THE BODY PARAMETRIC

Abstraction of Vocal and Physical Expression in Performance

Elena Jessop Nattinger

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

Master of Science in Media Arts and Sciences
Massachusetts Institute of Technology, 2010

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences at the
Massachusetts Institute of Technology
September 2014

Author:____
Elena Jessop Nattinger
Program in Media Arts and Sciences
August 8, 2014

Certified By:____
Tod Machover
Muriel R. Cooper Professor of Music and Media
Program in Media Arts and Sciences
Thesis Supervisor

Accepted By:____
Patricia Maes
Alexander W. Dreyfoos Professor of Media Technology
Interim Academic Head, Program in Media Arts and Sciences

THE BODY PARAMETRIC
Abstraction of Vocal and Physical Expression in Performance

Elena Jessop Nattinger

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on August 8, 2014,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences at the
Massachusetts Institute of Technology
September 2014

Abstract

Performing artists have frequently used technology to sense and extend the body's natural expressivity via live control of multimedia. However, the sophistication, emotional content, and variety of expression possible through the original physical channels of voice and movement are generally not captured or represented by these technologies and thus cannot be intuitively transferred from body to digital media. Additionally, relevant components of expression vary between different artists, performance pieces, and output modalities, such that any single model for describing movement and the voice cannot be meaningful in all contexts. This dissertation presents a new framework for flexible parametric abstraction of expression in vocal and physical performance, the Expressive Performance Extension Framework. This framework includes a set of questions and principles to guide the development of new extended performance works and systems for performance extension, particularly those incorporating machine learning techniques. Second, this dissertation outlines the design of a multi-layered computational workflow that uses machine learning for the analysis and recognition of expressive qualities of movement and voice. Third, it introduces a performance extension toolkit, the Expressive Performance Extension System, that integrates key aspects of the theoretical framework and computational workflow into live performance contexts. This system and these methodologies have been tested through the creation of three performance and installation works: a public installation extending expressive physical movement (the Powers Sensor Chair), an installation exploring the expressive voice (Vocal Vibrations), and a set of performances extending the voice and body (*Crenulations and Excursions* and *Temporal Excursions*). This work lays the groundwork for systems that can be true extensions of and complements to a live performance, by recognizing and responding to subtleties of timing, articulation, and expression that make each performance fundamentally unrepeatable and unique.

Thesis Supervisor: Tod Machover
Title: Muriel R. Cooper Professor of Music and Media

THE BODY PARAMETRIC

Abstraction of Vocal and Physical Expression in Performance

Elena Jessop Nattinger

The following person served as a reader for this thesis:

Thesis Reader____
Sile O'Modhrain
Associate Professor of Performing Arts Technology
University of Michigan

THE BODY PARAMETRIC

Abstraction of Vocal and Physical Expression in Performance

Elena Jessop Nattinger

The following person served as a reader for this thesis:

Thesis Reader____
Marc Downie
Artist and Cofounder, OpenEndedGroup

Acknowledgements

This work would not have been possible without the contributions of many others. To those who have guided me, inspired me, pushed me, and written code with me in theaters at 2AM, thank you.

TOD MACHOVER
Thank you for welcoming me into your Media Lab family during these last six years, and profoundly shaping this section of my path. I am so thankful for your guidance and inspiration, as well as for the opportunities you have given me to grow my skills, to share in exciting and groundbreaking projects, to create new work, and to figure out how to combine all of my fields of interest.

SILE O’MODHRAIN, MARC DOWNIE
Thank you to my thesis readers for your time, energy, feedback, and tremendous enthusiasm. I am very grateful for our many thoughtful conversations and explorations of ideas, which have been essential in developing my theories and my work on performance systems.

KAROLE ARMITAGE, TERESA NAKRA
To my general exam committee, thank you so much for helping me explore and refine my ideas.

PETER TORPEY
Thank you, always, for your friendship, for our collaboration, and for your boundless support in all things. I am so grateful and honored to have shared this Media Lab journey with you.

OPERA OF THE FUTURE GROUP
I have been fortunate to work closely with and be inspired by many amazing people in the Opera of the Future group. To present and past members of Opera of the Future, thank you for all of the interesting discussions, creative and technical brainstorming, international adventures, and late nights hard at work.

MIT MEDIA LAB FACULTY, STAFF, STUDENTS
Thank you for your wisdom, creativity, and playfulness. Thank you for helping make the Media Lab the place it is, and the best place for me to be.

COUNCIL FOR THE ARTS AT MIT, FUTURUM ASSOCIATION, PUNCHDRUNK, LE LABORATOIRE, THE PEROT MUSEUM OF NATURE AND SCIENCE, THE DALLAS OPERA
Some of the work presented in this thesis was created in collaboration with and supported by organizations worth special acknowledgement. I am grateful for the opportunities and shared ideas.

ELIZ, CHRIS, JAMES, ROSS, DONA
To my friends outside the Lab, thank you for your companionship and encouragement.

MOM, DAD, NOAH, GRANDMA, KEVIN
Thank you, always, to my family for your constant love, support, and confidence. I am so fortunate to have you.

Table of Contents

Table of Figures

Unless otherwise noted in captions, all figures are by the author.

