas has a prosthetic arm that somehow makes him specially able to do things that others cannot; following my group's tradition of Hyperinstruments research, we chose to create an engaging wearable instrument that would take advantage of the performer's vocal training and make the character more musically able. Such an instrument must be constrained to the physical form of an arm and limited by the unknown instrumental experience of an opera singer. To accomplish these goals, we designed a wearable controller with the form factor of a glove that allowed the performer to manipulate his or her own voice through gesturing with that arm. Thus, much of the audience's focus remains on the sound of the performer's voice, a key component in an opera production.

Because of the opera context of this instrument, it is also necessary for the gestural mappings to be intuitive and clear for an audience that may not have significant experience with electronic music or pre-existing wearable instruments, with their sensor readings tied to specific sound effects. How could it be absolutely clear that the performer's gestures really affected the sound manipulation in performance? I felt that it was necessary to have fairly direct mappings between a specific vocabulary of gestures and specific sound manipulations, such that a gesture could naturally evoke the resulting change in the sound. In keeping with the use of this controller for vocal performance, I chose to develop expressive mappings of gesture to sound that were inspired by the gestural vocabulary of choral conducting, as well as by the core gestural metaphor of "grabbing" and extending a note by pinching two fingers together by the mouth. I began my work on this system by creating a vocabulary of desired gestures (including both continuous gestures and discrete gestures) and determining their associations with control over the voice, then developed a sensor system and programming framework to recognize those gestures and make the necessary sonic manipulations. Due to this initial focus on setting a vocabulary of strong mappings between a performer's movement and sound manipulation, the resulting instrument avoided naïve interactions, and instead had metaphorical and meaningful behavior. The construction, implementation, and theory behind VAMP will be discussed in more detail in the following sections.

3.1.1: System Construction

The base of VAMP is a soft, stretchable fabric glove which extends to the performer's shoulder. This glove is made in the shape and size of a given performer's arm, in order to obtain the most sensitive data about that performer's movement. I constructed the current version of the glove out of thick, stretchable velvet and sewed it by hand to fit my arm. By using fabric stretched and form-fitted to the arm, the glove can stay in place without using a potentially uncomfortable elastic band around the upper arm.

Illustration 21: The Vocal Augmentation and Manipulation Prosthesis

A series of sensors on the glove measure various aspects of the performer's gestural behavior. These specific sensors were chosen in order to recognize major movement aspects of the predetermined gestural vocabulary defined in the next section. Two 4.5" flex sensors are sewn onto the glove, one over the elbow joint and one over the wrist joint. When the sensors are used as variable resistors in a voltage divider construction, voltage measurements correlate to the amount of strain. The flex sensor at the elbow measures only the amount of unidirectional bend in the elbow, while the sensor at the wrist can detect the wrist bending either forward or backward from center (though these directions are not differentiated in the output). Stitches at both ends and over the middle of each sensor keep the sensors secure to the glove and limited to bending with the associated joints.

Second, the glove is outfitted with an accelerometer attached to the top of the forearm. This accelerometer is aligned to detect acceleration along the axis that a conductor moves his or her arm when s/he conducts a downbeat. Finally, there is a small 1 lb. pressure sensor attached to the index finger of the glove. This sensor is approximately the size of a fingertip, with a thin, non-sensitive flexible extension that is sewn down the middle of the palm.

The data from all the sensors on the glove is collected using an Arduino-compatible Funnel I/O (attached to the upper arm of the glove), and sent wirelessly over a serial connection using Xbee to a Macbook Pro running a Java applet. This Java program utilizes the Processing API and Processing's Arduino libraries to enable communication with the Funnel I/O. In the Java program, the sensor information is collected, analyzed, and mapped, and the desired sound modifications are calculated. Instructions for the desired modifications are then sent to a Max/MSP patch running on the same computer, using [39]'s *MaxLink* libraries for Java. The performer sings into a microphone, sending audio data that is amplified and modified in the Max patch. This allows all of the audio input, processing, and output to be done through Max 5.0, while the sensor input and calculations are carried out using Java and Processing.

3.1.2: Gestural Mappings for VAMP

The mappings between the performer's gesture and the sound modifications were primarily inspired by the movement vocabulary of choral conducting. The specific conducting actions used as the basis for this controller's gestural vocabulary included setting a tempo, controlling amplitude, and adding vocal parts. This vocabulary was also extended with more controller-specific (though still intuitive) actions, such as physically grabbing and releasing individual notes. All these mappings are computed in real time.

When the performer closes his or her thumb and forefinger, putting pressure on the glove's pressure sensor, the audio signal that is currently coming into the Max Patch is captured and "frozen." For instance, when the performer sings a note and touches his or her thumb and forefinger together, the current note is held and extended, regardless of other notes the performer sings, until the performer "releases" the note by separating his or her fingers. The pressure from the sensor is regarded as a binary input: pressure above a given level represents a held note, and pressure below that level represents a released note. This gesture is separated in the audience's experience and the performer's experience from the specific sensor readings used to identify it; "grabbing" a moment of sound creates a strong metaphor, where the voice appears tangible and shapeable through the way it is held and moved by the performer's hand.

Illustration 22: VAMP -- capturing a note

The implementation of the "frozen" note processing uses the Max pfft~ subpatch *solofreeze.pfft* designed by Jean-François Charles [16]. This subpatch uses Jitter matrices to do spectral processing on a Fast Fourier Transform of the audio signal, which allows not only for the necessary computation to be done in real time, but also for a richer sound quality by repeating multiple frames blended together in a stochastic process.

A second gesture implemented in VAMP is inspired by one of the primary tasks of a choral conductor's gestures: setting a tempo for a given choral work and instructing the performers follow that tempo. VAMP provides the ability to pulse a sustained note to a beat pattern indicated by the performer's gesture. Using the accelerometer data from the movement of the performer's forearm, the software constantly examines data patterns over time and locates peaks in the data, which represent downbeats. When two consecutive peaks are detected less than two seconds apart, the length of time between those peaks is set as the beat length (the current tempo), and the program goes into "beating mode." All peaks detected at approximately one beat length apart afterwards trigger amplitude modifications of the sustained note; the amplitude is set to the current high level at each detected downbeat, then fades out after half the calculated beat length. This makes the sound pulse in time with the performer's downbeats.

While the system detects that this "beating" is occurring, it recalculates the beat length with every downbeat and allows the performer a little flexibility in the exact timing of beats. This allows the performer to adjust the tempo and still have the system respond correctly to each downbeat. When the system does not see a beat when expected, it waits for half a second before turning off the "beating mode" and restoring the amplitude of the sound to the previous high level. As with the "grabbing a note" gesture, this gesture and its relationship to the desired sound manipulations were defined in the early conceptual stages of VAMP's development, with the specific sensors and gesture recognition only being determined afterwards in the implementation stages.

Illustration 23: Vamp -- crescendoing a held note.

Additionally, this system allows the performer control over the amplitude of the note s/he is sustaining through gestures indicating crescendos and decrescendos. For a crescendo, the performer extends her arm and reaches out her hand; for a decrescendo, the performer pulls back her hand to near her body. Analysis of the sensor data from the glove indicated that these gestures are primarily characterized by the degree to which the arm is bent at the elbow. Thus, the amount of bend detected by the sensor on the elbow is mapped to the amplitude of the sustained pitch. The range of amplitude of this effect was empirically determined to allow the performer the greatest expressivity in volume without disappearance or significant distortion of the sustained sound.

In keeping with the choral style explored in this controller, the final effect that the performer can control through this system is the addition of another sustained note in harmony with the one that the performer is holding. The fundamental frequency of a held note is calculated with the *fiddle* external for Max, developed by Miller Puckette [19]. Given this fundamental frequency, any harmony *n* semitones above the fundamental can be calculated in 12-tone equal temperament using the equation $F_{harmony}=F_{fundamental}*(\sqrt[12]{2})^n$ This harmonic frequency is calculated in Max from the fundamental frequencies of any "captured" note. Then, by subtracting the fundamental frequency from the harmonic frequency, we can determine the amount by which the sustained signal needs to be shifted by Max's *freqshift* object. By the performer raising his or her wrist, s/he can bring in and adjust the amplitude of this harmonizing note. The harmony was chosen to be a perfect fifth above the held note. When the performer squeezes his hand tighter when a harmony note is present, the harmony compresses to a perfect fourth.

Originally, this gesture was mapped to the amount of frequency shift of the held note. However, the accuracy with which a performer can manipulate this sensor does not give the performer precise control over pitch. As [79] states, "... mapping of the output of a sensor that is precise but not accurate to a variable controlling loudness may be satisfactory, but if it is used to control pitch, its inaccuracy will probably be more noticeable." Instead, the particular harmony shift is set in the Java program and the performer controls the harmony entrance and volume. This is one example where the mapping between gesture and sound needed to be adjusted empirically during the implementation process.

3.1.3: Development of a Gestural Vocabulary

While developing VAMP, I began with a particular vocabulary of gestures that interested me and some expressive ways to connect these gestures to sound manipulation: capturing a note with pinched fingers, extending the hand to control volume, raising the hand to add a harmony note, squeezing the hand tighter to change the harmony, shaking the wrist to add vibrato and overtones, and beating the arm to pulse the note. These basics of this gestural vocabulary, inspired by choral conducting, were the first thing developed for VAMP, before any actual physical item, any sensor systems, or any code existed. The sensors on the glove were designed to be able to recognize this gestural language, and the code was written for the desired results. The gestures and their mappings were conceived of separately from, and prior to, the technology used to implement those mappings.

This separation of the desired gestural vocabulary and mappings from the technology needed to implement them resulted in a system that had some very intuitive and expressive interactions between movement and music, with the associations not limited by a specific set of sensors. When the gesture and the sonic result were closely coupled in metaphorically or emotionally resonant mappings, as when pinching the fingers together captures a note or when stretching out the arm creates a crescendo, the resulting interaction proved compelling and interesting. Even when the actual gesture recognition process was simplified into resulting particular sensor values (such as the crescendo/decrescendo being controlled by the amount of bend in the performer's elbow), the meaning and content of the mappings were still created by thinking at a higher level about the gesture. This was an important lesson for me in the way that strong mappings between high-level gestural descriptions (described as reaching out an arm, vs. as increasing voltage values from the elbow bend sensor) could be used to create almost magical interactions. However, the implementation of this specific gestural vocabulary was still tied to particular sensors' values. Thus, modifying mappings significantly or increasing the gestural vocabulary proved harder to envision or implement after the physical object and its associated streams of data existed. I felt the need for a system that would allow me to retain that higher level of thinking about gesture and movement throughout a iterative development process, rather than being caught in the particulars of sensor data while exploring new mapping systems.

3.2: Disembodied Performance

Another early project that inspired my work on the Gestural Media Framework was my experience developing wearable sensor systems for the Disembodied Performance System, designed by Peter Torpey [73, 74]. This system, created for Tod Machover's opera *Death and the Powers*, addresses a variety of questions about how to map a performance from one expressive modality, the human body, to a variety of other modalities including non-anthropomorphic visuals, lighting, movement, and sound. In the story of the opera, a rich and powerful businessman, Simon Powers, nears the end of his life and seeks to extend his mind, emotions, and influence by uploading his consciousness into a computer system he has designed and integrated throughout his house. Power's transformation from human being into the pervasive System occurs at the end of the first scene in the opera. Theatrically, this means that the

Illustration 24: Performance sensors on James Maddalena

(Photo: Tod Machover)

character of Simon Powers shifts from being portrayed by a live opera singer, James Maddalena, to being embodied by the theatrical set and the entire performance space. The stage must breathe, react, be emotionally expressive, and be as compelling as a human performer [49]. It would be possible to have pre-recorded Maddalena's voice and have the behavior of the set and visuals on the stage be pre-scripted and triggered for separate scenes; however, we felt that it this would be constraining to the other performers and the orchestra and not especially expressive or conducive to the story or the performance. We determined it was a theatrical necessity to keep the power and presence of Maddalena's live performance, even if the performer were not physically on the stage. Therefore, in our approach, the behavior of the scenic elements, including lighting, visuals, and robotics, are influenced in real time by Maddalena's live performance. His gestures, his breath, his voice are observed and used to shape the output media on the stage in expressive and active ways.

3.2.1: Affective Model for Abstractions in Mapping

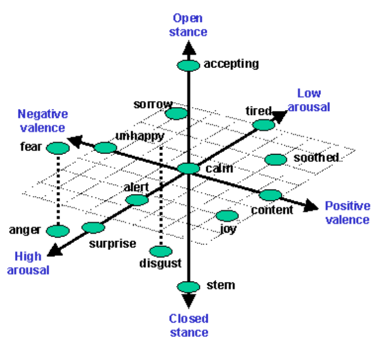


Illustration 25: Three-Dimensional Affect Model.

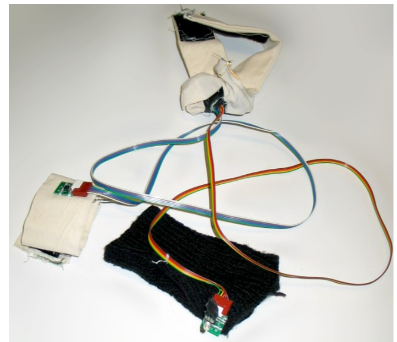
Axes of the model are Valence, Stance, Arousal [11].

This Disembodied Performance system takes the perspective that simply mapping sensors and data streams directly to output channels is not the most effective mode of creating compelling connections between performer and media. As mentioned in the prior chapter, a common mapping principle is to tie the variation in one input parameter (a particular data stream from a particular sensor) to the variation in a particular output parameter. With the Disembodied Performance system, we explored what could happen if these mappings were made through higher-level abstractions instead. All of the live sensor data streams from physiological sensors, movement sensors, and vocal analysis are abstracted into a model of the character's emotional state. This three-dimension model, frequently used for characterization of affective states, measures a character's location on the axes of valence (pleasure to displeasure), stance (engagement to disengagement), and arousal (active to inactive).

All sensor data streams, as well as data from the voice and the performer's location in space, are therefore mapped into the three-dimensional space of this affective model. With the emotional state of the character determined, the output modalities can be mapped to elements of the character's current emotional state and the character's trajectory through that emotional state. Thus, the actual performance capture inputs are not significant in designing the behavior of the outputs. However, it was necessary to design the sensor system to capture affective qualities of the performance, not simply movement information.

3.2.2: Sensor Systems for Performance Capture

I was responsible for designing and creating the wearable systems for the performer, seeking to measure some vital and expressive features of Maddalena's physicality in performance. One of the key aspects of this physical presence is the performer's breath – the breath delivers information about phrasing, emotion, and life that would be evident to audiences watching the performer live on stage. Therefore, part of the sensor system includes a flexible band around the performer's chest that detects his inhalations and exhalations. The fabric band contains a stretch sensor located in a region of elastic fabric at the performer's back. As the performer inhales, his chest expands and therefore stretches the elastic region and the sensor. The amount of stretch is tracked by a Funnel I/O microcontroller board and transmitted wirelessly via the Xbee protocol. This simple sensor was found to detect information about the breath of the performer and his vocal phrasing that was more detailed than the information obtainable from audio or the score.



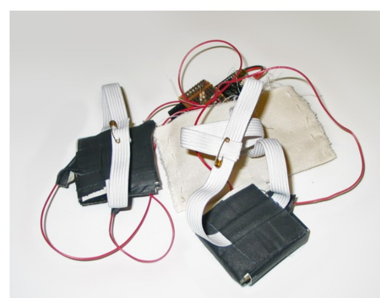


Illustration 26: Disembodied Performance Sensors.

From top to bottom: gesture sensor, breath sensor, foot pressure sensor.

Additionally, accelerometers on upper arms, forearms, and the backs of the hands were added to obtain information about Maddalena's gestures as he sings. We determined that it was not necessary to capture specific gestures from Maddalena; more important was the overall character and expressive quality of his natural motion while singing with emotion. Thus, we generally did not use the raw accelerometer data directly to affect the character's location in the affective space, instead looking at features of the movement such as its rugosity (amount of variation in the data over time), its amplitude, and its rate of change. These higher-level parameters drew on features of accelerometer data related to the quality of the movement (sharply changing, smooth, sudden, etc.). Such parameters can also be related to aspects of Laban's qualities of movement, as will be discussed later in this document.

The other feature of his movement that was seen as indicative of the character's emotional content was the way that Maddalena shifted his weight from side to side and back and forth. Pressure sensors on the soles of the performer's shoes capture this level of movement. All accelerometer and pressure data is similarly captured with Funnel I/O boards and transmitted wirelessly via Xbee. Vocal data from the performer is also collected using microphones and sent to the computer for audio processing. This vocal data, including both sung and spoken sounds, is analyzed for such audio parameters as amplitude, pitch, timbre, and purity of sound (consonance). These values can then be used as inputs to mappings.

The need to obtain higher-level information from the performance was quite influential in my work designing what sensors should be used in this system. I examined how those sensors could convey meaningful information about the performer's movement, and what sort of physical information would carry emotional content. Then, the movement was abstracted away from specific sensor values into a higher-level framework that could be used to more intuitively shape relationships between the input performance and the output media. Additionally, this project focused less on a specific movement vocabulary and more on the expressive details of the movement's quality. This way of thinking about and using movement was highly significant to me as I began developing a Gestural Media Framework.

3.3: Gesture Glove

From these earlier projects, I had found the need for expressive mappings with strong and resonant gestural vocabularies, the importance of the qualities of movement, and the limitations of existing mapping systems for inspiring creative, metaphorical interactions between gesture and movement. In exploring whether higher-level abstractions of gesture would help in the process of creating mappings, I began by implementing an initial proof-of-concept tool for gestural media manipulation, mapping a simple gestural vocabulary to parameters of interactive visual applications. For this proof-of-concept application, I developed the Gesture Glove. I wanted to design a glove that would recognize a simple vocabulary of hand movements, create encapsulated abstractions of those movements and dynamic information about those movements, and allow me to experiment with mapping such abstracted information to a variety of visual outputs. From VAMP, I knew that pre-planning specific mappings between media and gestures could lead to successful connections, but I wanted to see what would happen in experimenting with a particular gestural vocabulary and parameters of a range of applications, to discover and shape exciting mappings completely without thinking at the data level. What would happen if I created mappings to media from gestures or qualities of motion, instead of from raw or slightly-processed movement data? How would it change the process of coming up with, experimenting with, and changing these mappings?

3.3.1: System Design

Illustration 27: Gesture Glove with sensors and Funnel I/O board

For this system, the wearable input was a glove equipped with just two sensors, a 3-axis accelerometer and a bend sensor. Data streams from these sensors were collected by a Funnel I/O microcontroller and sent to a Java application using the Xbee wireless protocol. This Java application was hand-coded to recognize a small vocabulary of single-hand gestures and gestural parameters from the particular sensor data. For the purposes of this demonstration, the glove was designed to recognize squeezing the hand (and how tightly), flicking the hand (and how hard), waving the hand to the left or the right, holding the hand vertical, holding the hand horizontal, and how much the hand was tilted vertically and horizontally. Specific sensor patterns that correlated to each recognized gesture or parameter were determined empirically, and then could be used in Gesture Objects, encapsulations that hid the data and the specific sensor information and only revealed whether the gesture associated with that object was occurring and values of any quality parameters associated with that gesture. The goal of this project was to create a system where the user could think about the interaction in terms of flicking and squeezing and tilting, rather than in terms of bend sensor values and accelerometer values.

All of the sensor processing and gesture recognition was done in Java, which then sent that information to Gesture Objects created in Max/MSP. Each Gesture Object consists of a name, an output for whether the gesture is occurring or not occurring, and outputs for any quality parameters associated with that gestures (Illustration 28). These outputs of the Gesture Objects were able to be mapped in Max/MSP both to sound manipulation parameters and to parameters sent to freestanding Java-based visual applications programmed by Peter Torpey. Importantly, the gesture objects and their associated parameters were connected very simply to the output parameters, and these connections could be modified with sliders or multipliers or other controls adjustable on-the-fly. When one was creating and adapting mappings in Max/MSP, the actual parameters of the glove input input were completely hidden, and only the parameters of the specific gestures were revealed.

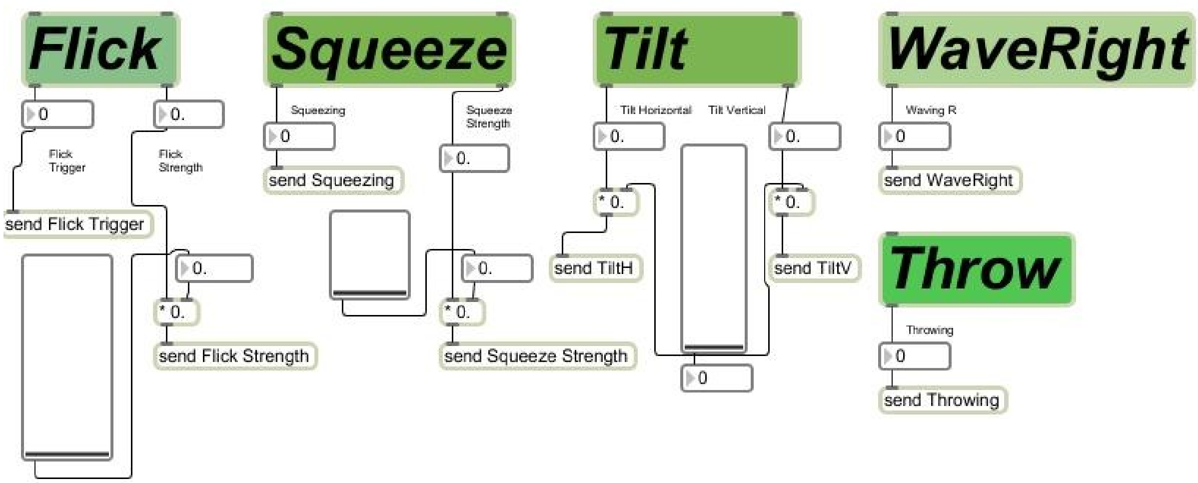


Illustration 28: Screenshot of Max/MSP patch incorporating Gesture Objects

3.3.2: Applications and Mappings

To explore the capabilities of this Gesture Glove, I designed interactions for three simple visual applications. The first application was a splatter painting program. A drop of "paint" on a black screen could be guided left to right by rolling the hand side to side, and could be moved up and down by tilting the hand up and down, with the paint moving faster and bigger the more the hand was tilted. Flicking the hand at the screen caused a paint splatter, whose size was determined by the strength of that hand flick. Additionally, the pant color

Illustration 29: Gesture Glove controlling visuals.

(Rendering by Peter Torpey)

changed randomly on a hand flick. Waving the hand to the right would clear the painting and let the user start over with a black screen. The range of the mapping between the hand's tilt and the speed of the paint movement, as well as the mapping between the flick strength and the splatter size, was adjustable by sliders in the Max patch. Thus, to make the effect of the hand tilt on the speed of the paint greater, one simply dragged a slider higher, without caring about how the hand tilt was calculated or how it was represented internally.

The second application was a fluid dynamics simulation, a graphical representation of a dark fluid with a colored dye injected into the fluid. The location and movement of this flowing color injection was controlled by the user's movements. To make a dye appear in the fluid, the user had to start squeezing his or her hand closed. Squeezing the hand more tightly changed the color of the dye, with a color range controllable through a slider. As in the painting program, the tilt of the hand controlled the direction of the color injection and the speed of its movement, with the effect scalable by sliders. Flicking the hand created a scattering of shining particles to appear in the simulation. The number of particles was again adjustable with a slider. I additionally designed a variation on this fluid dynamics simulation, a musical game. The user controlled a colored force in the fluid as before, but in this application there were also several colored circles that emitted arpeggios when the force passed through that circle. By aiming the fluid at these circles, the user could play simple melodies.

The third visual application was a slideshow application. Here, photographs were shown in a horizontal row. Waving the hand to the right slid the row of pictures over so that the photograph to the right of the center became the new center photograph. Waving the hand left slid the row of pictures the other way. Flicking the hand removed whichever picture was at the center from the slideshow. This application was a more standard human-computer interaction model, as it did not depend on the qualities of the user's gestures, simply on the fact that the user was performing particular gestures.

3.3.3: Results

With all of these visual applications, the Gesture Object construction made it very easy to quickly test and explore mappings between different gesture parameters and the various output parameters controlling the output programs. I demonstrated this Gesture Glove during Sponsor Week in October 2009 for a wide range of visitors to the Media Lab, as well as members of the Lab community. I showed visitors a selection of the programs and the basic gestural vocabulary for each, then let them experiment with wearing the glove and exploring the associations between their gestures and the visual models.

The overall user reaction was quite positive, both about the clarity and intuitive nature of the mappings that could be determined by thinking at a gestural level and about the possibilities for future development of the work. By having the gestural abstractions made visible in Gesture Objects, it was clearer how the gestures mapped to manipulations of the visuals, and how those mappings could be adjusted. It seemed that the way that users thought about possible new uses of the glove was expanded by its having a gestural vocabulary, rather than simply a collection of sensors. With this proof-of-concept project completed, I set my sights on the larger goal: creating a generalized Gestural Media Framework.

Chapter 4: The Gestural Media Framework

4.1: Intentions and Goals

I began work on this framework with the core idea of wanting a system that would allow mappings between gesture and media to be developed using higher-level descriptions of movement, such as gestures and movement dynamics, rather than using basic sensor readings and manipulations of raw or processed sensor data. In examining examples of potentially over-simplistic mappings such as "the higher I raise my hand, the bluer the lights should get," I found that I could determine that such mappings were not useful for particular performances when I described them at this sort of semantic level, focusing on a particular gesture and the resulting media action; however, I realized that no existing software allowed me to program such a mapping in the way that I thought about or described it. There was a significant gap between the semantic, symbolic, and metaphorically-resonant level at which I thought mappings should be made and the naïve level at which the software systems I was using encouraged me to think about mappings. So how could I create a tool that would make it easier to think about the mappings between gesture and media, so as to encourage more compelling connections? How could I encourage metaphorical and semantic associations between the rich linguistic and communicative movement of a performer and particular media manipulations? Along with this issue, I had learned through my work with the Vocal Augmentation and Manipulation Prosthesis, Disembodied Performance, and the Gesture Glove that good mappings between gesture and media needed to be developed for individual applications, as what is "intuitive" or "compelling" could be different for every performance-maker, every performance context, and every piece. The same gesture can be mapped to outputs in many different compelling ways, creating amplifications or juxtapositions, leveraging or subverting its metaphorical, emotional, or semantic context. What approach to describing and encapsulating movement would be supportive of a performance-maker's individual artistry?

In asking why we need to access the high-level content of performance movement, we can draw a comparison to the needs of processing and working with spoken language. The meanings and images conjured up by a particular sentence come from the words, the tone of voice, the structure of the sentence, what has been said before and after, and the knowledge and context of the speaker and the listeners. What if you wanted to create a video mix to amplify the act of someone speaking? How much would you be limited in creating interesting and rich mappings to spoken language if all you had to work with was a raw audio signal? Even a filtered audio signal, detecting concepts like amplitude or frequency, would not capture or make it easy to work with the expressive content contained in the words, the tone of voice. Similarly, movement, particularly in the framework of expressive performance, contains much more interesting and meaningful content then streams of sensor data reveal.

Additionally, I felt that the standard mapping-sensor-parameter-to-output-parameter approach, while quite common, was more suited for technologists who wanted to make art than artists who wanted to incorporate technology. Yes, one could make successful and rich mappings by predetermining a gesture vocabulary and the associated media manipulations, choosing and implementing sensor systems to detect that selected vocabulary, and writing code to process that sensor data, as I did in my work with VAMP. However, that process required me to be a computer scientist and electronics engineer, not only a performance or interaction designer. In order for non-programmers to be able to create interesting mappings, it seemed to be necessary to abstract the meaningful movement information away from the specifics of sensors and raw or filtered data streams.

For this system, I also wanted to address a major limitation I had found in gesture recognition systems: the lack of information about the dynamics of the movement. In performance contexts, expressive movement is not only composed of what movements a performer is making, it is also inexorably tied to how she is performing those movements. It would be necessary to develop a way to analyze and quantify that notion of "how" movement was performed. This high-level recognition and description of movement quality would provide vital expressive information as well as with some continuous input parameters that could be used to control continuous output parameters.

My desired use of this system builds on an instrumental framework, allowing for expansion and augmentation of a live performer's movement into a variety of media dimensions. The particular connections between specific gestures and qualities of movement are to be shaped by a performance-maker, and have deterministic relationships between the performer's specific movement (thus making particular interaction sequences learnable and repeatable). While the technological output may serve as either a pure extension or an accompaniment to the live performer, it is still controlled solely by the performance information.

So, the ideal Gestural Media Framework should provide tools for continuous gesture recognition, as well as a way to describe, recognize, and quantify qualities of movement, informed by previous theories of movement. These gestures and movement qualities would be able to interact but would also be accessible separately, such that a specific gesture can be recognized and categorized when it is performed with any quality of movement. The system should perform these recognition steps and abstract this information away from a particular sensor setup into a general, encapsulated form that could be communicated to other systems. It should provide a structure that incorporates a variety of potential movement capture systems, media outputs, and mapping processes. It would provide mapping tools to quickly create relationships between rich media aspects of a performance and the abstracted gesture and quality of movement information. Such as system would allow and encourage us to think about the interactions between movement and manipulation of sound, visuals, or other output media at a level of movement encapsulation that allowed us to think about the semantics of gesture, and draw on our metaphorical associations with gesture. To augment an audience's interpretation of movement with media results. To describe movement at a level that has resonance, emotional and communicative content, and expressivity, and use those descriptions in creating mappings. To envision performances where thrusting a hand forward suddenly could shatter a soundscape, where a performer could reach out and grab a sound and pull it into her, where a caress could blur the edges of a video. To ask, "What should happen when I make this gesture? When I move in this way?" To play with developing these mappings in rehearsal, to create interactions that are tailored to a particular performance's expressive aims and reflect the creativity and style of the performance-maker. To create a space for metaphor, a space for magic, a technologically-rich performance where the technology fades into the background.

With all of these aspects of the system in mind, I developed, implemented, and experimented with a Gestural Media Framework. I created a performance work, *Four Asynchronicities on the Theme of Contact*, that incorporated this system, and continued to develop and refine the framework throughout the rehearsal process. The specific dance pieces and the use of the gestural system in *Four Asynchronicities* are primarily discussed in Chapter 5, though I discuss some ways in which the rehearsal process influenced my development of the Gestural Media Framework in this chapter.

4.2: System overview

The standard framework for working with movement in performance can be simplified to three layers (see Illustration). The first layer is the input layer, consisting of sensors and their associated data, perhaps with some processing or filtering of that data. The second layer is a mapping layer, where parameters of the input are correlated to parameters of the output. The final layer is the output layer, consisting of whatever programs and devices are needed to produce sound, video, or other media. My Gestural Media Framework consists of four layers. The first layer, as in the standard model, consists of gesture capture input devices, such as wearable sensor systems, computer vision systems, motion capture systems, and capacitive sensing systems. The next level is the unique feature of this model, an intermediate layer that recognizes and creates meaningful, high-level semantic abstractions of gestures and qualities of movement. The third layer is a mapping level at which the information about gestures and movement qualities can be transformed into other modalities. The final layer consists of the output devices producing media such as sound, music, visual imagery, lighting, or even robotic movement. Each of these layers and its implementation in this system will be described in more depth in the following sections.

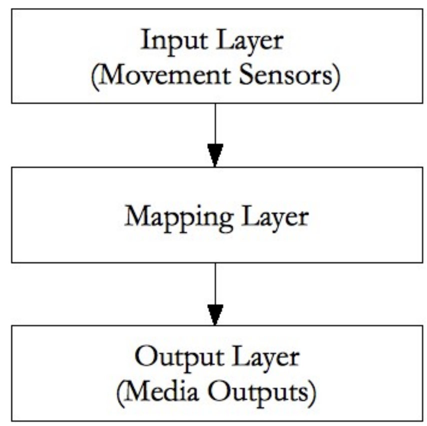


Illustration 30: Standard three-layer framework for interactive performance.

The key aspect of this approach that differs from preexisting work is the recognition and meaningful abstraction layer. Using machine learning and recognition algorithms, this stage transforms raw sensor data into information about gestures and movement qualities in a format that can then be sent to mapping layers and manipulated without concern for the specific input devices or recognition algorithms. In contrast, standard mapping systems such as Max/MSP or Isadora take in and map the low-level sensor data directly. Even a program such as EyesWeb, which attempts to perform some level of abstraction on its raw video input, does not push that abstraction all the way to recognizing specific gestures or intuitive qualities of motion, instead limiting itself to lower-level abstractions such as how much space the performer's body is currently occupying [13]. In my system, in order to perform this high-level abstraction, this

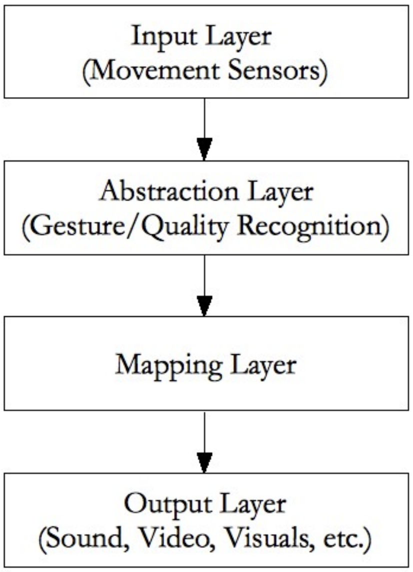


Illustration 31: The Gestural Media Framework for interactive performance.

layer employs machine learning for gesture recognition, as well as algorithms for recognizing continuous qualities of movement. These movement qualities are inspired by Laban's Effort theory, drawing on his research in the field of dance movement and his attempts to categorize and define a wide range of dynamic information about movement. A gestural vocabulary for recognition is selected by the user, and the system is trained to recognize that particular vocabulary. The resulting continuous information about the currently recognized gestures and qualities is then transformed into messages in the Open Sound Control format, an optimized protocol for communication between computers and a variety of multimedia devices. The resulting OSC messages can then be sent to a variety of mapping implementations [54]. Therefore, in the mapping layer, mappings are created using the abstracted gesture and dynamics information, with the specifics of the recognition engines and the input sensor systems removed.