1. Introduction: Expressive Extension of Physical and Vocal Performance

1.1. New Visions of Performance

The human body and voice are two powerful instruments that every person possesses. They are infinitely expressive, extremely personal, and deeply compelling. Indeed, the majority of our performance traditions center on performers' ability to communicate emotions, evoke experiences, and take the audience on a journey through their bodies and their voices. When a pianist thrusts his hands down on the keys, a dancer sinuously curves his body through space, or an opera singer softly glides to the top of a melodic phrase, their physical actions send meaningful, metaphorical, and emotional information to the audience. With the integration of new technologies into live performance, artists have frequently attempted to use technology to sense and extend the body's natural expressivity into live control of a range of digital media: rich sounds, compelling visualizations, rapidly changing lighting, and even robotic movement. However, with this increasing use of technologies in performance contexts, the expressive power of the live human performer is at risk. The sophistication, emotional content, and variety of expression possible through the original physical channels are generally not captured or represented by these technologies, and thus cannot be intuitively transferred from body to digital media. Why should a performer stay behind a laptop, controlling multimedia through the standard interaction paradigms of a computer, or be overwhelmed by giant projections and sound that have no direct connection to the live performance? What tools and techniques are needed to create technologically-enhanced performance works that not only retain but actually enhance the expressivity of a live performer? To envision some of the risks and opportunities of new technologies in performance, let us imagine two rather different kinds of performances.

In the first performance, a man stands at a table onstage behind his laptop computer. A projection screen behind him shows colors and shapes. The man stares at his computer screen while speakers on either side of the stage emit low sounds and melodies. As the man mouses, clicks, and types, the patterns of sound and light change. Sometimes, the sound changes when the man appears to be doing some action. Occasionally, his actions do not cause any obvious change. Other times, sound or projection patterns seem to change while he is not doing anything. Although the musical and visual results of this performance may be interesting, the connection between the performer and the performance is unclear. The audience knows, academically, that the piece is being created or at least controlled live, but the connection is hidden. It is not clear what portion of the audience's experience has anything to do with the performer's actions; indeed, the entire production could have been pre-composed and played back with a few clicks of the mouse.

Now let us imagine a second performance with similar visual and sonic content. The projection screen displays shifting colors, while a soundscape of textures and melodies plays over the speakers. In this performance, a man stands alone onstage facing the audience. As he stands completely still, the screen grows static and dim and a quiet low drone is the only sound in the space. With a sharp flick of his hand, colored shapes dance across the screen and a high note plays. A smooth sweep of his arm and curl of his hand expands and morphs the drone texture. Gentle beats of his hand are echoed by an emerging rhythm. When he wraps his hand into a fist, the music grows louder and harsher, pausing at this crescendo until he relaxes his hand and it regains its original rhythm. He begins to rapidly, nervously shake his hands, and the rhythmic pattern becomes increasingly unstable while the images grow brighter. He stops suddenly with his hands up, breath held: the projected color stays bright and steady and the sound maintains a loud drone. As he drops his hands slowly, the colors fade and the sound diminishes to a whisper. In this second performance, the behavior and development of the digital media appear continually connected to the performer's body. The fluctuations of the performer's timing, intensity of movement, and smoothness or sharpness, seem to shape the sonic and visual material. The audience's focus is on the physical expressivity of the live performer and the ways that expressivity is extended into digital domains.

How can computational systems support artists in extending the body's natural expressivity into the control of multimedia elements? How can expressive qualities of movement and the voice be described and represented? What tools and methodologies are necessary for artists to create performances and installations with media extensions that are shaped by a live performance in rich, thoughtful, and evocative ways? These are questions that this dissertation seeks to address.

1.2. Thesis Objectives: Framework, System, Examples

The research described in this thesis is designed to support performances and installations that use technologies to capture and extend physical and vocal performance into digital media. It provides guidelines and systems for creating such performances and for developing compelling connections between physical behavior and output media. Through my prior research, I have found that current computer systems lack the kind of high-level analysis of physical and vocal qualities that would support artists in taking advantage of movement or vocal data to augment a performance work in meaningful ways. Additionally, current analysis systems are not flexible enough to handle the expressive variation between different artists, performance pieces, and output modalities. By applying machine learning techniques to expressive analysis of movement and voice, the work presented here aims to help the field of performance and technology overcome some of these hurdles.

This research on technologically extended performance has several primary components:

- The Expressive Performance Extension Framework, a new framework of parametric abstraction of expression in vocal and physical performance
- A collection of guidelines, questions, and principles to assist in the development of new extended performance works and systems for performance extension
- A multi-layered computational workflow that incorporates machine learning for the analysis and recognition of expressive qualities of movement and voice
- A set of expressive parametric axes for describing vocal and physical qualities
- The Expressive Performance Extension System, a toolkit for live performance that incorporates key aspects of the theoretical framework and computational workflow
- New performance and installation works centered on extending expressive physical movement and the expressive voice

The goal of the Expressive Performance Extension Framework is to analyze and recognize continuous qualities of movement rather than to perform traditional gesture recognition. Similar goals are true in a vocal context, where the aim is to recognize expressive vocal qualities, not to recognize particular notes like a score-following system, or particular words like a speech recognition system. Though gesture and speech recognition could be added to this framework at a later stage, this research focuses on the aspects of expression in performance that can be captured and extended purely through the analysis of continuous qualitative parameters.