4.3: The Input Layer – Sensing Systems

In any system that incorporates movement or gesture from a performer, it is necessary that the first system layer consist of some way of gathering information about the performer's movement. A variety of methods have been developed for sensing movement, varying depending on the type of movement information desired. These strategies include devices held by the performer [see 48, 10, 80], computer vision systems [13], laser or ultrasonic systems segmenting the stage into sections [20, 17], and wearable sensor systems [12, 17, 71]. Off-the-body sensing strategies are often used for detecting the performer's location in space, while on-the-body strategies are primarily used for detecting the motion of specific limbs. Hand-held devices for recording movement, while applicable in certain performance contexts such as musical performance, are often limiting in choreographic applications. While the Gestural Media Framework is applicable to all these kinds of inputs, in the current implementation I chose to work with wearable sensors due to their ease in capturing very specific gestural information from a performer, direction-independent gesture capture capabilities, and significant portability.

4.3.1: Wearable Systems

Important characteristics that drove the use of on-the-body sensors rather than a full computer vision system included the desire for gesture to be recognized with the performer at any angle or location, which would have been more difficult with a vision system that would be hampered by occlusion. Similarly, it was deemed important for the system to be very easy to use in a variety of rehearsal and performance situations in different spaces with a variety of controlled and uncontrolled lighting conditions, as well as for this system to be easily set up and calibrated in this range of situations. With a wearable sensor system, the Gestural Media Framework implementation could be incorporated into rehearsal wherever there was enough room to dance, without requiring the highly controlled lighting and background conditions necessary for computer vision systems. Additionally, wearable systems made the identification of individual performers a trivial problem, and made the system immediately extensible for a variably-sized group of performers. With vision systems, it would be challenging to detect individual performers without some kind of color identification or statistical modeling to predict performers' movement paths. For an example, what happens when two performers move into the same space in a video system? How can we predict which performer is which when they separate? They would need to be visually identifiable or the system would have to make an educated guess. This method becomes increasingly difficult with an increasing number of performers.

In the particular implementation of the GMF system that I used in creating *Four Asynchronicities*, the wearable input system is a set of sensor shirts, one for each performer. This system was designed to primarily recognize arm and upper-body motions. I had originally envisioned a full-body system with sensors on arms, legs, and torso, and a vocabulary of whole-body movement, but was guided by my committee to focus on a smaller-scale sensor and recognition system so as to limit the scope of the project and constrain the recognizable movement vocabularies within a smaller and more possible framework. Additionally, hand and arm motions more often have preexisting semantic connotations than foot or leg movements, so limiting my vocabulary to the arms would still give me meaningful and metaphorical abstractions to explore in mapping.

Each performer wore a close-fitted stretchable long-sleeve shirt with thin gloves sewn onto the end of each sleeve to create a continuous line between fingertips and shoulders. I selected white as the base color for these systems, such that they would be less noticeable when layered under costume pieces. I chose to build sensor systems that could be fitted under a variety of different costume options, rather than integrating sensors into the specific costume design for particular pieces. I knew that it was necessary to have the sensor systems incorporated into the rehearsal process from the beginning, likely before I would have finalized costume designs. Additionally, the wearable input systems are now reusable for a variety of performance contexts, provided that they are worn by performers of similar sizes.

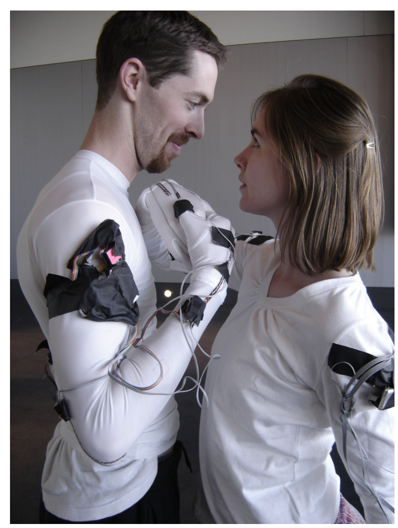


Illustration 32: Kevin Burchby and Lisa Smith rehearsing in sensor shirts.

Each shirt is equipped with a variety of sensors to detect the motion and position of the arms and torso, including flex sensors on each wrist, each elbow, and each index finger, as well as a three-axis accelerometer (an

ADXL335 chip) on each forearm. The location of each flex sensor was individually determined on each dancer to best cover the performer's joint. The data streams from these sensors go to a Funnel I/O Arduino-compatible microcontroller board and are sent using the Xbee protocol to the computer running the gesture recognition system. This set of sensors was found to detect a variety of interesting information about the movement of the upper body, including specific joint angles of major arm joints and overall directional acceleration of the mass of the arm, while still being relatively affordable and contained to a limited number of analog inputs. Each Funnel board accepts six analog inputs, so the sensors on each arm of the performer were routed to an individual Funnel board mounted on the upper arm of the performer.

All sensors and the associated wiring were sewn down to the outside of the close-fitting shirt, with some extra wire allowed to compensate for the stretch of the fabric when worn and during motion. Wire was used rather than conductive thread or other high-tech textile materials in order to limit the resistance of the connections and for increased durability. It was considered extremely important that the sensors have a fairly flat and low-profile form factor such that they would not be damaged through contact between performers or between performers and the ground. Sensors that could be easily harmed by pressure on the performer's body would be limiting both in the performers' range of movement or in the interactions that would be possible between pairs or groups of performers. This was accomplished with the majority of the sensor components, though the Funnel I/O boards mounted on the upper arms had a larger package than I would have liked. However, these boards had a number of useful characteristics that drove their use over connecting sensor inputs directly to Analog-to-Digital converters on the Xbee chip, including battery and voltage management circuitry, the built-in capability to recharge the battery via USB, and the ability to use more robust and reliable libraries for the remote serial communication.

Some challenges with using on-the-body sensors included developing secure, well-positioned and durable sensor systems that would output similar ranges of data every time they were used, and that would hold up to a range of highly physical and interactive choreographic sequences. In particular, it became necessary to reinforce all places where wires connected to sensors (especially to the bend sensors) so that strain on the wires caused by movement would not result in poor connections after repeated use. For this purpose, the Spectra Symbol flex sensors proved to be fairly functional, but had to be carefully attached such that repeated stresses would not cause a crack to develop in the unreinforced end of the sensors at the point where wires could be attached. Additionally, it was necessary to locate and attach the sensors so as to be comfortable to the performers and not restrict their movement. This was a primary justification in mounting the Funnel boards, with their fairly large form factor, on the outside of the performers' upper arms, where they would be fairly out of the way of most contact interactions between performers.

4.3.2: Other input possibilities

While I designed a specific sensor shirt system to capture movement input for my performance implementation, the Gestural Media Framework approach is flexible enough to adapt to a variety of systems that could be used for incorporating input data into this system. As long as the system is trained and set up to recognize gestures and movement qualities using a specific set of input systems, it should be able to use those input systems in the performance process. The information about currently occurring gestures and qualities of movement is stored in the system distinct from the sensor data. Mappings to outputs are completely separated from the specific input implementation used to gather movement data and from the machine learning techniques used to process that data into higher-level gestures.

For example, wearable sensor input systems other than the sensor shirt system previously described could all be used as input. Another popular form of movement input for performance is a computer vision system, though this kind of input system has the challenges with occlusion and uneven lighting described in the previous section. This input system is the basis of previously mentioned performance works such as Trisha Brown's *How long does the subject linger on the edge of the volume* [22], Camurri's performance works using the EyesWeb system [13, 15, 15], Palindrome's work with their EyeCon camera tracking system [21], and Sparacino's performances with DanceSpace [67]. In order for computer vision to be successfully used as input, it would likely be necessary to do some higher-level processing (such as figure extraction) on the video input. In order for an input system to provide a successful level of information to the system, it would have to track sufficient information about movement and changes of movement through time, which likely would require a notion of the physical form of a performer.

4.4: Abstraction Layer

The key component of the Gestural Media Framework that distinguishes it from other approaches for using expressive movement input to shape output media is the abstraction layer, where sensor data is turned into meaningful semantic information about specific gestures and qualities of movement. In this layer, it is necessary to achieve several steps: the definition of "gestures" and movement qualities, the recognition of those gestures and qualities, and the output of that information in a useable and flexible form. Here, gesture recognition engines driven by machine learning are used to process the sensor data into specific gesture information, and quality recognition engines process the sensor data into qualities of movement based on Laban's Effort theories. This gestural and quality information is then turned into Open Sound Control messages so that it can be sent to a variety of mapping systems.

4.4.1: Machine Learning and Gesture Recognition with Hidden Markov Models

In this framework, the first step of the abstraction process is the stage of gesture recognition. Sensor data, from whatever set of sensors is chosen, comes into the system, and that data must be parsed to identify particular gestures that are labeled as important. The vocabulary of those gestures and the labels that they are given in the system can be determined by a performance-maker. For the purposes of this implementation, I chose to incorporate machine learning into the gesture recognition stage, rather than calculate what data patterns were specific to a particular gesture through empirical exploration, as I had done in my work on the Vocal Augmentation and Manipulation Prosthesis. It was determined that this would give the system greater flexibility and the potential of larger gestural vocabularies.

It also is necessary to perform a kind of pattern recognition that would be independent of the length and specific performance details of the gesture. If the gesture to be recognized is squeezing the hand into a fist, it becomes highly limited in performance if that gesture must always be performed at an identical speed, over an identical timescale, with an identical orientation, and with an identical movement arc. The same basic movement over time should be recognized as a squeezing gesture, regardless of the character and speed of the movement. For purposes of my current implementation, I incorporated elements from Georgia Tech's Gesture and Activity Recognition Toolkit [81], as discussed in Section 4.6. This set of libraries provided tools to make it easier to use Hidden Markov Models in Java.

In this framework, I chose to perform gesture recognition on dynamic time-varying movement sequences rather than static pose recognition. Pose recognition has been previously explored in interactive dance contexts by some researchers, such as James et al. [34]. Their research used motion capture markers on a dancer's body to identify the location of all body parts at a given time, and labeled particular combinations of joint angles as a "gesture." However, we are developing a system for the context of expressive performance (particularly dance performance), with its wide range of potential movement patterns, rather than a limited movement vocabulary of human-computer interaction. A variety of different paths traversed by the body over time can pass through or end at the same physical shape. For example, in making a "stop" gesture with the right hand, the arm could start at the side and bend at the elbow as the hand comes up. Alternately, the arm could come up to the side, circle forward with the palm face down, then bend at the elbow and bring the hand up. I do not believe those paths should necessarily all be identified as one identical gesture/pose, nor should a choreographer have to avoid all alternative movement sequences that pass through a specified pose. In my performance-centered framework, a "gesture" is a specific sequence of changes in the body's physical position over time that can convey expressive, emotional, or communicative content, as defined in Chapter 2.

4.4.2: Learning a Vocabulary Versus Setting a Vocabulary

When I first envisioned the design of a Gestural Media Framework, I thought that I would specify a specific movement vocabulary of gestures that the system could recognize, some kind of "iconic" gestural vocabulary that would be usable for all sorts of performance applications. There have been previous attempts at creating movement primitives and standardized gestural vocabularies, such as Ari Benbasat's work with atomic gesture recognition using inertial measurement units [5]. However, these gesture recognition models either break gesture into fairly low-level units of limb movement, as Benbasat does, or into gestural vocabularies that are designed for particular use-cases. In order for me to have a system that would work for theatrical, dance, and musical performance, flexible enough to be used with potentially quite different kinds of movements in different artistic contexts, was it possible to develop a preset vocabulary that could serve a variety of needs? How could such preset gestures be identified so as to be intuitive to others?

As I continued working on the project, particularly as I started to think about choreographing pieces using this system, I determined that it would be a more productive and more freeing step to create a framework in which a user could specify his or her own important gestural vocabulary, labeled by whatever names were intuitive and clear to that user. Any basic vocabulary I dictated would likely not only be constrictive to anyone else using the system, but also would be constrictive to myself as I set about creating dance movement. If I wanted to be able to change and develop mappings on the fly, why should that creative process be limited by a preset gestural vocabulary? Therefore, I decided against creating a gestural vocabulary that was generic or iconic. In keeping with my original motivations of making a system that would be flexible and allow for the most adaptability of mappings in the rehearsal process, I determined that important gestures should be found during the process of creating movement, rather than specified before the process of creating movement. Metaphors and semantic associations with specific gestures would arise during the rehearsal process and the act of creating a particular piece, so I might first then be discovering the particular gestures that I wanted to recognize and use for media control. Therefore, the system needed to allow a user to include an individual gesture vocabulary and be able to add to and remove from that gestural vocabulary as desired.

For purposes of this implementation, creating a piece-specific gestural vocabulary was done by incorporating machine learning into the system and training the system on the set of gestures that one wanted to be recognized. This process required giving the system a number of labeled examples of a performer making each gesture, with the starting and ending points of the movement indicated by a secondary person observing the movement and operating the system. In the training program for this system, clicking the mouse marks the start of a gesture, and releasing the mouse marks the end of a gesture. All the sensor data and associated information between those two points in time is captured by the system and stored as an example of a particular type of gesture. The labeled name of a particular gesture is also set by the user, such that it can be intuitively recalled and used by that user. (The name I use for one type of gesture may not evoke the same movement for you.) In practice, we typically recorded between 8 and 15 examples of each relevant gesture, with a number of variations on each gesture (particularly tempo variations). This number of examples was found to provide acceptable material for recognition (approximately 80-90% of samples recognized correctly), with the system becoming more accurate at recognizing given gestures as it was provided with more training examples.

One limitation in the implementation of this gesture recognition was the choice to record data from both arms (the entire sensor system) in an example of a gesture. After experimenting with training samples consisting of data from only one arm, I found that the sensor data, especially the data from the three axes of the accelerometers, was not the same from the left arm performing a specific gesture to the right arm performing the same gesture. The recognition system did not perform sufficiently accurately when given one-arm gestural examples from each side. Therefore, I made the decision to record data from both arms for each gesture and have a gesture with the right hand be recorded separately from, and with a different identifying label than, the "same" gesture with the left hand.

Another choice in the gestural recognition process during this thesis was to record a separate movement library for each performer. At the moment, the recognition training is tied to examples from a single body wearing a single sensor system, rather than seeking some way to recognize a generic gesture as it could be performed by any person. It is easily possible to train the system on gestures performed by a variety of people to gain some greater generality in the recognition process. It would also be possible to easily change the details of the wearable sensors or other sensor systems and retain the same gesture recognition process, with samples of sensor data between two points in time being stored as an example of a particular gesture.

4.4.3: Qualities of Movement

Another very important aspect of this approach's Abstraction Level is the extraction and encapsulation of the quality of a performer's movement. While pure gesture recognition systems are becoming increasingly popular and useful in the field of human-computer interaction, there are significant aspects of these systems that limit their use in expressive contexts such as systems for musical or theatrical performance. In particular, these systems tend to recognize discrete events or sequences of discrete events, while expressive musical performances generally require some parameters of continuous control. For example, the movements of playing the piano cannot be fully described by a sequence of specific "gestures" (say, the performer touching particular keys) recognized at specific points in time. This description would lose the strength with which the performer strikes the keys, the specific arc of the movement, how long the note is sustained, and a number of other continuous and dynamic elements required for an expressive and compelling performance. Current gesture recognition systems lack not only the knowledge of *how* a gesture is performed, but also a design paradigm that believes *how* that gesture is performed is as relevant as *what* gesture is performed. In this framework, both the specific gestures that are executed and the manner in which the performer moves are relevant, and are useable together or independently.

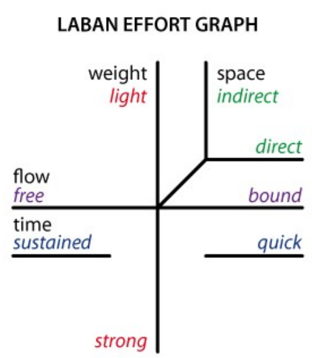


Illustration 33: Laban's 4-axis theory of Effort.

This model uses the axes of Time, Weight, Space, and Flow. (Image from [18]).

For my Gestural Media Framework, I decided to explore gestural qualities related to those analyzed and described in the field of dance by Rudolf Laban. In particular, I was inspired by Laban's description of *Effort*, the dynamics of movement. Laban held that the quality of any movement could be viewed as a point in a four-dimensional space, with the four axes of *Time*, *Weight*, *Space*, and *Flow*. In examining Laban's descriptions of these qualities, I believed that they could be meaningfully related to incoming sensor data and could also form a useful higher-level description of movement quality that could then be used intuitively when mapping that quality data to media manipulations. Thus, in the implementation of this framework, I map sensor data onto four quality axes, each normalized to a range of -1.0 to 1.0. At any point in a performance, a given performer's dynamics of movement correspond to a particular point on each axis. The quality recognition system implemented is dependent on the particular sensor system used with this implementation, though the theoretical foundation can be used to describe qualities of movement with a variety of other input systems.