While users should be able to define their own sets of expressive axes that they find most meaningful for a particular performance work, I also provide a core set of axes developed through my research and exploration that can serve as a useful starting point for expressive analysis. This example set of expressive parametric axes includes six parameters: *energy* (calm to energetic), *rate* (slow to quick), *fluidity* (legato to staccato), *scale* (small to large), *intensity* (gentle to intense), and *complexity* (simple to complex). These axes can be combined to form a high-level expressive space. Other potential sets of expressive axes could include Laban's effort model of *weight*, *time*, *space*, and *flow*, as well as frameworks for describing emotion through parameters such as *arousal*, *valence*, and *stance*. Positions in and trajectories through any selected parametric space can then be mapped to control values for multimedia. The Expressive Performance Extension Framework includes guidelines for the process of selecting axes that are meaningful for a particular work.

Throughout this work, I seek to examine several key questions in the field of technological extension of physical performance. How can raw sensor data be abstracted into more meaningful descriptions of physical and vocal expression? What features of physical performance can convey particular expressive and emotional content? How can we create evocative high-level descriptions of movement and voice so that they can be used intuitively and creatively in the process of choreographing, composing, and performance-making? How can we create tools that encourage metaphorical, meaningful, and rich associations between movement and media, rather than naïve one-to-one sensor to output mappings? What principles should systems for performance extension follow in order to be easily incorporated into existing creative processes? What are good practices for extending live physical and vocal performance through machine learning techniques?

1.3. The Current State of Interactive Performance Systems: Data is Easy, Information is Hard

We are now at a point in the practice of technologically-extended performance work where it is relatively straightforward to collect substantial amounts of data about a live performance, not only through specialized devices such as motion capture systems and wearable sensors, but also through readily available and affordable systems such as webcams, microphones, and even the Microsoft Kinect. Even wearable sensor systems have become affordable and easy to implement, through the prevalence of microcontroller boards such as Arduinos and wireless devices such as XBee modules. However, while our sensing systems have become increasingly precise, cheap, and easy to use, analysis systems have not made comparable progress in determining the expressive significance of all of this performance data. How do we make sense of a performer's movement? Indeed, what does it mean to "make sense of movement" in an expressive context? As the field has moved past the question of how to *sense* data about a physical performance, we can explore how to best *interpret* that data to turn it into useful information and use it to create compelling performance work.

For the continuing practice of technologically-extended expressive performance and installations, it is vital that we step away from the details of specific sensor setups and data streams to shape these works through more meaningful descriptions of performance expression. Imagine a system that you can teach how you move and sing. In performance, it knows when you are moving in a tiny, delicate manner, when you make a sharp and bold movement, how gently and smoothly you are singing, or how energetically you are singing. Now imagine you are asked how you would like to use the information given by the system to control and shape music or sound, or a generative visualization, or theatrical lighting, or other digital media in a performance. While your answers would likely vary in different performance contexts, it is easy to begin imagining interactive relationships between the body and media. In contrast, most existing performance capture systems speak in terms of the values of individual data streams (e.g. the XYZ coordinate of your right hand, the amplitude of your voice, the amount of acceleration along the X axis of the accelerometer mounted on your arm) and low-level features analyzed from this data (e.g. the derivative of your hand's position, your voice's average volume over the last second, how many pixels changed in a particular region of the webcam since the previous frame). Which kinds of information do you think would be more helpful or inspirational in your creative process of mapping your physical performance input to your desired digital output? I argue that the more closely that the input parameters you are given by the system describe expressive aspects of a performer's movement or voice that are relevant for a given performance, the easier and more intuitive it is to create meaningful relationships between the performance and the digital extension of that performance.

And what if you could pick your own set of sensors and your own output media? If you could select, define, and refine your own set of qualitative descriptions in rehearsal as you develop the performance content of a piece? If you could easily change your sensors, select and train a new set of expressive descriptors, adjust mappings, go back to yesterday's model? Each performance creator needs this kind of flexibility in technologies, definitions of expression, and relationships between sensing and output to create interactions that are meaningful in the context of a specific performance or installation.

While there exist a variety of systems and frameworks designed for digitally enhancing performances, none of these systems yet allow the amount of flexibility and high-level expressive description envisioned here. The majority of the systems for mapping some performance input to a digital output do not incorporate any definition of higher-level expressive parameters. Those that do have some conception of expression, such as EyesWeb (Ricci, Suzuki, Trocca, & Volpe, 2000), generally limit their definitions and analysis to predefined sensing setups and sets of descriptions. A variety of systems incorporate techniques from the machine learning repertoire for gesture recognition or note identification, but these sorts of binary recognition processes are not sufficiently conducive to sophisticated continuous control of media.

Additionally, most mapping systems are structured to facilitate the typical approach of mapping a particular input sensor dimension to a particular output control value. This methodology seems more suitable for technologists wanting to create artistic work than for artists hoping to integrate technology into their performances and installation works. Certainly, it is possible to make rich mappings by carefully choosing a gestural or vocal vocabulary, implementing sensor systems designed to detect that vocabulary, and writing software to process that specific data and associate it with the desired control values. I have used this design process in some of my early gestural analysis projects, such as the Vocal Augmentation and Manipulation Prosthesis, a glove that allows a singer to manipulate his own voice (Jessop, 2009). However, that design process required me to be a computer scientist and electronics engineer as well as a performance creator. Especially when a system incorporates machine learning algorithms, the majority of existing systems require high levels of technical knowledge to obtain desirable results. In order for those who are not primarily programmers or machine learning specialists to be able to create interesting mappings, it is necessary to abstract the meaningful movement and vocal information away from the specifics of sensors, streams of input data, and pattern recognition algorithms.

In addition, any definition of "meaningful movement and vocal information" is likely to vary between different performance-creators, between different performance and installation works, and perhaps even within different sections or moments of a particular piece. Similarly, different pieces will capture movement and vocal data through different sensing mechanisms, guided by other goals and constraints of the work. No pre-programmed or pre-trained system would be able to provide a sufficient qualitative movement or vocal description to satisfy every user's needs. Instead, such a system would give one fixed structure about how to describe movement and the voice. Ideally, systems should suggest ways to think about qualitatively analyzing the voice and the body in expressive contexts, but allow users flexibility in creating their own definitions and expressive models. While existing systems provide some of this flexibility for simple gesture recognition tasks, allowing users to define their own gestures, no existing system has provided the ability to flexibly define continuous, high-level expressive parameters.

The Expressive Performance Extension System (EPES) presented in this dissertation allows users to capture raw input data, compute expressive features, perform pattern recognition to identify desired vocal and physical qualities, and manually map information about these high-level expressive parameter spaces to output control parameters for digital media. No prior mapping systems have allowed a user to train that system to recognize specific continuous qualities of movement and voice, rather than classifying movement into emotional categories or into particular labeled gestures. Additionally, this architecture differs from existing work in its support for creative practice at a higher level of abstraction than either raw sensor data or computational feature spaces. I will not be focusing on quantitative evaluation metrics for the Expressive Performance Extension System, as in-depth quantitative evaluation would be impractical within the scope of this thesis. More valuably, and more relevantly given the context in which these systems are designed to be used, these systems and methodologies are evaluated in the tradition of qualitative analysis of artistic practice, particularly focusing on their utility and expressivity as artistic tools.

1.4. Outline of Thesis

This chapter has introduced the concept of expressively extending physical and vocal performance through technology and highlighted the need for systems that allow creators to flexibly define their own high-level descriptions of expression. It also has outlined the goals and deliverables of my research: a creative framework for performance extension, a computational system for high-level analysis and mapping of physical and vocal qualities, and several example performances and installations.

Chapter 2 places my research in a larger artistic and technical context, reviewing prior work on technological extension in live performances and public installations, particularly extension of movement and voice. It examines methodologies for movement and voice analysis and description from various domains including human-computer interaction, dance notation, gesture recognition, and digital musical instrument design. The concept of "mapping" and a variety of existing tools and strategies used to create mappings for interactive performance systems are also discussed. This chapter examines these existing models from multiple domains to derive important principles for performance extension technologies and to suggest necessary additions to the field. This chapter also provides definitions for the key concepts in this dissertation such as *gesture*, *quality*, and *expression*.

Chapter 3 analyzes some of my prior interfaces for expressive movement and vocal extension, with a particular focus on how features of these works suggest some principles for extended interface design. This section includes analysis of three primary projects: the Vocal Augmentation and Manipulation Prosthesis (VAMP), a wearable gesture-based instrument for a singer to control an extension of his own voice; the Gestural Media Framework, a system for abstraction of gesture recognition and Laban-inspired qualities of movement; and the Disembodied Performance System (DPS), a sensing, analysis, and mapping framework for extending the expressive behavior of an opera singer through transformations of sound and scenography.

Chapter 4 presents the Expressive Performance Extension Framework, a theoretical and technical framework for the technological extension of body and voice in performance and interactive installations. This includes a set of the key principles, guidelines, and necessary questions that should be considered by practitioners seeking to design technologically-extended performances, particularly those that incorporate machine learning techniques for movement and vocal analysis. This framework also outlines a workflow for incorporating machine learning of high-level expressive qualities into performance and rehearsal contexts. Through examination of the concepts of expression and liveness, and discussion of some of my additional projects at the Media Lab (including an online extension to the show *Sleep No More*, the interactive Chandelier Hyperinstrument in *Death and the Powers*, and Bibliodoptera, an interactive public art installation), I propose guidelines and best practices for incorporating technologies into performance and installation contexts and for designing systems to extend live performance.

Chapter 5 describes the Expressive Performance Extension System (EPES), a flexible system for sensing, analyzing, and mapping expressive parameters. This system incorporates the use of machine learning techniques to support mappings using abstract parametric qualities, allowing users to capture raw input data, compute expressive features, perform pattern recognition to identify desired vocal and physical qualities, and manually map information about these high-level expressive parameter spaces to output control parameters. EPES extends the mapping program designed for the Disembodied Performance System (Torpey, 2009) to concretely implement the design principles outlined in Chapter 4. This chapter outlines the structure and features of EPES, walks through the EPES workflow for defining and learning abstract qualities of movement and voice, and discusses the process of incorporating EPES into a performance or installation and its integration with other technical systems.