The Time axis describes the speed at which a particular movement is being performed, from very fast and sudden to very slow and sustained. According to Laban "sudden" consists of movements at a quick speed and the sensation of short length of time ("a feel of momentariness"), while "sustained" consists of movements at a slow speed and the sensation of a long length of time ("a feel of endlessness") [40, pp 73]. This quality of movement is generally the sole quality incorporated into other movement description systems, with their fixed descriptions of the length of movement. For implementation of this axis in my sensor system, I saw that, intuitively, the "speed" of a motion is a measurement related to how quickly the body is changing its position and/or orientation. Thus, with the implemented sensors system measuring acceleration and the bending of joints, speed can be seen as the overall rate of change of the sensor data over a short time window. This was calculated by summing the amount of change in each sensor's data over each of the last four time steps, with these rates of change empirically weighted by the type of sensor to equally emphasize changes in joints and changes in acceleration. This sum's upper bound was determined empirically. The lower bound is 0.0, recorded when the body is still. This range was mapped to the range -1.0 (moving very quickly) to 1.0 (not moving/moving very slowly).

In Laban's system, the Weight axis describes movement on a scale from firm to gentle. Firm movements are forceful, strong, resisting, heavy; gentle movements are relaxed, unresisting, light, weightless [40, pp 73]. This quality, while intuitive, seemed difficult at first to connect with analog sensor data. However, while further exploring Laban's definition of Weight, I came across a useful simplification: Weight is a measurement of how much energy is being put into the movement. The concept of energy can be more directly linked to the specific sensor data in this application, as acceleration is related to the amount of energy put into the movement. Thus, by examining the total acceleration across both arms of the performer on all axes of the accelerometers, we can get a sense of how much energy the performer is using. A range of this acceleration sum was empirically determined between very energetic movement and still, gentle movement; this range was mapped onto the range -1.0 (intense, energetic, heavy movement) and 1.0 (light, low-energy, gentle movement).

The third quality that Laban discusses is that of Space, which explores the way in which a movement travels through the space around the body, whether it moves directly or indirectly from one point to the next. Movement ranges on this axis from direct (moving in a straight line) to flexible (moving in curved, varying lines) [40, pp 73]. For the purposes of this implementation version, since I was using only on-the-body sensors and no fixed external reference points in the performance area, Space was not implemented as a quality of movement recognized by the system. Given any means of getting an external frame of reference between the body and the space, such as a computer vision system, it would be fairly straightforward to implement such a quality measurement. For the purposes of the framework, the range of movement through the space would be mapped from -1.0 (very direct movement from one point to the next) to 1.0 (very indirect movement). For this measurement, it would also be necessary to have a sense of the beginning and end of a gesture as two fixed points between which the quality of the movement could be calculated, or some time range over which a gesture could be examined.

The final quality of motion discussed in Laban's Theory of Effort is Flow. Flow is primarily descriptive of the amount of freedom of energy in a particular movement, reflecting how smoothly and continuously the movement is changing. This quality is on an axis from "fluid" movement to "bound" movement. If the movement is changing smoothly and evenly, continuously, and uninterrupted, the movement is considered to be "fluid"; if the movement changes in stops and starts, jerkily and unevenly, the movement is considered to be "bound"[40]. Bound movement can be stopped at any moment, while fluid movement is hard to stop suddenly. For this implementation of a quality recognition engine, the system kept track of the amount of change in the movement (change over all sensor values) over time and incremented or decremented the previously calculated value for Flow by a small amount related to whether the current movement was changing quickly or slowly. Thus, at any point in time, the value of Flow (from -1.0, bound, to 1.0, fluid) reflects the overall trend in the change of the motion. The more fluid the movement, the more smoothly and continuously it has been changing, and the more bound the movement, the more harshly it has been changing. Thus, this quality axis contains a kind of hysteresis, as its present value is shaped not only by immediately recent sensor values but also by previous sensor values over a range of time. The fact this framework calculates both Time and Flow based on the amount of change in the movement is consistent with Laban's statement that "Flow cannot be imagined without a movement evolved in time. It is in this sense that the two motion factors Flow and Time belong together" [40, pp 172]. In my implementation, Weight and Time are qualities that can change rapidly within the window of a few sensor readings, reflecting the immediate state of the performer's body, while Flow changes more slowly over the course of a gesture or a sequence of gestures.

This set of movement dynamics, and combinations of these qualities, is found to capture a wide range of expressive movement. Laban describes eight basic "actions," such as thrusting, dabbing, flicking, and gliding, through various combinations of the efforts of Time, Space, and Weight. All the basic actions can be moderated and changed through the addition of different Flow qualities, as well as through different intensities or emphases. Thus, I felt that this set of movement qualities would be able to capture a large amount of information about how movements are performed while requiring a relatively limited number of abstraction parameters.

4.5: Mapping Layer

Once the input data has been analyzed and abstracted to Open Sound Control messages about gestures and qualities of movement (using the format I describe in section 4.7.1), that information is sent to the Mapping Layer. This layer consists of a variety of systems that take in the abstracted gestural information and provide tools for connecting the parameters of those inputs to output parameters in the desired output systems, regardless of the output media format. These systems can consist of any existing mapping software that accepts OSC input, such as Max/MSP or Isadora. Additionally, specialized mapping systems can be designed in Java to make the desired connections between the OSC gestural information and outputs.

In the ideal Mapping Layer, it is necessary to have representations of the gestures and qualities defined by the Abstraction Layer, operations that can be performed on those input parameters and other possible input parameters (including preprogrammed inputs, user-controlled inputs, and some representation of the output parameters). For the system to be useful in the rehearsal process, it is also necessary for any implementation of the Mapping Layer to provide significant flexibility and ease of changing mappings, preferably in real time without needing to recompile mapping information and restart mapping programs. An ideal mapping system used in this layer would be a visual environment with evocative representations of particular gestures and the quality of motion axes, where this information could be connected to a variety of high-level output parameters in more or less complex and sophisticated ways. Importantly, the mapping layer needs to maintain a clear sense of the relationships between encapsulated gestural information and the output media parameters, such that these relationships can be easily experimented with and explored in rehearsal.

For the implementation of the Gestural Media Framework that I used in *Four Asynchronicities*, I primarily used Max/MSP as the mapping system, as that provided the necessary flexibility and speed of developing and modifying mappings, as well as sophisticated control over sound creation, playback, and modification. I was thus able to incorporate my concept of Gesture Objects, Max/MSP objects with information about and parameters of a specific gesture, that I developed during the Gesture Glove project (see section 3.3). Additionally, I needed to add Quality Objects to the model, representing information about a particular performer's qualities of movement. In addition to the mapping steps done in Max/MSP, I wrote other Java programs to handle tasks such as mapping gestural information to OSC messages that were sent to control the stage lighting, as well as programs that mapped gestural information to parameters of computer-generated visuals.

4.6: Output Layer

The Mapping Layer then sends output parameters to the Output Layer, which interprets those parameters into control information for the media associated with the system. Thus, in this framework, an Output System is any media output that can be controlled by a Mapping System. It is also possible for Output Systems to send status information back to the Mapping Systems to be incorporated into the mapping algorithms. Possible Output Systems include sound generators, musical playback and manipulation systems, stage lighting control systems, visual generators, video control systems, and robotics control systems. All of these systems have been used as output mechanisms in a variety of performance contexts, so it is necessary for the Output Layer to be able to incorporate all and any of these systems. As these systems are separated from the input systems, it is only necessary for the communication format required by an Output System to be compatible with a usable Mapping System. This could take place over OSC, MIDI, serial, or any other protocol that can be generated by a Mapping System.

The current system implements mappings to several of these Output Systems, including sound and music control software (such as Max/MSP and Ableton Live), software to generate visuals (written primarily by Peter Torpey), and lighting control software. These output systems have a variety of control parameters, which can be set in particular relationships to the input gestural parameters.

4.7: Current Software Implementation

For purposes of this thesis, I implemented a version of this framework that I then used for my performance work. The abstraction layers of this software implementation were developed in Java, running on a MacBook Pro. The mapping systems that were used for the implementation included specialized Java applications and Max/MSP, also running on the same MacBook Pro. Output systems for sound were also located on the same machine. A Java-based output system for generating visual projections was located on a second Mac computer, and a program for lighting control was located on a Sony Vaio laptop running Windows. All programs communicated over Open Sound Control.

4.7.1: System Design

In the software implementation of this system, it was very important to maintain the framework's abstraction layers between the specific raw sensors/sensor data and the output systems. (See Illustration 34.) The main class in the implementation is the PerformanceSystem, which keeps track of the current performers in the system (represented by Performer classes) and the current outputs to particular mapping systems (represented by AbstractionOutputSystem classes), both of which can be added and removed freely. Additionally, the PerformanceSystem allows the outputs to query the performers about particular gestures and qualities of movement without the AbstractionOutputSystems needing to know about the Performers or vice versa. When an AbstractionOutputSystem polls the PerformanceSystem about whether a particular gesture is currently occurring, the PerformanceSystem can query all Performer classes whether they are performing that gesture and return the results to the AbstractionOutputSystem. Similarly, if the AbstractionOutputSystem queries whether a particular Performer is executing a given gesture, that information also is handled through the PerformanceSystem.

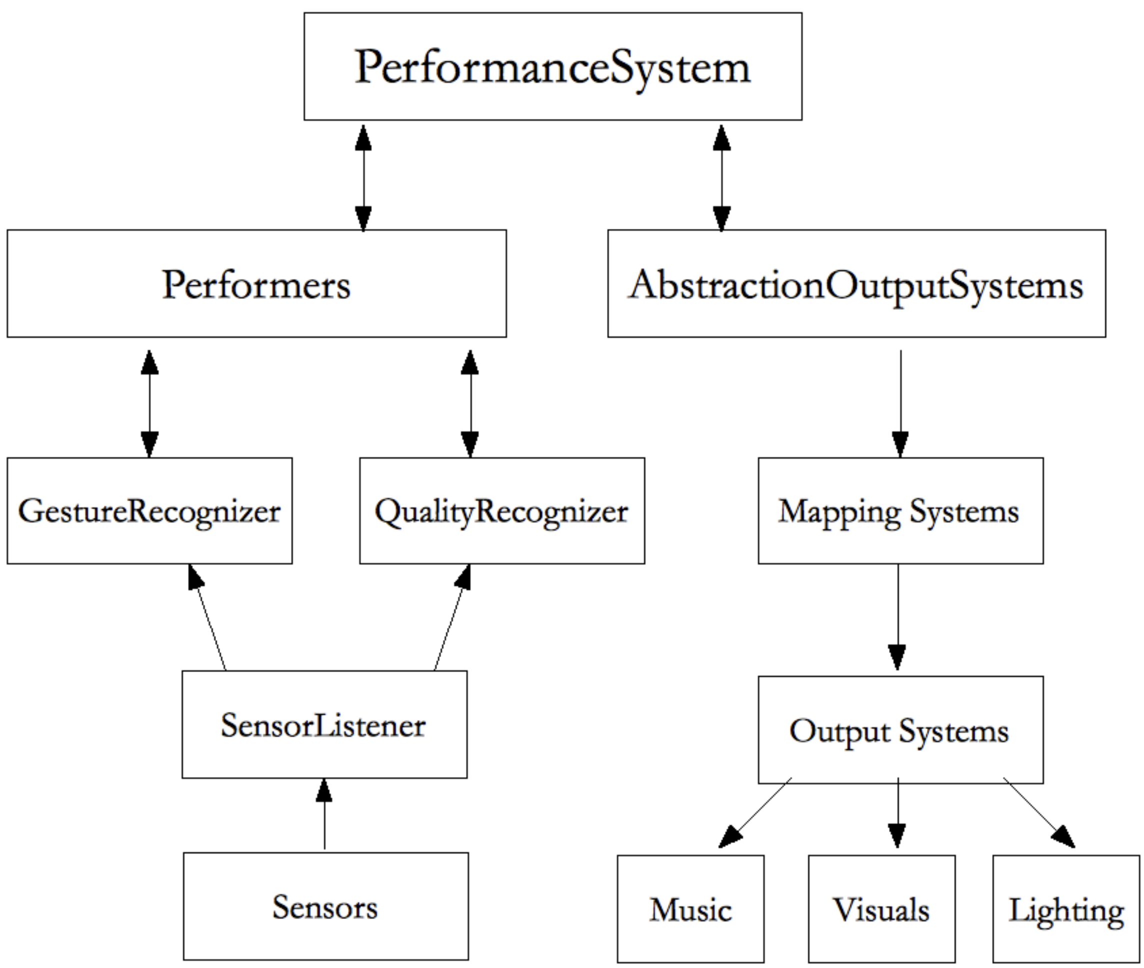


Illustration 34: System Flow Diagram for the Gestural Media Framework implementation.

Each Performer class incorporates a set of Sensor classes, which are classes that take in data about the performer. In this application, that data was information from the sensor shirt system previously described, but it could also be data from other on-the-body sensor systems, computer vision systems, or other movement input systems. The Performer class also contains a SensorListener class, a GestureRecognizer class, and a QualityRecognizer class. A Performer is labeled with a particular name that is known to the PerformanceSystem so it can be added and removed by name. Additionally, each performer stores a table of Gesture classes, each representing a particular gesture that is important for this performer, and a model of that performer's location in the four-dimensional quality space. These Gesture classes, which are continuously updated by the GestureRecognizer class, contain information about whether the associated gesture is currently occurring and any other parameters associated with that gesture.

The Sensor class takes in data from its associated sensor inputs and stores a history of that data. In the particular implementation developed for *Four Asynchronicities*, the Sensor class, given the Xbee addresses that connect to the Xbees on an individual performer's costume, stores the most recent 10,000 values from each data stream – twelve total streams. Each Sensor class also formats this data and some metadata (the average value for each stream over the last several timesteps, the difference between the current value and the last value, etc.) into the necessary vector format for the gesture recognition libraries. The SensorListener class monitors the data vectors formatted by the Sensor class, stores a number of these vectors in Samples, and decides when to send off particular sensors to the Gesture Recognizer for processing by Hidden Markov Models, as described in the next section.

All sensor data is also examined by a QualityRecognizer class at a rate of 20 times per second. This class calculates the current location of a performer in the four-dimensional quality space described in Section 4.4.3 and notifies the performer to store this location. This point in the quality space can then accessed when the performer is questioned by output systems and can be used together with or separate from the gesture recognition data. The implementation of the recognition algorithms in the QualityRecognizer class is specific to this application's sensor data. For the sensor shirts used in this implementation, the recognition algorithms for the qualities of *Time, Weight,* and *Flow* are described in Section 4.4.3.

In this implementation, an AbstractionOutputSystem queries the main PerformanceSystem about all of the gesture and quality of movement information for each performer. The PerformanceSystem in turn queries each Performer class, which responds based on the information it has stored in its list of Gesture classes and its stored point on the quality axes. The AbstractionOutputSystem then formats this data into valid Open Sound Control messages that are then sent to all ports that are listening for data from the performance system. This OSC data for each individual gesture the system recognizes has the address /PerformerName/GestureName, where PerformerName and GestureName are both variables replaced with a string identifier for a particular performer in the system and a name of the gesture. This message takes a value of 1 if that gesture is currently being performed by the specified performer, and a value of 0 if the gesture is not currently occurring. For the qualities of motion, the OSC message has an address of PerformerName/Qualities, and an argument that is the performer's current position on each of the four quality axes (scaled between -1.0 and 1.0). These OSC messages can be used as input to any program for mapping and media output control that accepts OSC input, including other Java programs, Max MSP, and Isadora.

4.7.2: GART toolkit

For the gesture recognition portion of this implementation, I incorporated portions of Georgia Tech's Gesture and Activity Recognition Toolkit (GART), mentioned in Section 2.3.2, a set of Java libraries designed to allow programmers to easily incorporate gesture recognition using Hidden Markov Models [81]. These libraries serve as a wrapper for Cambridge University's scripting libraries, HTK [33, 85], which allow the user to interact with a Hidden Markov Model system developed for speech recognition. The developers of GART took advantage of the similarity between the needs of gesture recognition systems and the needs of speech recognition systems: both require the classification of variable-length input signals that change over time. Both gestures and words can have inputs classified as "the same" despite those inputs taking variable amounts of time to complete. A recognition system for either modality thus cannot depend on algorithms that expect or require fixed-length input sequences, such as decision tree classification schemes. Hidden Markov Models, with their chain of internal states and transition probabilities, do not depend on the specific length of an input system, and thus allow for not only different gestures of varying lengths but also examples of gestures that take varying lengths of time. Additionally, both gestures and speech fit the theoretical basis of Hidden Markov Models. The data that is being examined (sensor data in the first case, sound waves in the second) is output from an underlying state (a particular gesture, a particular word). While the computer can observe the sensor data, it cannot observe the state directly.