Chapter 6 describes and discusses the three key projects created for this dissertation research that have incorporated the Expressive Performance Extension System for analysis and extension of the voice, the body, and the body and voice together. The first of these is the Powers Sensor Chair, a movement-based instrument designed for the general public to shape their own musical experience with the sonic world of *Death and the Powers*. The second project outlined is Vocal Vibrations, a public art installation designed to encourage participants to explore their own voices and the vibrations generated by their voices, augmented by musical and tactile stimuli. By expanding the Opera of the Future group's work in technologies for sophisticated measurement and extension of the singing voice in performance, we aim to create new kinds of powerful vocal experiences in which everybody can participate. The third project incorporated in this thesis is a series of performance and installation pieces, *Crenulations and Excursions* and *Temporal Excursions*, where a performer's body and voice control a sonic environment. This chapter also discusses a variety of other works that have incorporated EPES, from the *Death and the Powers* global interactive simulcast and second-screen experience, to a short multimedia theatrical performance for the Hacking Arts Festival, to a series of performances and installations developed by Blikwisseling workshop participants in the Netherlands. This chapter addresses the design goals of these experiences, their development processes, the ways in which they incorporate the technologies and frameworks described in this dissertation, and the ways in which they illustrate various design principles for technical performance extension.

Chapter 7 summarizes this dissertation's contributions: a framework and set of guidelines for developing technologically extended performances and performance systems; a flexible software system for incorporating machine learning technologies into extended performances; and a set of new performance and interactive installation works centered on the expressive voice and the body. This chapter also addresses the next steps for this research, lessons from the research that may be relevant for other fields, and future directions for extended performance.

2. Background and Context

This chapter seeks to place my work in technologies for extending expressive physical performance into a larger artistic and technical context. It examines historical and recent work in interactive installation and performance systems, qualitative and quantitative systems for voice and movement analysis, the use of voice and body in human-computer interaction, and systems and strategies for mapping input sensor information to output media. It also defines the key terminology and concepts used in this dissertation, and highlights ways that the research presented in the remainder of the dissertation connects different areas of study and addresses issues in existing systems.

2.1. Terminology and Definitions

There have been many efforts to study and categorize bodily movements and their relationship to emotional effects or linguistic concepts. Many of these have focused on taxonomies of gesture, from Quintilian's advice on appropriate gestures for Ancient Roman orators (Kendon, 2004) to the systematized set of gestures in Delsarte's acting technique, developed from his observations of naturalistic movement (Delaumosne, 1893). Many gesture categorization techniques have also been created by researchers in the psychology of gesture and speech (Efron, 1972; McNeill, 1992). However, the definition of the term "gesture" is not precisely fixed. Kurtenbach and Hulteen define a gesture broadly as "a movement of the body that conveys information" (Kurtenbach & Hulteen, 1990). Adam Kendon uses "gesture" as the label for movement that appears intentionally communicative and deliberately expressive:

> "…if movements are made so that they have certain dynamic characteristics they will be perceived as figure against the ground of other movement, and such movements will be regarded as fully intentional and intentionally communicative... 'Gesture' we suggest, then, is a label for actions that have the features of manifest deliberate expressiveness." (Kendon, 2004)

Such movement is often identified by having sharp onsets and offsets and being a temporary excursion from a position. This implies that certain temporal and spatial characteristics indicate a meaningful gesture. In addition, the term "gesture" can also be extended past physical actions into musical and sonic domains: for example, Hatten broadly describes gesture as "any energetic shaping through time that may be interpreted as significant" (Hatten, 2006). Many examples of gestures in musical contexts can be found in (Godøy & Leman, 2009; Gritten & King, 2006).

Volpe defines "expressive gestures" as movements that convey particular types of information: expressive content or implicit messages. He also describes a method to broaden this definition through the use of technology: if a movement of the body results in expressive content through, say, music or visuals rather than only through the pure movement content, it is an "extended" expressive gesture. These "extended expressive gestures" are the result of a juxtaposition of several dance, music, and visual gestures. However, Volpe argues that they are not simply the sum of these gestures, since they incorporate the artistic view of a director and are perceived as multimodal stimuli by audiences (Volpe, 2003).

For the purposes of this dissertation, I will define a *gesture* to be a vocal or physical action that conveys information. A gesture is *what* a performer does. Similarly, I define *quality* as the elements of movement or voice that convey individual variation, such as dynamics, timbre, and timing. As described by Kendon and Hatten (Hatten, 2006; Kendon, 2004), these temporal (and spatial, where appropriate) elements delineate an expressive gesture. This is *how* a gesture or action is performed. The research presented in this dissertation does not focus on analyzing the semantic or emotional content contained in a particular gesture (raising the hand, for example), but instead focuses on the expressive content contained in the quality of that movement. The recognition and classification of gestures is not a focus of this framework. We are more interested in the *how* than the *what*.

I additionally define *expression* as emotional and evocative communication through movement or the voice. It is important to clarify, as highlighted in Juslin (2003), that expression is not a single-axis phenomenon of which a performance has "more" or "less," but a space outlined by multiple qualitative parametric axes. In this context, a *parameter* refers to a value that varies over time within a clearly-defined semantic space. This semantic space may be something that can be defined quantitatively, such as the numerical value range of a particular sensor data stream, or it may be qualitative, such as *complexity*, *rate*, or *intensity*. These qualitative parameters change over time though they may not have one obvious numerical value. For example, the rate of a performer's movement will vary throughout a performance. It is important to note that this definition differs from the standard mathematical definition of *parameter*, where parameters are values for configuring a mathematical model to produce particular results given particular inputs. This dissertation particularly focuses on *high-level expressive parameters*, metrics of movement and vocal quality whose definitions vary in different performance contexts, as discussed further in following chapters. Handles adjusted over time to control the shaping of output media are typically referred to as *output control parameters*.

As with *gesture*, *expression*, and *quality*, the concepts of *performance* and *installation* are quite broad and may have a variety of definitions. In this thesis, a *performance* is an expressive artistic presentation for an audience that is carried out in a specific space during a specific point in time and that varies with each presentation. A key aspect of this definition is the concept of an audience, as it limits the definition of *performance* to presentations that are observed. Another aspect of this definition to highlight is that a performance must vary with each presentation, it must be "different every night." This variation may be subtle or significant, from slight nuances of timing or articulation coming from a performer's internal state to complete changes of content, such as in the case of an improvisational performance. Importantly, this definition excludes works that have no temporal component to their experience (such as a painting or sculpture) and works that are experienced temporally but are fixed and unchanging with each presentation (such as screenings of film or video). I additionally define a performance to be a work that is observed during the same period of time in which it is presented, though it may take place or be experienced in different spaces simultaneously, as is the case for some of the remote performance extensions that are described in this dissertation.

The term *installation* will generally be used to refer to a space that is augmented artistically so that a visitor to the space becomes immersed and involved in the experience of observing and exploring the space. Frequently, a visitor to an *interactive installation* becomes not only an audience member observing the work but also a performer shaping the work through his actions. An interactive installation may be considered a particular type of performance. A specific performance or installation will be referred to interchangeably in this document as a *work*, a *piece*, or a *production.*

While many different kinds of technologies are currently incorporated into performance art and artistic installation contexts, the most relevant for this dissertation are technologies that *extend* a live performer or installation visitor's expressive actions into the behaviors of digital media. Such *performance extension technologies* are the core of this dissertation, and an *extended performance* incorporates these kinds of technologies. In these cases, the behavior of the technology is affected by or is shaped by a live human, whether or not that human is conscious of the manner in which the technology is under his control. Thus, I would consider a video projection in an artistic installation that changes its content depending on how quickly a visitor is walking through the installation space to be a technological extension of a performance. However, a video projection in a different installation that changes content based on the projected weather forecast would not be a technological extension of performance, despite the fact that it may be different every time it is experienced. Similarly, a digital musical accompaniment that is randomly generated according to a stochastic algorithm would not be considered a performance extension technology.

Finally, in almost all technologies for performance extension, the concept of *mapping* must be addressed. The standard mathematical definition of mapping is an operation that associates each element of a given set (the *domain)* to one or more elements of another set (the *range*). In the context of this dissertation, a *mapping* is the associations between some performance input and a desired output, and creating a mapping is the process of defining this associative space. This definition of mapping does not constrain the form of the input (sensor data, expressive parameters, random processes) or the form of the output. Mappings may be constructed by human performance-makers, learned or developed by a computer, or any combination of these methodologies.

2.2. Technologies for Extended Performance and Interactive Installations

As my research in movement and voice extension technologies is specifically designed for performance and installation scenarios, it is useful to locate this research in its broader artistic context. This section examines some prior work in technological extension for live performances and public installations, particularly the extension of movement and voice.

2.2.1. Technology in Performance

Throughout the history of performance, new technologies have been incorporated and explored to add to the expressive potential of a work, from electric lighting to digital video to networked systems to robotics. As Steve Dixon describes 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." (Dixon, 2007, p. 10)

Performance-makers have frequently sought out new technologies to incorporate into their productions. The Ancient Greek and Roman theaters were full of technological developments in mechanical stage engineering, such as the Greek mechanisms for lowering a performer from the heavens onto the stage (*deus ex machina*) and revolving prism-shaped set pieces to allow for quick scene changes (*periaktoi*). Simultaneously, performance practices have often been a driving force in technological development and adoption, as performance-makers push the boundaries of existing technologies. For example, the widespread adoption of electrical grids across the United States was due in part to the needs of theaters across the country that were experimenting with and incorporating electric lighting into their productions (Dixon, 2007).

The field of dance has also incorporated many early examples of technology in performance. Loie Fuller, one of the first modern dancers, created solo dance pieces that combined new electric lighting techniques with flowing costumes of her own design to produce never-before-seen visual effects that extended her live dance performance, transforming the shape and movement of her body by the way that the costumes caught the light. She even created a dance where her costume glowed, thanks to the use of radium (Mazo, 1977). While these technologies did not vary their behavior based on Fuller's performance, she created effects that relied on the interplay between technology and her body.

Today, computational technologies are present in theater, music, dance, and opera performances, as well as many performances that combine or fuse different types of performance practice. These technologies include sophisticated robotics and set mechanisms, complex theatrical lighting equipment, sound amplification and manipulation tools, multimedia projection, live cameras, digital music and instruments, and many other techniques. As will be discussed further in Chapter 4, live performance provides a particularly challenging use case for new technologies, requiring great flexibility, temporal precision, speed of response, and control over technological effects. In addition, technologies for performance, particularly for extending a physical performance, have to reliably act in a way expected by performers: their behaviors should be learnable, predictable, and repeatable.

As formerly-separate artistic modalities become more and more integrated, performances also have started to require the technological control of multiple synchronous media. As an example, Cirque du Soleil is known for its elaborate circus productions incorporating projection, lighting, sound, and robotic scenery. While much of this technology is non-interactive, Canadian director Robert Lepage's show *Totem* for Cirque du Soleil also incorporates interactive elements through projections that are generated in real time and affected by the movement of the performers. Infrared cameras positioned around the stage detect performers' movements so the system can produce "kinetic effects such as ripples, splashes and reflections in the water and the flames" ("TOTEM Set Design and Projections," n.d.).