Using the GART libraries, one can develop code to record libraries of gesture data and then use those libraries to train Hidden Markov Models to recognize new examples. Each library entry, a Sample, consists of a label identifying what gesture it represents; a series of vectors, each of which holds all sensor values and other current data at one time step; and the length of that Sample (i.e. the number of vectors in the sample). Each library can then be used to train Hidden Markov Models, with one model being trained for each type of gesture in the library. Once a model is trained, it can be passed a new Sample (an observed sequence) and returns the probability that this model generated that observed sequence. If this probability is calculated for each model in the system, the model with the highest probability can then be determined. Thus, if a model has the highest probability of producing a particular Sample, the gesture associated with that model is returned as the "recognized" gesture, with an accuracy level related to how probable it is that this is accurate.

4.7.3: Recognizing Gestures from Streams of Movement

One significant difficulty I encountered with the gesture recognition portion of this project was how to pick out a gesture from a continuous stream of movement data. In typical HCI gesture recognition projects, any movement of the user has a delineated beginning and end, and is supposed to be one of the gestures that the system should recognize. Therefore, the problem is simply one of recognition: given a sequence of data, what gesture in the vocabulary is most similar? However, in performance situations, there is likely to be a large amount of movement that is not significant, in that the performer is not executing a gesture in the desired vocabulary. Additionally, there is no clear way in which those gestures can be segmented; there is not necessarily a rest state in between gestures of interest or between a gesture of interest and other choreographic movement. In other research on segmentation of movement, the movement data must be broken into gestures by detecting the beginning and end of any physical activity using acceleration or EEG thresholds [5], labeling the start and stop of gestures through an external trigger such as a button pressed by the user or an observer [81], or else segmented manually (and not in real time) by human annotation [81]. None of these approaches seemed particularly fitting for a performance situation.

Billon et al. [6] developed a gesture recognition technique that is intended to be used in the theater, incorporating Principal Component Analysis and intelligent agents. Each agent is programmed to recognize a specific gesture, then watches a continuous data stream mapped into a two-dimensional vector space, where each pose corresponds to a point in this two-dimensional space and each gesture corresponds to a particular curve through that space. Each agent attempts to pick out occurrences of its own gesture. This technique at first seemed as if it could be useful for my application; however, while the researchers referred to this as a technique for recognizing gestures from continuous data, their implementation of this system still required gestures to come to a definitive, identifiable start and stop (for example, returning the hands to a neutral position).

Nor could I incorporate the technique used by Kelly et al. in their framework for continuous sign language recognition. These researchers trained a Hidden Markov Model for each important gesture and an additional HMM to recognize the small hand movements in between gestures [36]. While this strategy worked for sign language recognition, in a dance performance context the dancer's "non-significant" movements between gestures to be recognized are quite varied from one another and may significantly differ in scale, body parts involved, and length of time to execute (depending on particular choreographic decisions). I believed that training a Hidden Markov Model to recognize such a variety of movement as "non-significant" would be particularly challenging and would likely bring down the accuracy level of the actual trained gestures.

I experimented with several approaches to segmenting and pre-processing the data in the attempt to determine what sequences of movement represented "actual" gestures. Since the start and end of a gesture that the machine has been trained on and should recognize are not marked by consistent, particular movement characteristics that are different than the start and end of gestures that should not be identifiable, having some acceleration threshold or other movement threshold was not an option. I also debated having a pre-gesture movement that was unique enough to be identified by the system, but determined that such a solution would lead to significant choreographic restraints.

The solution that I finally employed for this context was continual examination of the most recent elements in the data stream, periodically sending data windows of several different lengths (the past 30 time steps, the past 60 time steps, the past 90 time steps) to the Hidden Markov Models for analysis. Each Hidden Markov Model reports the probability that the gesture it represents would have produced that sample. The system examines all these probabilities, picks the highest one, then returns what gesture in the gesture library it thinks that sample represents, with a measurement of how accurate its guess is. The system then screens those answers and the accuracy levels, and decides that any gesture reported with an accuracy over a predetermined accuracy level is really occurring, while any gesture reported with an accuracy below that predetermined level is a false positive. This method returned sufficiently reliable information, given appropriately trained gesture libraries, though the accuracy was not as great as the accuracy of the system on gestures when it is given the correct start and end points of the gesture.

For performance requirements, in order to make the gesture recognition even more reliable, I added an optional trigger for the performer, based on suggestions from Joe Paradiso. By touching together small pieces of conductive copper tape on the thumb and fourth finger of the glove on the right hand, the performer would send a command to the machine to collect and analyze a number of samples (again with lengths of 30, 60, and 90 time steps). This step eliminated the possibility of a false positive, of detecting a gesture occurring when the performer was not making a gesture in the trained vocabulary. With the trigger, the system can be sure that it is looking at a significant gesture, a gesture in the trained vocabulary. In this condition, the error in the system is reduced to the potential misidentification error. Additionally, using triggers could greatly decrease the computational load on the machine running the gesture recognition program, as it is not consistently running Hidden Markov Model algorithms on new data windows, but only querying the HMMs when it has been told that an actual significant gesture is occurring. However, one issue that had to be addressed with the conductive fabric on the fingertips was the occasional situation when a dancer's hands (and gloves) would get sweaty enough that the conductivity across the hand was high enough to set off the trigger. This issue was fixed by simply putting some isolating tape in between the performer's fingers and the copper tape.

The final system recognizes two types of movement information: specific discrete gestures, identified by the Hidden Markov Models, and continuous movement quality information. The discrete gesture recognition system can be run in either of two separate modes, the first where the recognition system continually sends windows of movement data to the HMMs to see whether a desired gesture is occurring, and the second where the system only sends data to the HMMs after a trigger occurs. The continuous movement quality information is calculated throughout the performance in the same manner regardless of the gesture recognition mode.

4.8: Implementation Requirements

The software implementation has a variety of key requirements to make it usable for rehearsal and performance. First, gesture recognition and mapping need to be executed in real time, without a perceptible lag between the gestures of performers and the visual or sonic results. Otherwise, in live performance, the immediate connection between the movement and the digital media would be unclear. Second, the software must make it quick and easy for the user to reconfigure mappings and change gestural vocabularies, especially during the rehearsal process. When I work as a choreographer, I develop movement during the rehearsal process, creating and setting choreography directly on the performers. In order to work with this abstraction framework and mapping tools as improvisationally as I work with performers, the framework and mapping tools must be extremely flexible. Third, gestures and qualities of movement must be abstracted to appropriate levels in this framework, so as to be conducive for discovering powerful and expressive mappings that can take advantage of the semantic, metaphorical, and expressive content of the input movement. Finally, the system must allow for some outside control of mappings in case of significant failures of gesture recognition or classification. In order to examine the use of the system and how it meets these implementation criteria, I next needed to test the system's use in the performance and rehearsal process.

Chapter 5: Performance Explorations - *Four Asynchronicities on the Theme of Contact*

5.1: Process

As an exploration of my Gestural Media Framework system, I decided to choreograph and design a performance piece that incorporated this system into the rehearsal and performance process. By testing the framework and working process for gesture and quality mappings in a real rehearsal context, I could develop the system further and discover how it affected the ways I developed a technologically-enhanced choreographic work. Thus, I set out to create a piece that I called *Four Asynchronicities on the Theme of Contact*. This performance piece consisted of four separate movements with a total of five performers: two duets, a solo, and a quintet. Each movement explored different ways that people try (or refuse) to connect in the digital age, where our contact with one another has become increasingly placeless, detached, and asynchronous (as with email and instant messages, where responses take on very different rhythms and time frames). I began with the idea of seeing how different kinds of connection and different time frames could be explored and enhanced through the gesture recognition technology. I wanted to see how a range of interactions between performers could be echoed in music or visuals, and ways that moments of contact and gestures could be fragmented and removed from their partnership origins to create new types of interactions.

With this basic structure and theme in mind, I held dance auditions in December. I cast five performers, all current or former MIT students or Amherst College graduates: Kevin Burchby (Amherst '08), Noah Jessop (MIT '09), Danbee Kim (MIT '09), Lisa Smith (Amherst '09), and Xiao Xiao (MIT '09, MIT Media Lab graduate student). It was important for me that the performers have movement and dance experience, which was true of all participants; in addition, Noah and Xiao had significant musical backgrounds. While I have experience choreographing for performers who are not professional dancers, I knew that I needed a certain level of physical precision and dance experience in my performers for me to create my desired movement vocabularies.

Rehearsals took place over two months, beginning in early January and culminating in a set of three public performances on February 26, 27, and 28. These performances were held in the E14 event space in the new Media Lab building. I choreographed the pieces and developed the content with input from the performers, designed costumes, and sound-designed the pieces. The lighting design was by Peter Torpey, who also assisted with some of the design for the visual projections. There was no set design, but a projection screen and projector already built into the space were used for the projection of visual imagery. For this performance, I obtained partial funding, which paid for sensor systems, costumes, and networking equipment, through a grant from the Council of the Arts at MIT.

During the rehearsal process, I worked with each set of performers on developing movement and story content. Often, this would begin with improvisational exercises around a particular theme (five variations on meeting for the first time, for instance), and then I would shape movement specifically for the performers. I would frequently choreograph shorter sections of movement, then experiment with repeating, sequencing, and varying those sections to create longer units. In particular, I often used the choreographic technique of repeating the same sequence of movement several times with tempo, timing, and quality variations on each repetition. We also worked simultaneously in the rehearsal process with the sensor outfits and gestural system, selecting and developing elements of gestural vocabularies that we felt were important, training the system to recognize specific gestures, then exploring mappings to sound or visuals driven by both already-choreographed sequences and improvised movement. As the pieces developed, it became clear what movements seemed like significant parts of the gestural vocabulary and thus would be useful to amplify with the sound, visuals, or lighting. Additionally, we explored ways in which the performers' qualities of movement could relate to media outputs for each piece.

5.2: Movement 1

Illustration 35: Lisa Smith and Kevin Burchby in Movement 1.

(Photo by Peter Torpey)

The first piece in the program was a duet performed by Lisa Smith and Kevin Burchby. As we worked improvisationally in rehearsal and started to develop movement content, a story started to emerge: this piece was about an earlier time, before the Internet and cell phones, where people communicated with one another using physical things like handwritten letters. In this world, a man and a woman were in a long-distance relationship, with intense, physical moments of time in the same space, separated by times when their only contact is through pieces of paper. When they come back together, however, the sense of connection has changed, and they have to struggle with whether they want to maintain the same kind of relationship. This piece was focused primarily on the interactions and emotions of the story. Additionally, this was the only piece of the four that used text spoken by the performers: the piece went back and forth between sections of duet choreography with the performers in close proximity and physical interaction and sections of spoken text with the performers in two separated spaces. For the media elements of this work, I chose to explore shaping projected washes of color and theatrical lighting with the performers' movements. As this piece was developing into a story that strongly centered on the interactions between the two performers, it was important to me that the technology be subtle and primarily affect the mood of the environment. Additionally, as this movement had an "old-fashioned" feel to it, I did not want the technology to be overly visible and "modern," and thus feel jarring and anachronistic to the story.

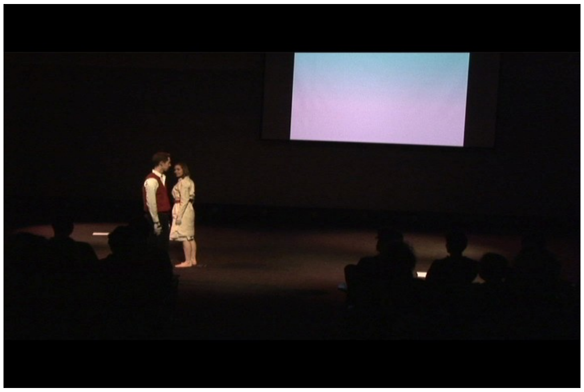


Illustration 36: Color gradient projection in Movement 1.

Illustration 37: Touching hands changes the screen color.

The visual design of the piece involved a gradated color wash on a projection screen behind the performers, using a palette of warm and muted colors (such as mauve, pale yellow, and gray) chosen for their connection with nostalgia and memory. Throughout the piece, pairs of colors were projected on the screen with a vertical gradient. These colors slowly faded first to paler versions, then from one pair to another, with the rate of the fading affected by the speed of the performers' movements: the faster the performers moved, the quicker the transition between color pairs. Specific gestures of the performers related to strong moments of physical interaction in their relationship (such as embracing one another, spinning with hands joined, reaching for one another) caused a quick shift to preset color combinations.

Illustration 38: Kevin spins Lisa around quickly, brightening the stage.

(Photo: Marina Porter).

The lighting design was similarly influenced by the movement of the performers. To control the stage lighting, we used a USB to DMX converter that allowed the computer to give commands directly to the lighting system. Peter Torpey wrote a library to take in Open Sound Control commands and output them as serial commands, so external programs could also control the lighting. As in a standard lighting design, there was a preset list of lighting cues that were externally triggered at different points of the piece; however, with this system, the specific light level at each point was adjusted through the control system according to the speed of the performer's movements. As Lisa and Kevin moved faster and faster, the light level similarly increased in intensity. As they moved more slowly and in a more restrained manner, the light level decreased in intensity. This resulted in subtle, real-time shifts of the lighting that were immediately reactive to the performers' movement.

Illustration 39: A spoken section in Movement 1, with performers in separate spaces.

5.3: Movement 2

Illustration 40: Danbee Kim in Movement 2.

(Picture: Peter Torpey)

The second piece was a solo, performed by Danbee Kim, about a woman in relationship with a pervasive computer system. The story of this piece shifted over the course of the rehearsal process. In the beginning, I knew that this was a story about a woman who was alone, as this was a solo piece in a work about communication. I also knew that I wanted to work with a model of performance where Danbee could play with the gesture recognition system as with an instrument, so her interaction with the system would be fairly direct. During rehearsals, we recognized that this story could be the future of Movement 1, a relationship in a time where it was no longer necessary for someone to be interacting with another human being, and could instead be relating to some pervasive computer infrastructure. Danbee's character was a woman (perhaps part machine as well) who uses her movement to interact with an intangible system in the space, but finds herself missing the physical contact of a tangible interaction. While in interaction with a mechanically-inspired sound environment, she finds herself performing gestures linked to interactions with another physical body, such as an empty embrace, which drives her to frustration and eventually a reluctant acceptance of her separation.

Perhaps best of the four pieces, Movement 2 incorporated the use of both the gesture recognition abstractions and the quality of movement abstractions. The performer's journey through a three-dimensional quality space (Laban's Effort qualities of *Weight*, *Time*, and *Flow*, as described in Section 4.4.3) was connected to her journey through a soundscape. Different sampled sounds were mapped to different regions of the three-dimensional quality space, with each sound becoming active when the performer passed into that sector in the quality space and fading out when the performer left that part of the quality space. Additionally, other samples for particular effects were triggered by the performer making certain trained gestures. Thus, the soundscape at any moment in the performance was a set of sounds influenced by both the qualities of the performer's movement over time and the performer's specific gestures. Additionally, another layer of sonic control was added later in the rehearsal process, after Tod pointed out that the sound felt overly static when a specific sample was triggered on a performer moving into particular regions of the quality space, but then was not further shaped by her movement as long as she remained in the region of the quality space where that sample remained active. Thus, the performer's current locations on the Time axis and the Flow axis were used as parameters to control reverb levels affecting the entire sound mix. However, this solution did not reach quite far enough to make the sound always feel in the performer's control and always be affected by the subtleties of the performer's movement. I am currently in the process of reworking and expanding this solo piece, and the concept of constant control is a major focus in this development.

Illustration 41: Danbee performs a "ding" gesture for the system to recognize.

In the process of developing the soundscape for this piece, locating specific sounds at particular regions of the quality space, I found that my abstraction framework made it fairly simple for me to think at a higher level about what sorts of sounds in my palette should be connected to particular types of motion. Some sounds were, in my mind, associated with harsh, quick, and firm motion, and thus correlated with low values on all three quality axes. Other sounds were associated with light and quick motion, so were related to high values on the Flow axis and low values on the Time axis. Other sounds intuitively fell at various other points throughout the quality space. Additionally, it was very easy to experiment with which sounds should be linked with specific trained gestures, which conjured up strong and metaphorical associations between gesture and sound in the context of the piece (for instance, a ringing bell upon the tap of a raised hand, or electric buzzing with an empty embrace).

Illustration 42: Movement 2.

(Photo: Marina Porter).

This piece came the closest of the four to the model of using the gestural system as an instrument, rather than to provide an accompaniment. This was partially helped by its being a solo performance, where the focus was on the movement of one performer and the relationship of that performer to the sound score and the sound manipulations, rather than on the relationships between performers. Additionally, I began the rehearsal process with Danbee by first developing a vocabulary of potential upper-body gestures that we thought were interesting and potentially meaningful. We developed movement sequences using this gestural vocabulary, elements of which were then later correlated with particular musical triggers and sound manipulations. By very

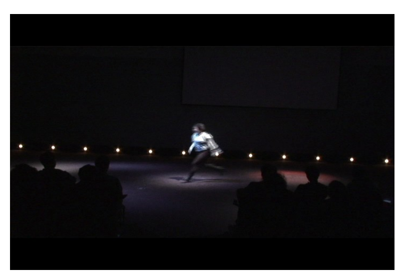


Illustration 43: Quick, sudden movements such as running changed Danbee's location in a three-dimensional quality space.

early discovering and experimenting with a particular vocabulary and the potential sound associations, we were able to develop a much more instrument-like interaction as I found the story that was being told by those interactions and the particular movement vocabulary.