Lepage has also included interactive technologies in opera productions. His 2008 staging of Hector Berlioz' *La Damnation de Faust* for the Metropolitan Opera uses microphones to capture the 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 is used to shape projected images in real time, such as projected curtains waving behind dancers and giant projected flames that vary based on a singer's voice (Wakin, 2008). Lepage's Ring Cycle for the Metropolitan Opera similarly incorporates dynamic projections affected by the performer's voices and movement and the orchestra's music. This Ring features a giant robotic set that serves as a continually varying surface for projection (Wakin, 2010). While the level of interactivity in these contexts is quite limited, Lepage's goal is to create technology that can be flexible and responsive to constant variation in performance:

> "Now machines allow us to make use of a bit of luck or spontaneous improvisation, so for instance movement, silence or the singer's vocal density, which is never quite the same from one performance to the next, directly influence the images being projected. Humans drive the electronic play." (Machart, 2010)

Other opera productions that have centered on the use of technologies include Tod Machover's *Valis*, which uses two digitally-extended instruments to create the musical score and performance, 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, incorporates 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 (Hewett, 2008). StarChild (Oliverio & Pair, 1998) 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.

Another modern opera that draws on highly sophisticated technology as an integral part of the performance is Tod Machover's *Death and the Powers* (*Death and the Powers*, n.d.; Jessop, Torpey, & Bloomberg, 2011; Torpey, 2012), discussed further in Chapter 3. In this opera, the main character of Simon Powers seeks to extend his life and influence in the world by uploading himself into his house. The actor leaves the stage to be replaced by the theatrical set: bookshelves that communicate through a language of light, color, and movement; a chandelier that is also a musical instrument; a chorus of robots that serve as characters and scenic elements; and surround sound throughout the performance space. All of these elements are expressively shaped in real time by the live performance of the actor playing Simon Powers.

For the remainder of this dissertation, I will primarily constrain the discussion of technological performance systems to interactive systems that extend the behavior of a human performer in real time. The majority of current uses of technology in performance still remain non-interactive, whether that technology comes in the form of a projected backdrop, a computer-generated audio track, or the pre-programmed movement of a scenic element. While these technologies may be cued by a stage manager or technician, their form remains the same from performance to performance, static and unchanging regardless of the variation in the live performance that shares the space with them. For the future development of technology in performance, we want systems that, at their core, extend the live expression of a human, that react to the live nuances of a performance.

2.2.2. Instrumental Model: Hyperinstruments and Musical Systems

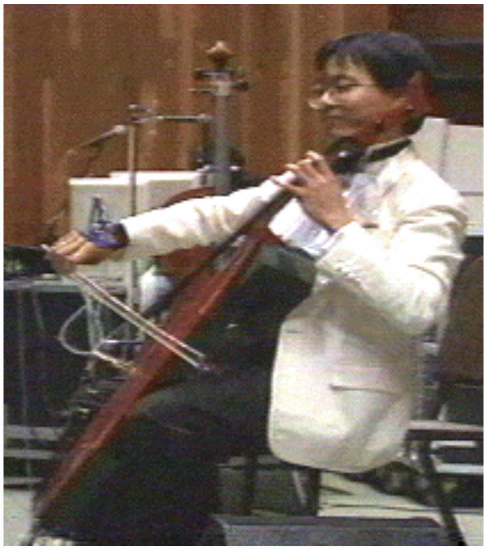


Figure 1. Yo-Yo Ma playing the Hypercello
Photo via Tod Machover.

There is a long history of performance extension through technology in the field of digital musical instruments. Frequently, these instruments have used the performer's movement as a primary control mechanism. Tod Machover's paradigm of Hyperinstruments provides virtuoso musicians with additional levels of expressivity and control through digitally-enhanced traditional musical instruments (Machover, 1992). This model seeks to combine the performance appeal of a live musical instrument with the flexibility and extended sonic range of digital instruments. Importantly, a Hyperinstrument seeks to capture and extend a musician's existing expressive technique, rather than inventing an entirely new vocabulary of movement. For example, Yo-Yo Ma can play the Hypercello expertly as he would a standard cello, while controlling additional processing and layers of sound through his variations in bowing technique, articulation, and other expressive performance elements (Machover, 1992). A Hyperinstrument follows an *instrumental model*, where the behavior of the system is learnable, predictable, and repeatable. Thus, these kinds of instruments allow a professional to take active control of a broader palette of musical manipulation through variations on their existing instrumental technique.

Many other researchers have developed digital musical instruments that extend the behavior of traditional musical instruments and the existing expression and technique of trained musicians. For example, Overholt et al. have designed a computer vision system to recognize gestures of a flute player and use them to cue a computer-generated instrument (Overholt et al., 2009). Thibodeau and Wanderley present an overview of a dozen augmented trumpets, and how their analyses have informed the design of their own “standardized” augmented trumpet (Thibodeau & Wanderley, 2013). Young's Hyperbow is designed to pick up the most subtle details of violin playing (Young, 2002).