5.4: Movement 3

Illustration 44: Noah Jessop and Xiao Xiao in Movement 3.

(Photo by Peter Torpey)

The third movement, a duet performed by Noah Jessop and Xiao Xiao, had a similar story to Movement 1 in that it explored a relationship where two people were sometimes in the same place and interacting in "real life," and other times were having separate interactions. In this piece, I wanted to explore through movement and sound the different rhythms of interaction that take place when people communicate over email or instant messaging or other digital technologies today, and how those differ from the rhythms of face-to-face conversation. We developed highly interactive sequences of movement that could be executed with both performers together in the same space, made disjointed by breaking synchronous movements into call-and-response timing, or fragmented completely when the two performers were separated and no longer physically interacting with one another. I became very interested in the ways that the same set of gestures could take on new meanings depending on the timing and spatial relationships between performers. When Noah takes Xiao's hand and pulls her close to him, the emotional effect of the movement is very different than when the identical movements are performed in a different physical and temporal context. Noah reaches out a hand, closes his fingers, and pulls it to his chest, while two seconds later Xiao, in a different part of the stage, lifts up her hand and steps forward as if pulled by that hand.

From early on in my work on the piece, I knew that I wanted a sound design that could highlight these different rhythms of interaction, and show the expressive differences between two people having simultaneous interactions, having staggered, asynchronous interactions, and having completely separated interactions. I thought that a sound design where each performer controlled a separate instrumental voice, with notes controlled by the performer's gestures and movement, would help explore and expand the changes in the interaction, such that the same gestures would be seen to be very different when the two performers were in the same space at the same time from when they were separated in time and space. Therefore, I could expand on the differences and variations in taking the same movement out of its origins in close interaction.

Illustration 45: A waltz with the performers disconnected.

Illustration 46: Fragmented movement sequences.

In Max/MSP, I designed a sound score where each performer's movement controlled a semi-random walk through notes in a particular scale. Each performer's sequence of notes was voiced with a different instrument and individually controlled by the Weight and Time parameters of his or her movement; as in Movement 2, when the performer passed into a particular range on the Weight and Time axes, a new MIDI note was triggered. These locations on the quality axes were empirically tuned such that each separate gesture would create a note (or multiple notes on particularly rapid and strong gestures). Additionally, the range of the semi-random walk (how far a given generated note could be from the previous notes) was affected by the quality measurements of the associated performer. As the performer's movements became less fluid, the range would increase, resulting in a more jumpy sequence of notes.

Both performers' instruments used the same scales as a base for their composition, such that all notes played remained in the same tonality. Additionally, a drone tone at the root of that key was consistently played to make the overall sound less sparse. The given key and scale for the generated notes could be switched on command to a set of other pre-programmed pairings, so specific trained gestures (such as unfurling a hand or pulling the other person close) were used to trigger these changes. This helped bring additional musical variety to the score.

However, this sound score highlighted for me that often I had the two performers doing movements at separate times during the "connected" sequences, due to the nature of partnering movement. Thus, the distinction between the separated choreographic sequences and the connected choreographic sequences was not nearly as clear aurally as it was visually, with the exception of the specifically asynchronous movement sequences where the turn-taking was obvious both in the choreography and in the sound. Another major limitation in the sound score was my use of MIDI for the musical content, which limited the sonic quality of the piece. This also resulted in occasional problems with notes being cut off prematurely (without a fade) by earlier messages turning that note number off.

Illustration 47: Brief moments of connection in Movement 3.

Illustration 48: Separate spaces, connected choreographically.

5.5: Movement 4

The final movement of the performance featured all five performers: Lisa, Kevin, Xiao, Noah, and Danbee. In contrast to the more restrained choreography of the first three movements, I found this piece developing around the excitement of playing with five bodies in space and the myriad combinations of interaction that could occur. We began creating sequences of movement that were centered strongly on physical interactions. Additionally, interaction motifs and pieces of choreography from the other three movements were brought into the choreographic design for this piece to create more continuity throughout the performance. In this quintet context, earlier-seen movement phrases transformed through different tempi and phrasing, as well as through their juxtaposition with other motifs.

Illustration 49: Xiao, Noah, Lisa, and Kevin in Movement 4.

(Photo by Marina Porter).

As the piece developed further, I found that this piece, in contrast to the other three stories of separation and distance, was a celebration of the complexity and excitement of same-time, same-space interaction. Therefore, I sought out a visual design for this piece that would echo the complex, frequently changing, layered relationships and interactions that the performers. I wanted individual elements that would be clearly shaped by each performer, but that would combine to produce something richer than the individual elements. I chose to begin with the fluid dynamics simulation that I had explored with gestural control previously (See the Gesture Glove description in Chapter 3). Here, I created a separate colored force in the fluid for each of the five performers, with a color that stayed consistent throughout. Rather than a performer specifically controlling the direction of the force's movement, however, I had the quality of the performer's movement affect the quality of the movement of the force. For example, the Time axis was mapped to the speed of the force's movement: the faster the performer moved, the faster the force would move through the fluid. When a performer's Time value reached 1.0 (no change, such as when a performer was standing still), the speed of the colored force would drop to 0.0 and that color would gradually fade out.

Thus the forces visible in the image at any point corresponded to the performers who were currently moving. The Weight axis was mapped to changes in direction: the more energy the performer put into her motion, the more the force varied in direction, resulting in more loops and zigzags in the output motion. Additionally, as the performer's movement became less fluid, the position of the force would exhibit more randomization within a small range.

Illustration 50: Traces of color on the screen build up over the course of the piece.

Illustration 51: The performance starts and ends as a solo.

One interesting piece of feedback that I received from some audience members about this performance was that while they quickly recognized the relationship between performers and the visuals, the fact that this relationship remained unchanged throughout the performance made them soon stop paying attention to the visuals. While the visual appearance of the projection developed and continued changing and growing over the course of the performance, it would have been interesting and powerful to also have the mapping between these visuals and the performers continue changing and growing in a similar manner.

Illustration 52: Individual points of color correspond to particular performers.

Illustration 53: Shifting relationships echoed with imagery.

5.6: Performance Evaluation and Audience Reaction

5.6.1: Evaluation

I feel that *Four Asynchronicities* was a useful first test of the Gestural Media Framework in a performance context. I developed a complete performance work, lasting around 45 minutes in all, that incorporated the recognition system in all pieces, five separate performers and sensor systems, a variety of quite different output systems, up to five gesture and quality recognition systems running simultaneously, and a high level of necessary system function for three separate performances. All the hardware and software worked smoothly in general, with one exception during the Saturday night performance when one performer failed to immediately connect to the system (caused by the Xbee networking rather than my code). Even this one failure case occurred in between pieces, and everything worked smoothly after two minutes and a computer reboot. The remainder of the system worked satisfactorily.

Also, with the four separate pieces in this performance, I had the chance to explore how the system would support a variety of different amounts of interaction between the performers' gestures and the resulting media transformations. I was pleased to see that the system supported different models and kinds of interaction between the movement and the media equally well. This interaction was the most subtle in Movement 1, where the overall intensity of the movement drove the intensity of the stage lighting and the fade speed of different color washes, and specific gestures triggered particular changes in the color washes. The mappings between gesture and media became most specific and precise in Movement 2, with immediate sound connections between specific gestures and particular sounds. It was clear that movement affected the music in Movement 3 and the visuals in Movement 4, but the precise details were not directly mapped. The model of interaction also varied significantly between pieces, with Movement 2 having the most instrumental interaction with the system, and Movement 1 having a much more affective model of interaction.

I also think the particular recognized gestural vocabularies that were used for different pieces met with different amounts of success. Interactive moments, such as embraces or reaching out a hand to someone else, carried so many preexisting implications...while this was the reason I had selected them as important elements of a recognized gesture vocabulary, it is possible that their use as a trigger for new visuals (in the case of Movement 1) or key changes (in the case of Movement 3) may not have been clearly identifiable as important gesture recognition moments and instead were seen primarily as a particular interaction between two people. In contrast, much of Danbee's solo vocabulary such as the bell-ringing gesture could be easily seen as causing a specific sound effect, as there was no other person on stage to add layers of social meaning to each gesture.

It is also necessary to examine the extent to which the framework and implementation satisfied the requirements discussed in section 4.8. The first requirement was the real-time execution of gesture recognition and mapping. That this requirement was adequately met by the system. In the performance and rehearsal explorations, I did not experience significant delays between the performance of a gesture and the recognition of that gesture, nor between the recognition and the triggering or shaping of specific mapping elements. In fact, due to the system's use of varying-length windows in the recognition process, certain gestures could often be identified with sufficient accuracy by the system before the performer had completely finished executing those gestures, thus speeding up the recognition/mapping/output path.

The second requirement was that "the software must make it quick and easy for the user to reconfigure mappings, especially during the rehearsal process." Particularly with the mapping systems that incorporated Max/MSP, mappings were simple to reconfigure. The biggest hurdle to adjusting mappings was the time needed to add parameters to some of the output systems. Additionally, the abstraction of movement into gestures and qualities made it easy for me to locate where I should be changing the mappings, rather than having to tweak code full of specific sensor values. I also found the ability to reconfigure mappings quickly and intuitively to be very useful during the performance process. In one performance, one arm of Danbee's sensor shirt did not start sending data at the necessary time in Movement 2, resulting in inaccurate ranges for all the quality values. However, with the fast reconfiguring of mappings possible in Max/MSP with the Gesture Objects and Quality Objects, I was able to recode the ranges necessary for shaping particular sound sequences early in the performance of the piece, such that the overall soundscape resulting from the performance was still quite similar to what it would have been if both arms had been sending the proper movement data.

One difficulty with creating mappings quickly was the complexity of the output media I was working with. In contrast to the Gesture Glove project, where I had constrained each output visualization to have a fairly small set of control parameters, the media outputs in Four Asynchronicities had room for a large number of control parameters. Additionally, I was first discovering and shaping what input gestures and what control parameters were interesting in the context of the rehearsal process. Even if making connections between the gestural input and the output control was fairly straightforward, defining the output control parameters and setting them up in the mapping systems often took a bit more time. I was aware that I did not want to spend too much time in rehearsal programming new output parameters, which limited the number of things I could experiment with each rehearsal. For example, in Movement 2, it was trivial to adjust the range of quality values that would trigger a particular sample or that would shape a particular sonic affect, but it was not so quick a task to introduce a new sound manipulation technique.

The third requirement was that the semantic abstraction of gestures and qualities of movement be at a level such that use of those gestures and qualities in mappings would be intuitive and easy, supporting powerful, metaphorical, and rich connections between mappings and movement. I found the process of working with the abstracted gestures and qualities of movement easy, and thought the system was conducive to thinking about mappings and gestural languages. However, I do not believe that all of the resulting mappings were equally powerful. In certain pieces, such as Movement 2, the mappings were very strong and drew a fairly compelling connection in between the performer and the soundscape she was creating. It was clear that the quality of Danbee's movement was directly affecting the soundscape in compelling ways, including the timbrel changes related to different qualities of movement and the evocative associations of certain gestures with particular sounds. In other pieces, such as Movement 1, the mappings (particularly of movement to stage lighting) were too subtle and not significantly noticed by the audience. I suspect this fluctuation in mapping quality and expressivity is partially due to the variation in the amount of rehearsal time that each performance piece had with the system and thus the amount of time that I had to imagine and develop interesting mappings.

The final requirement was that the system "must allow for some outside control of mappings in case of significant failures of gesture recognition or classification." With my addition of outside failsafe triggers that could I could control from my position at the computer, I was able to step in at any occasion when a gesture was not recognized. Additionally, we did not encounter significant errors in gesture classification that would need to be corrected. Thus, I feel that the system adequately satisfied this requirement.

5.6.2: Audience Reaction

Four Asynchronicities was performed three times, with an audience of 20 to 30 people each night. As part of the evaluation of this work and the use of technology in the context of the performance, I distributed brief evaluation forms to those who attended the performance and made those forms available online. Over the three performances, I collected written feedback from approximately 25 audience members. The feedback forms asked the audience for their immediate reactions, what aspects of the pieces they found more or less compelling, and how well they thought the performers' movements correlated to the media elements. These broad questions were intended to elicit responses about many aspects of the piece. The performance met with a variety of reactions from audience members, who responded to the emotional arcs of the pieces, the design, the levels of connection between the performers and the media, and the performance of the dancers.

I was surprised by the extent to which the process of obtaining feedback from the audience also turned out to be an exercise in shaping audience expectations. I specifically did not provide much information about my thesis research or the specific kinds of things I was measuring or mappings I was exploring, seeking to obtain feedback that was not colored by this information. However, I might have received different feedback had I asked the audience to give their feedback in a different way. By giving questionnaires to the audience before the performance, mentioning that there was gesture capture technology involved, and explicitly requesting their feedback, I was priming the audience to experience the performance in technological terms and question "what worked well" or "what didn't work well." Particularly with an MIT-based, technical audience, it was likely a mistake to tell people before watching the performance that there was gesture recognition technology incorporated into the pieces and that they should be watching for and critiquing details of the relationship between the technology and the performance. I would have been interested to see how the audience would have responded had I allowed them to see and experience the performance with no prior requests for feedback, and then provided forms afterwards for them to submit comments if desired. Would they have seen the use of the technology, if it had not been pointed out to them beforehand? One audience member even specified the effect of these questionnaires on his/her preconceptions of the performance, stating "My expectations for the performance were greatly lowered upon reading the survey questions upon entering." While a slightly harsh comment, it is also an accurate assessment of the potential impact of the survey on audience members' perception of the work.

Overall, audience members seemed excited by the way in which technology was incorporated into the pieces. Movement 2 was seen to have the clearest (the most "obvious") correlations between Danbee's performance and the sound score created by her movement. The immediate relationships between movement and sound caused audience members to imagine "interactions with invisible objects and strange extensions of her gestures. Was she shooting electric slinkies out of her palms? Did she have yokes of engine block hanging from her wrists?" Understandably, as a solo, this piece was more easily able to explore a direct relationship between performance and technology–one performer and one sound score reacting in various complex ways. Movement 3, with two performers generating a musical score, faced more difficulty with people understanding how movement of each individual dancer, or the two combined, affected the developing music.

Movement 1 met with a variety of reactions: some people loved the story and the relationship explored in the piece, others found it too slow-paced and boring. The fact that the lighting was affected by the movement was only clear to those who were watching carefully, though audience members who noticed this correlation were excited to discover this dynamic visual subtlety. For myself, I feel that this piece could have been made stronger by removal of the slow-paced text sections and having much more time to develop more interesting and clearer mappings between the movement and the lighting design. While the lighting effects created by the performers' qualities of movement had far more variety and liveness than what would be possible with standard lighting cues, that the connection between expressive performance and expressive lighting could have been drawn much more tightly.

In general, the mappings between movement and sound were seen as more straightforward and intuitive than those between movement and visual elements. This may be related to a piece of feedback that I got from several audience members about the position of the projection screen in relationship to the performance space. Unfortunately, the height of the screen as preset in the event space meant that it was difficult to keep both the screen and the dancers in one's field of view at the same time, resulting in the audience having to either bounce their attention between the two elements or choose to only pay attention to one at a time. In retrospect, I realize that the mappings between the dancers' movements and the visuals in Movement 1 and Movement 4 could not be clearly grasped from all angles of the audience, since one would not see a movement and the resulting visual manipulation at the same moment. I would be interested to see how the audience would have responded to the mappings in these pieces if the spatial relationship between projection and performers had been different, if, for example, I were to put a rear projection screen immediately behind the dancers or if I were to project on the floor beneath the dancers.

Interestingly, Movement 4, while the performance and the physicality of the choreography were generally enjoyed, met with quite varied opinions about the correlation between the fluid visuals projected on the screen and the dance movement. Some audience members found the connection between these two elements to be quite clear, with a few remarking that the mapping was so obvious that it became less interesting after some time. However, other audience members felt that they did not have a good sense about how the visuals in this piece corresponded with the movement. I suspect this range of reactions is due to the way in which certain elements of the mapping were directly connected (one colored element of the visuals moving with one particular performer), while others had higher-level interactions (the exact path of a colored fluid shaped non-deterministically by the quality of the associated performer's movement).

I also appreciated seeing the audience's comments on the ways in which the different experiments in mapping related to one another. One audience member had a particularly interesting reaction to the sequence of the pieces and the varied kinds of intersections of movement and media that they contained. This audience member described Movement 1 as having non-direct mappings but a clear storyline and emotional content, guiding the audience to think about what higher-level interactions might be at play between the technology and the performers, and Movement 2 as showcasing some clear and precise connections between particular gestures and sonic results. This audience member saw the first piece as focusing on the story and the performance while introducing subtle mappings that you had to focus on closely to pick up the connection, the second piece showing more virtuostic one-to-one mappings, and then the final two pieces having mappings that were less direct, but still able to be appreciated because of the way the first two pieces had introduced the range of possible interactions between performers, sensors, and reactive media. This audience member stated:

> Essentially, I think the evening ably avoided the ordinary "technology-in-performance" audience fears: that they will miss the mappings, or not be able to read any semantic content into them, or that they will be watching a boring, linear performance in which dancers simply play their sensors. By quickly teaching the audience that the show is not about linearity, and in fact, not about technology (augmented by it, sure, but not *about* it), you avoided the typical pitfalls of this type of performance.

This statement that the show was not about the technology, but about the performance aspect of the piece, was echoed by other audience members. To me, that is a significant success for the work. The goal of my technologies for the performing arts is to empower performance work and add powerful, interesting layers to performance works; I do not aim to create works purely "about" technology. It is possible for technology to be a vital part of a performance's context and content without being its primary content, for technology to enhance and support the story rather than be the entire story. Thus, I consider it a success to have created a performance work that took advantage of, was shaped by, and drew on gesture recognition and mapping technologies while still having its own content and stories to tell.

Chapter 6: Conclusions

6.1: Gestural Abstractions in Mapping – Lessons Learned

Having described and reflected on the first performance explorations using the Gestural Media Framework's approach and toolkit, I will now discuss further some of the challenges I encountered and things that I discovered throughout the process, as well as some of the successes of the approach and of the current framework implementation. With these elements in mind, I will propose some future development directions for the Gestural Media Framework, as well as other applications for this framework.

While some of the mappings created in the performance pieces for *Four Asynchronicities* proved more compelling than others, the general process of working with the Gestural Media Framework supported creating high-level mappings with specific gestures and qualities of movement, leading to some metaphorically resonant interactions between performer and media. Especially with the layer of abstraction from sensor data introduced by the machine learning step and the Hidden Markov model, it actually became difficult to think of a gesture as a particular set of component signals rather than as a holistic motion. In this framework, the gesture recognition step outputs details about the gesture performed without explanation of how it determines that gesture. With this condition, it seemed necessary to perform and experience a gesture as a gestalt, without trying to manipulate the system through knowing which angles of joints or patterns of acceleration signed for a particular gesture. A large success of this system was that it did obtain the desired goal of letting users work with gestural and quality abstractions directly in mappings. In the rehearsal process, I found it fairly easy and freeing to think about relationships between specific gestures or qualities of motion and output behavior.

In the creation of *Four Asynchronicities,* I was also satisfied with the ability of the same gestural and quality input and abstraction system to handle a wide range of output media, including sound and music, projected visuals, and stage lighting. I found that the framework was flexible enough to handle all these varieties of mapping, and that it was quite freeing to focus on each mapping case as a high-level artistic question rather than as a low-level implementation question.

Additionally, the resulting system was stable and able to be used in three consecutive performances with a minimum of error in receiving sensor data and a minimum of sensor failures. In the one performance where there were networking issues with the Xbee modules, the failure was caused by a timing problem in communicating to the Funnel I/O boards that had been previously experienced by others in the Opera of the Future group working with this hardware. This issue will need to be addressed to do significant future work with the Funnel boards. However, the on-the-body sensors proved to be robust, and the system as a whole was sufficiently stable and reliable to serve the demands of a performance context.

One of the significant challenges during this process, which took up much more time than I had planned for in my initial schedule, was developing and working with the gesture recognition system. While the GART libraries made the initial implementation of gesture recognition using Hidden Markov Models much simpler, the amount of abstraction incorporated into the libraries made debugging a challenge. In particular, it took a significant amount of debugging and parameter tweaking to get the system to recognize different sets of gestures from different performers, as it had been constructed to use only one set of Hidden Markov Model files at a time. These issues slowed down the full integration of the system into the rehearsal process, which was non-ideal in developing the media content simultaneously with the movement.

It would also be useful to rewrite the GART libraries to be more flexible with the details of the Hidden Markov Model implementation. In particular, the standard setting for the size of the HMM for gesture recognition was eight states, but I would have liked to experiment with larger and smaller HMMs (containing more states and transitions), to see how those changes affected the accuracy of the movement recognition. However, this size setting was completely inaccessible in the libraries as written, and required more restructuring of those underlying libraries than I had time to undertake before the performance. This rewriting is a step that I will likely take early in my future work with this project.

It is possible that having the ability to adjust the HMM algorithms might provide more flexibility in defining gesture vocabularies. I limited the specific gestures that needed to be recognized to fairly small sets for each piece in Four Asynchronicities, so as to maintain the highest level of accuracy in the gesture recognition process. However, I would be interested to see whether I could maintain equal accuracy values with larger gesture sets by increasing the number of states in each HMM.

Another challenge was the recognition of gestures from continuous streams of motion, as discussed in section 4.7.3. While the final solution in my implementation, a combination of capturing overlapping set-length data windows and backup on-the-body triggers from the performers, was functional for this application, there need to be even more accurate methods for filtering "important" movement information from the majority of the movement data. In future work, it would be useful to explore other solutions to this problem.

Additionally, it would be interesting to explore the ways in which the framework could support users in determining vocabularies of key gestures each of which is distinct enough from the others to have very low rates of gestural misidentification. In my rehearsal process for *Four Asynchronicities,* the shaping of these gestural vocabularies was determined empirically. There were a couple of occasions where two particular gestures on which I was attempting to train the system were repeatedly confused, even with additional training examples. In these cases, I generally chose to remove one of those gestures from the main gestural vocabulary. In order to expedite this process, a system could provide feedback about similarity metrics of gestures (at least from the point of view of the machine learning process) to encourage the user to develop a wide gestural vocabulary.

There was compelling evidence that it was necessary to incorporate the use of the system as a constant and integral part of the rehearsal process. For the pieces such as the solo second movement, where the system was a strong presence in the rehearsal and the thought process of developing the piece, the resulting mappings were strongly linked to the content of the work and the content of the work was shaped by the mappings that were developed in the rehearsal process. In pieces such as Movement 1, where some of the technology (such as the stage lighting control) had to be incorporated later in the rehearsal process, I found that while it was simple to create mappings and connections between the performers and the media using GMF, we did not have a chance to fully develop and explore those mappings in the context of the piece. As in any piece, more rehearsal time with the system and the mappings led to better mappings, more exciting and compelling extensions of the live performance, and more expressive control.

The Gestural Media Framework could be used for all three interactive performance system models that Wanderly describes in [79]: digital instrument, interactive installation, dance accompaniment. However, in my work with *Four Asynchronicities,* the model of interaction between performers and system that appears to have led to the richest mappings was the model of the system as an instrument shaped by the performer, rather than the system as a coexisting accompaniment to the performer's movement. When the content of the piece seemed most strongly linked to the technology, there was the impression that the performers were "playing" the technology, interacting with the system with deterministic finesse and control, not simply performing in juxtaposition with the system's results. When the media and the performance occurred simultaneously without the performer feeling a strong level of control over and relationship with the system, this introduced some uncertainty about the connection between the two elements and how they were connected in real time, and risked being seen as the standard dancing-to-music relationship.

I also found that mappings that used movement qualities to do continuous control were much more necessarily linked to the performer's movement than triggers based on specific gestures. As a requirement of the implementation for performance, I had built in a few failsafe triggers, controlled by me, in case particular gestures were not recognized correctly; however, in the situations when it was necessary for me to use those triggers (for instance in the Saturday night performance when one of the performers had a Funnel I/O board that could not be correctly connected to the computer), I found that there was not a significant difference in timing between a sound being triggered by the system recognizing a gesture and a sound being triggered by a human recognizing the same gesture. With my experience in stage-managing dance performances, I know that stage managers are able to very quickly react to specific movement cues, as well as to anticipate those cues. Thus, associating triggers with particular gestures in already-choreographed sequences may not require the gesture recognition software. Where the strengths of the computer technology came in were in following complex quality parameters that changed moment-to-moment (such as the Weight and Time parameters), and in tracking movement or dynamics over longer timescales than people generally do, as with the Flow parameter or with the graphics generated by the performers in Movement 4, which maintained a record of movement over a number of minutes.

Working with qualities of movement rather than pure sensor data is also interesting in how it can affect the timescale (distinct from "delay") over which the results can be affected. We experience movement, sound, visuals on different scales than sensors. For example, if one looks at a visualization whose scale is tied to the immediate amplitude of speech, this visualization will quickly flicker bigger and smaller over a wide range due to momentarily silences caused by consonants, breaks between words, etc. However, in listening to the same speech, we generally would experience the change in amplitude more holistically. Similarly, we do not experience movement in discrete periods of less than $1/30^{th}$ of a second, as movement data is captured by the computer system. Qualities of movement, in particular such qualities as Laban's Flow, describe movement over a longer time than windowed sensor data, and thus may be more able to capture movement in the way we experience viewing that movement.

6.2: Future Directions

Because of my background in choreography and the performing arts, I have experienced the extreme importance of gesture and movement as a vehicle for emotional and narrative expression, both in the arts and in our daily lives. The human body is fixed in scale, fixed in location, fixed in form, yet still able to communicate immense amounts of emotion and expression. As we seek to create new technologically-based forms of expression, performance, and interaction, it is important to look at the relationship between technology and movement, as well as the relationship between technology and the body. The organic rhythms of a human body are very different than the rhythms of a computer system, so programs that draw their input from movement and breath and gesture can interact with humans to very different effect. If we can create systems that make it increasingly easier to pay attention to those interactions, the resulting technology will be richer and more connected to our physical experiences.

Additionally, there is still currently a gap between artists who use technology and technologists who create art. Generally, creating a technological performance piece requires collaboration between a choreographer/performance creator and a technologist/system creator, or to have the performance maker also be fairly technologically advanced. I would like to increase access to creating interactive performance works for those who do not necessarily have a strong technological background, by creating tools that make it as easy and high-level as possible to work with gesture recognition and quality of movement parameters in shaping media. How would performance-makers whose primary background is in performance rather than technology create pieces with a system that no longer requires the user to think like a programmer? While the Gestural Media Framework is a step in this direction, the current framework implementation still has several aspects that require the user to be familiar with Java coding, preventing it from being completely accessible to a non-technologist. For example, in order to add new types of sensors or change from the sensor shirts that I have developed, one would still have to write new Sensor classes to take in and appropriately process the data. Likewise, different Gesture classes are hand-coded, along with the QualityRecognizer classes, so the implementation requires some more development to become easily available to and usable by people with little programming experience.

Additionally, in future work, it would be useful to create a single mapping system that could provide easy and sophisticated control over a variety of these output systems, as my current implementation used a combination of Max/MSP for sound generation and individual hard-coded Java mapping systems for controlling projected graphics and stage lighting. While the OSC output of the gesture and quality recognition engines can be used as input for a variety of systems, it would be useful to have one mapping system designed for creating these sorts of interactive gesture-driven performances that can handle many different kinds of output. What kind of mapping system would support performance-makers in creating sophisticated interactions between the gestural abstractions and the details of output media? Overall, what tools (programs, sensor systems, mapping software, etc.) can allow skilled artists who are non-programmers to incorporate technology into their work? How can we create and encourage technology for performance and expression that becomes a vital and necessary component of that artistic expression, a key part of the story?

In my future work, I would also be quite interested to expand my analysis of movement, gesture, and expressive qualities into full-body sensor systems, as well as to develop a fuller descriptive framework of movement and dynamics. For this thesis, I limited the scope to movements of the upper body (or only the upper-body component of full-body movement). While the implemented sensor system was able to detect a variety of interesting movement, there were occasional performance limitations due to the fact that the system could not pick up movement of only the performer's feet, if the arms and upper body were being held steady. For example, in Movement 3, there was a brief section where the performers waltzed together; in that section, the sensor data on the arms showed little movement, but one would expect some system reaction to the swift and rhythmic movement of the performer's steps. Additionally, this restricted movement scope is occasionally choreographically limiting, as it encourages one to think of movement in terms of the upper body. While this is fairly conducive to my personal choreographic format of stylized pedestrian movement, it would have been helpful to have a tool that inspired a greater variety of movement.

Particularly in the field of technological enhancement of dance performance, we can gain significant expressive capabilities and variety of movement by paying attention to movements not only of the upper body, but also of the lower body, the torso, the head, and the entire body. These explorations could be done with full-body sensor systems, though full detection of gestures would likely need richer movement capture capabilities than existing full-body systems such as Troika Ranch's MIDI Dancer system (which has bend sensors on major joints, but no further input capabilities) [71]. Computer vision systems could also be incorporated to detect full-body movement, though these systems would need significant knowledge of the form of a human body in order to properly recognize movements performed at a variety of locations and angles in relationship to a camera setup. Perhaps with a combination of on-the-body and computer vision systems, full-body motion and movement qualities could more richly explored.

Additionally, it is necessary to explore further integration of both dynamic gesture and static poses as forms of evocative movement. In the current implementation of this Gestural Media Framework, gestures are learned and defined by the system as time-varying, dynamic movements. If one wanted to record a static pose as a "gesture," it would still be necessary to capture a time-dependent sequence of that static pose, but without any change occurring in sensor values over time. A significant portion of human physical expressiveness comes from live movement changing and developing over a period of time. However, there are also semantically meaningful poses that should be integrated into a gesture recognition model. In a given performance context, it may be important not only to detect a performer spreading his arms out, but also to detect him standing still with his arms stretched wide open.

Quality of movement is another area that has significant room for future exploration. The success of my explorations with qualities inspired by Laban's theories of Effort lead me to believe that use of movement qualities has the potential to provide rich continuous control and intuitive connections between the type of movement and the resulting output. My integration of Laban's theories could be immediately expanded by incorporation of the qualitative axis of Space, given an absolute frame of reference between the performer's body and the performance space. Additionally, Laban's framework, while a useful starting point in these explorations, may not be the best descriptive system of movement dynamics; however, there are few other existing systems. It may be necessary to develop entirely new frameworks for the description of movement dynamics and qualities, frameworks that may vary for different applications of the Gestural Media Framework (such as dance performances, musical compositions, interactive installations).

Perhaps, for greatest flexibility in mapping systems, it would be useful for the framework to not only incorporate specific movement qualities, but also to provide some access for defining one's own important dynamic and quality specifications and how they should be determined from the incoming movement data. This would be similar to the flexibility currently in the framework to define individual gesture vocabularies for specific pieces or users. That flexibility of training the system on a specific gesture vocabulary proved very important in the development of *Four Asynchronicities,* as I was free to discover what gestural vocabulary was relevant and important in the context of the movement that I was creating, rather than having to figure out how to incorporate key gestures from a predefined vocabulary that had nothing to do with any of the four pieces I was choreographing.

With the knowledge I gained from *Four Asynchronicities,* I am currently continuing to develop Movement 2, the solo, attempting to push the interaction with the system much farther. While many aspects of Movement 2 were successful, it is clear that there needs to be a more complex and sophisticated sound design that can be manipulated at a greater level of control by the performer. In particular, I am seeking to extend the system-as-instrument paradigm for this piece. The reworked piece will have a similar movement vocabulary and soundscape as the original version, but with much greater flexibility. The performer needs to be able to affect the sound in subtle and detailed ways, rather than primarily triggering samples with their own arcs. Every movement or change in movement quality that the performer makes should be able to have some result on the sound; the previous design of triggers and limited general shaping was not sufficiently reactive and subtle. I am developing this second version as a partially improvisational performance piece for myself. Interestingly, in my own work wearing the sensor suit and interacting with the media mappings, I have found strengths and weaknesses in the mappings that were not immediately obvious when I was the choreographer observing the connections between movement and sound. I am continuing to refine this solo work and the instrumental metaphor and will explore how far the gestural system as currently implemented can be used to give a performer quite nuanced control.

Additionally, this system and the theories behind it will likely be used in future development of the interactive performance aspects of Tod Machover's *Death and the Powers.* I am continuing to work on sensor systems and interaction design for the Disembodied Performance system and other interactive wearable systems for this opera, and the specifics of these interactions and the relationship of performer movement to output are still being shaped and explored. As this opera centers around a performer who is offstage for most of the show, it is necessary to continue developing ways in which this offstage singer's performance can be measured and interpreted in meaningful and powerful ways. In fact, some of Laban's qualities of motion are surprisingly related to some aspects of the performer's motion that we were already measuring (though under different, lower-level names) in earlier work on the Disembodied Performance system, as described in Chapter 3. Flow can be associated with the rugosity of the movement, Weight with the overall amplitude of the movement, and Time with the rate of change of the movement.

I have primarily discussed the Gestural Media Framework in the context of an instrumental paradigm, where the performer controls media elements in a deterministic and sophisticated manner through his gestures and qualities of movement. In the works in *Four Asynchronicities,* the media outputs (whether serving as a solo instrument or as an accompaniment to a group's interactions) were directly shaped by the movement abstraction input, and had no intelligent behavior or particular goals of their own. It is also useful to note that the abstracted gestural information in this framework could be used as inputs into an artificially intelligent system, such that the output media would be affected by the performers' movement, but also exhibit behavior of its own that was not directly tied to the movement. For example, intelligent agents like those Downie has developed for a variety of productions [22] or like Sparacino's Media Actors [68] could be programmed to take in gesture and quality of movement data as part of their perception layer and consider this information when they decide what actions to take next. This would create a duet model between performer and computer system.

That this framework could also be quite useful outside of the field of performance in informing new design tools for interactive installations, human-computer interaction, or storytelling. The basic four-layer design model of my Gestural Media Framework– movement inputs, gesture and quality abstractions, mapping tools, output media – is equally applicable for a variety of scenarios other than strict performance settings. For example, the movement of an audience member/viewer is often used as input into interactive sound or visual installations. What if the designers of those installations were able to explore and develop how the viewer's behavior would shape the installation experience by looking at the higher-level gestures and movement qualities explored in the Gestural Media Framework? How much richer could installation experiences become by incorporating concepts of quality of movement along with specific recognized movements? Similarly, in human-computer interaction scenarios, what if interaction designers had tools that helped them focus on the significance of how the user was performing a specific communicative gesture, not just recognize the fact that the user had performed that gesture?

As we continue to explore technologies for recognizing, quantifying, and employing gesture, I hope that these technologies can focus on and take advantage of the expressive, finely nuanced capabilities of human movement. The Gestural Media Framework is a step toward this kind of work, as it develops a methodology for describing and recognizing qualities of motion; creates a format for high-level abstraction of movement data, quality of movement, and gesture recognition algorithms into meaningful gestural content; and contextualizes the system in the demanding area of live performance. The subtleties and evocative details of physical movement can then be used to create even richer interactions, performances, and experiences: experiences that use and benefit from digital technology, but which are still inexorably linked to the extraordinary presence and expressive capabilities of the human body.

Bibliography

[1]ANGESLEVA, J., OAKLEY, I., HUGHES, S. & O'MODHRAIN, S. Body Mnemonics: Portable device interaction design concept. In *Proceedings of UIST'03*, Vancouver, Canada. 2003.
[2]ASHBROOK, D. and STARNER, T. MAGIC: A Motion Gesture Design Tool. In *Proceedings of SIGCHI conference on Human Factors in Computing Systems (CHI)*, Atlanta, GA, 2009.
[3]AVILÉS-ARRIAGA, H. H. and SUCAR, L. E. Dynamic Bayesian Networks for Visual Recognition of Dynamic Gestures. Journal of Intelligent and Fuzzy Systems. 12, 3,4 (Dec. 2002). pp 243-250.
[4]AYLWARD, R. and PARADISO, J. Sensemble: A Wireless, Compact, Multi-user Sensor System for Interactive Dance. In *Proc. New Interfaces for Musical Expression*. 2006. pp. 134-139.
[5]BENBASAT, A. An Inertial Measurement Unit for User Interfaces. MS Thesis. Media Arts and Sciences, Massachusetts Institute of Technology, 2000.
[6]BILLON, R., NÉDÉLEC, A., and TISSEAU, J. Gesture Recognition in Flow Based on PCA Analysis using Multiagent System. In *Proceedings of the 2008 international Conference on Advances in Computer Entertainment Technology.* ACE '08, vol. 352. ACM, New York, NY, 2008. pp 139-146.
[7]BOKOWIEC, M. and WILSON-BOKOWIEC, J. "The Suicided Voice," "Etch," in *Proc. of the 2008 Conf. on New Interfaces for Musical Expression.*
[8]BOKOWIEC, M. and BOKOWIEC, J. The Suicided Voice: The Mediated Voice. In *Proceedings of the International Computer Music Conference.* Barcelona 2005, pp. 459 – 462.
[9]BONGERS, A. J. Tactual Display of Sound Properties in Electronic Musical Instruments. *Displays*, 1998, pp 129-133.
[10]BONGERS, B. Physical Interfaces in the Electronic Arts: Interaction Theory and Interfacing Techniques for Real-Time Performance, in *Trends in Gestural Control in Music*, M. Wanderley and M. Battier, Eds. Paris: IRCAM, 2000.
[11]BREAZEAL, C. Emotion and sociable humanoid robots. In *International Journal of Human-Computer Studies 59*. 119–155. 2003.
[12]BROMWICH, M. and WILSON, J. "Bodycoder": A Sensor Suit and Vocal Performance Mechanism for Real-Time Performance, in *Proc. of the 1998 International Computer Music Conf.*, pp 292-295.
[13] CAMURRI, A. et al. EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance and Music Systems. In *Computer Music Journal*, 24:1, pp. 57-69. Spring 2000.
[14]CAMURRI, A., et al. KANSEI analysis of dance performance. *In Proceedings of the International Conference IEEE Systems Man and Cybernetics,* Tokyo, 1999.
[15]CAMURRI, A., CANEPA, C., et al. Social Active Listening and Making of Expressive Music: The Interactive Piece *the bow is bent and drawn*. In *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*. ACM, New York, NY, pp. 376-383. 2008.
[16]CHARLES, J. A Tutorial on Spectral Sound Processing Using Max/MSP and Jitter. In *Computer Music Journal*, Fall 2008. pp. 87-102.
[17]CONIGLIO, M. The Importance of Being Interactive. *New Visions in Performance.* s.l.: Taylor & Franci, 2004, pp. 5-12.
[18]COTTIN, R. File:Laban-effort-graph.jpg – Wikimedia Commons. [Web Site], accessible at http://commons.wikimedia.org/wiki/File:Laban-effort-graph.jpg.
[19]Cycling '74 || New Tools for Media. [Web Site], accessible at http://www.cycling74.com/.
[20]DELAUMOSNE, SHAW A., DELSARTE, F., et al. *Delsarte System of Oratory*. E.S. Werner,

New York. 1893.
[21]DIXON, Steve. *Digital Performance: A History of New Media in Theater, Dance, Performance Art, and Installation.* MIT Press, Cambridge, 2007.
[22]DOWNIE, M. Choreographing the Extended Agent: Performance Graphics for Dance Theater. Doctoral Thesis. Media Arts and Sciences, Massachusetts Institute of Technology, 2005.
[23]DOWNIE, M. Field: A New Environment for Making Digital Art. In *Computers in Entertainment*, pp. 1-34. December 2008. FAGERBERG, P., STÅHL, A., and HÖÖK, K. Designing Gestures for Affective Input: an Analysis of Shape, Effort and Valence. In P*roceedings of Mobile Ubiquitous and Multimedia,* Norrköping, Sweden. 2003.
[24]EICKELER, S., KOSMALA, A., and RIGOLL, G. Hidden Markov Model Based Continuous Online Gesture Recognition. In *Proceedings of the 14th International Conference on Pattern Recognition.* Volume 2. IEEE Computer Society, Washington, DC, 1998. pp 1206.
[25]FAGERBERG, P., STÅHL, A., and HÖÖK, K. Designing gestures for affective input: an analysis of shape, effort and valence. In *Proceedings of the 2nd International Conference on Mobile Ubiquitous Multimedia* (MUM 2003), Norrköping, Sweden, 2003.
[26]FERNANCES, C. *Pina Bausch and the Wuppertal Dance Theater: The Aesthetics of Repetition and Transformation.* Peter Lang, New York, 2001.
[27]GOLDBERG, M. Trisha Brown. *50 Contemporary Choreographers.* Routledge, London, 1999.
[28]GRADINGER, M. Pina Bausch. *50 Contemporary Choreographers.* Routledge, London, 1999.
[29]GRAHAM, M. *Blood Memory*. Doubleday, New York, 1991.
[30]GUEST, A. H. *Dance Notation: The process of recording movement on paper*. Dance Horizons, New York, 1984.
[31]HARTMANN, B., ABDULLA, L., MITTAL, M., KLEMMER, S. Authoring Sensor Based Interactions Through Direct Manipulation and Pattern Matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, New York, NY, 145-154.
[32]HEWITT, I. *Lost Highway:* Into the Dark Heart of David Lynch –Telegraph. [Web Site], accessed 3/2010. Available: http://www.telegraph.co.uk/culture/music/opera/3672082/Lost-Highway-into-the-dark-heart-of-David-Lynch.html
[33]HTK Hidden Markov Model Toolkit home page. [Web Site] Available: http://htk.eng.cam.ac.uk/
[34]JAMES, J., INGALLS, T., et al. Movement-based interactive dance performance. In *Proceedings of the 14th Annual ACM international Conference on Multimedia.* ACM, New York, NY, 2006. pp 470-480.
[35]JESSOP, E. The Vocal Augmentation and Manipulation Prosthesis (VAMP): A Conducting-Based Gestural Controller for Vocal Performance. In Proceedings *of NIME*. Pittsburgh, 2009.
[36]KELLY, D., et al. A framework for continuous multimodal sign language recognition. In *Proceedings of the 2009 International Conference on Multimodal Interfaces.* ACM, New York, NY, 2009. pp 351-358.
[37]KIRN, P. "Émilie Simon, Making Homemade Sessions in Her Apartment" [Web Site], accessed 3/2010. Available: http://createdigitalmusic.com/2010/03/04/emilie-simon-making-homemade-sessions-in-her-apartment/
[38]KO, T., DEMIRDJIAN, D., and DARRELL, T. Untethered gesture acquisition and recognition for a multimodal conversational system. In *Proceedings of the 5th International Conference on Multimodal Interfaces.* ACM, New York, NY, 2003. pp147-150.

[39]KRIS, J. jklabs:: maxlink, [Web Site], 2008. Available: http://jklabs.net/maxlink/.
[40]LABAN, R. *Mastery of Movement*. 4th edition, Northcote House. 1980.
[41]LEVIN, G. Painterly Interfaces for Audiovisual Performance. MS Thesis, Massachusetts Institute of Technology. 1994.
[42]Life Forms/Dance Forms 3D Animation Gallery. [Web Site]. Accessible: http://www.credo-interactive.com/gallery/index.html
[43]MACHOVER, T. Hyperinstruments: A Progress Report, 1987-1991, MIT Media Laboratory, 1992.
[44]MACHOVER, T. Shaping minds musically. *BT Technology Journal*, Vol 22 No 4, pp 171-179.
[45]MARRIN, T. and PICARD, R. The 'Conductor's Jacket': A Device for Recording Expressive Musical Gestures, in *Proc. of the 1998 International Computer Music Conf.*, pp 215-219.
[46]MARRIN, T. Inside the Conductor's Jacket: Analysis, Interpretation, and Musical Synthesis of Expressive Gesture. Ph.D. Dissertation, Massachusetts Institute of Technology, 2000.
[47]MAZO, J. H. *Prime Movers: The Makers of Modern Dance in America*, 2nd ed. Princeton Book Company, Hightstown, 1977.
[48]MERRILL, D. FlexiGesture: A sensor-rich real-time adaptive gesture and affordance learning platform for electronic music control. M.S. Thesis, Massachusetts Institute of Technology, 2004.
[49]MIT Media Laboratory, Opera of the Future. Death and the Powers. [Web Site]. Available: http://powers.media.mit.edu.
[50]MODLER, P. and MYATT, T. A Video System for Recognizing Gestures by Artificial Neural Networks for Expressive Musical Control. In CAMURRI, A., VOLPE, G., editors, *Gesture-Based Communication in Human-Computer Interaction*, *5th International Gesture Workshop, GW 2003, Genova, Italy, April 15-17, 2003, Selected Revised Papers.* Volume 2915 of *Lecture Notes in Computer Science.* Springer, 2003. pp 541-548.
[51]oblong industries, inc. [Web Site], accessed 3/10. Available: http://oblong.com/
[52]OLIVER, W. The Singing Tree: A Novel Interactive Musical Experience. M.S. Thesis, Massachusetts Institute of Technology, 1997.
[53]OLIVERO, J. and PAIR, J. Design and Implementation of a Multimedia Opera. In *Proceedings of the 1996 International Computer Music Conference*.
[54]opensoundcontrol. [Web Site], accessed 4/10. Available: http://opensoundcontrol.org/
[55]OVERHOLT, D., et al. A multimodal system for gesture recognition in interactive music performance. In *Computer Music Journal*. 33, 4 (Dec. 2009). pp 69-82.
[56]PARADISO, J. and HU, E. Expressive Footwear for Computer-Augmented Dance Performance. In *Proc. of the Int. Symp. on Wearable Computers*, Cambridge, MA, 1997. pp 165-166.
[57]PARADISO, J., HSIAO, K., HU, E. Interactive Music for Instrumented Dancing Shoes. In *Proceedings of the 1999 International Computer Music Conference*, October 1999, pp. 453-456.
[58]PARKER, M. Benesh Movement Notation. In *Dance Notation for Beginners*. Dance Books Ltd, London, 1984.
[59]RABINER, L. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, 77:257–286*, February 1989.
[60]ROKEBY, D. David Rokeby: Very Nervous System. [Web Site], accessed 4/10. Available: http://homepage.mac.com/davidrokeby/vns.html
[61]ROWE, R. *Interactive Music Systems: Machine Listening and Composing*. The MIT Press, Cambridge, MA, 1993.

[62]SAFFER, D. *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. O'Reilly Media, Inc., CA. 2009.
[63]SCHLÖMER, T., et al. Gesture Recognition with a Wii Controller. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction.* ACM, New York, NY, 2008. pp 11-14.
[64]SCHMIDT, J. Pina Bausch. *Pina Bausch und das Tanztheater Wuppertal.* Müller + Busmann, Wuppertal, 2002.
[65]SIEGEL, W. and JACOBSON, J. The Challenges of Interactive Dance: An Overview and Case Study. *Computer Music Journal*, Vol. 22, pp. 29-43, 1998.
[66]SONAMI, L. "Lady's Glove," [Web Site], accessed 3/2010. Available: http://www.sonami.net/lady_glove2.htm.
[67]SPARACINO, F. et al, Augmented Performance in Dance and Theater, *International Dance and Technology 99*, Arizona State University, Tempe, AZ.
[68]SPARACINO, F., DAVENPORT, G., PENTLAND, A. Media Actors: Characters in Search of an Author. In *Proceedings of the International Conference on Multimedia Computing and Systems, Firenze, Italy, June 1999*.
[69]STARNER, T., WEAVER, J., and PENTLAND, A. A Wearable Computer Based American Sign Language Recognizer. In A*ssistive Technology and Artificial intelligence, Applications in Robotics, User interfaces and Natural Language Processing* MITTAL, V. O., et al., Eds. *Lecture Notes In Computer Science*, vol. 1458. Springer-Verlag, London, 1998. pp 84-96.
[70]STEIM. Products, BigEye. STEIM. [Web Site]. Available: http://www.steim.org/steim/bigeye.html.
[71]STOPPIELLO, D. and CONIGLIO, M. Fleshmotor. [ed.] Judy Malloy. *Women, Art, and Technology.* MIT Press, Cambridge, 2003. pp. 440-452.
[72]STRACHAN, S., MURRAY-SMITH, R., and O'MODHRAIN, S. BodySpace: inferring body pose for natural control of a music player. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems.* ACM, New York, NY, 2007. pp 2001-2006.
[73]TORPEY, P. Disembodied Performance: Abstraction of Representation in Live Theater. M.S. Thesis. MIT Media Laboratory, 2009.
[74]TORPEY, P. and JESSOP, E. Disembodied Performance. *Proceedings of ACM CHI Conference on Human Factors in Computing Systems.* Boston, 2009.
[75]TROIKARANCH. YouTube – 16 [R]evolutions – About the Technology. [Web Site]. Available: http://www.youtube.com/watch?v=Rbv7n0ZgA98.
[76]TroikaTronix – Isadora. [Web Site]. Available: http://www.troikatronix.com/isadora.html.
[77]VICKERY, L. The Yamaha MIBURI MIDI Jump Suit as a Controller for STEIM's Interactive Video Software Image/ine. *Proc. Australian Computer Music Conference*. 2002.
[78]WAKIN, D. Techno-Alchemy at the Opera: Robert Lepage Brings his "Faust" to the Met. *New York Times*. November 7, 2008, p. C1.
[79]WANDERLY, M. *Performer-Instrument Interaction: Applications to Gestural Control of Sound Synthesis.* PhD Thesis, University Paris VI. June 2001.
[80]WEINBERG, G., AIMI, R., and JENNINGS, K. The Beatbug Network -- A Rhythmic System for Interdependent Group Collaboration. *Proceedings of the Conference on New Instruments for Musical Expression (NIME).* Dublin, 2002.
[81]WESTEYN, T. et al. Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition. *Proceedings of ICMI 2003.*
[82]WILKINSON, S. Phantom of the Brain Opera. *Electronic Musician.* January 1997.

[83]WINKLER, T. Fusing Movement, Sound, and Video in *Falling Up*, an Interactive Dance/Theater Production. *Proc. New Interfaces for Musical Expression, Session: Demonstrations.* 2002. pp 1-2.
[84]YOUNG, D. The Hyperbow Controller: Real-Time Dynamics Measurement of Violin Performance. *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*. Dublin, 2002.
[85]YOUNG, S. et al. *The HTK Book (for HTK Version 3.4).* Available: http://htk.eng.cam.ac.uk/docs/docs.shtml
[86]ZHANG, X., et al. Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors. In *Proceedings of the 13th International Conference on Intelligent User Interfaces.* ACM, New York, NY, 2009. pp 401-406.