In contrast to digital musical instruments that enhance existing vocabularies of performer-instrument interaction, other digital musical instruments have novel interaction models. Particularly relevant to the research presented in this dissertation are those instruments that incorporate free gesture as the mechanism of control. The Theremin is an early free-gesture analog instrument where capacitive sensing allows the user to manipulate the pitch of a generated tone by the movement of one hand in relationship to an antenna, and the amplitude by the movement of the other hand near a second antenna. In this early analog system, the movements of the performer have a fixed mapping to the generation of sound. The “Radio Drum” (Mathews, 1991) is an early example of a computer music system controlled by the free gestures of a performer holding drumsticks whose positions are tracked in three-dimensional space. The “Biomuse” system measures EMG data from moving limbs and can be used to control musical parameters (B. Knapp, 1992). In these and later digital systems, movement parameters can be mapped to a variety of sonic outputs.

Other gestural instruments include Waisvisz's "The Hands" (Waisvisz, 1985), Bokowiec's "Bodycoder" system (M. A. Bokowiec & Bokowiec, 2005), and Sonami's "Lady's Glove" (Bongers, 2000). All of these are wearable instruments that have been used to trigger and manipulate audio in live performance through movements of the performer's arms and hands, though "The Hands" also includes interaction with buttons on the device. Interfaces such as The Hands, the Bodycoder system, and the Lady's Glove have been designed such that the input sensors on each device are separable from the output sound control processes. The sensors can be mapped to different kinds of control for different performance pieces. For example, the Bodycoder system consists of resistive bend sensors on knee and elbow joints and keypad switches in gloves. These sensors can be mapped in a variety of ways to trigger particular sound processing patches, video events, and sound samples, as well as to continuously manipulate sounds in different ways. These mappings can be changed from performance piece to performance piece, and between different sections of a performance (M. A. Bokowiec & Bokowiec, 2005).

There also have been interfaces developed to capture the expressive movement of a conductor either through free gesture or through a handheld device, such as the "Conductor's Jacket" (Nakra, 2000) and the "Radio Baton" (Mathews, 1991). The Conductor's Jacket measures the gesture of a conductor through a variety of sensing strategies including EMG sensors to detect muscle tension, as well as physiological data such as heart rate and galvanic skin response. Data from conductors performing in the jacket has been examined to determine which elements of movement are expressive and communicative to performers. Interestingly, the sensors that measured physiological data in the Conductor's Jacket were found to be connected less to the conductor's expressive intentions and more to the conductor's own emotional reactions. The biggest spikes in a conductor's galvanic skin response did not indicate that the conductor was in the middle of an emotional passage of music, but instead were correlated with the conductor reacting to his own mistakes, mistakes in the orchestra, or events in the audience disrupting the music.

An important extension of the research on capturing expressive performance and extending traditional instruments through technology is the use of similar techniques for amateur contexts. Digital technologies allow the separation of the performer's action from the sound-generating mechanism in a way not possible with traditional musical instruments. Thus, simple interfaces can allow amateurs meaningful control over complex sonic worlds. The form of these amateur performances can be quite varied, from novel percussion instruments (e.g. Weinberg, 2008; Wilkinson, 1997); unconstrained movement, as in the Brain Opera's Gesture Wall (Wilkinson, 1997); tangible interactions with physical objects, such as the Shapers in Toy Symphony (Machover, 2004); or vocal explorations, such as the Brain Opera's Singing Tree (Oliver, 1997).

This sampling of prior work shows a wide range of interfaces and interaction models used for extending musical performance through sensing the behavior of a live performer. It is relevant to note that these digital musical instruments have a separation between the performer's input gesture and at least part of the output sonic result. They are not designed to have only one mapping, one way that they can sound given a particular input; instead, they are often used in multiple performance contexts to create different results. The sound of a Hypercello is not limited to a particular set of sonic samples and manipulations as a real cello is. Instead, the same sensing strategies and analysis can be connected to a variety of different sonic outputs (such as manipulations of the real cello sound and additions to the real cello sound) for different performance pieces and even for different moments within a piece. For an interface like the Bodycoder system that does not incorporate any analog sound generation, the relationships between action and resulting sound are completely determined within a computer.

Since the relationship between sound and action in a digital musical instrument is not constrained by the physics of the input device, any kind of movement can be used to trigger or shape any kind of sound. The same tiny movement of a finger could cause a small tinkling noise or a giant explosion. But what makes sense? What kinds of relationships seem to be meaningful or clearly intentional? With the dissociation between sound production methods and produced sounds granted by digital musical instruments, how can we design digital instruments to be as expressive as analog ones? In designing these instruments, there is a need to create meaningful "action-response associations" (Nakra, 2000).

2.2.3. Technologically Extended Vocal Performance

In the research world of new interfaces for musical expression, there is also a long tradition of performers manipulating and augmenting their voices through a range of technologies, including keyboards and mixers, handheld devices such as Waisvisz's "The Hands" (Waisvisz, 1985), and wearable devices. The majority of technologically enhanced performances that center on the voice do not use the voice itself as a control mechanism, but instead as material that can be controlled by other interfaces such as gesture or tangible interfaces. Gestural interfaces for vocal manipulation have perhaps been particularly popular, including Sonami's solo vocal work with the Lady's Glove, where she shapes her voice via her gesture (Bongers, 2000); the Bokowiecs' use of the Bodycoder system in vocal performances such as "Etch" and "V'Oct (Ritual)" (M. A. Bokowiec, 2011; M. A. Bokowiec & Bokowiec, 2005); Pamela Z.'s and Laurie Anderson's performances with the BodySynth in works such as "Voci" (Lewis, 2007; McBride, 1997); and Imogen Heap's recent performances with the Mi.Mu Glove (Mitchell, 2011; *The Gloves Project*, n.d.). Other interfaces for augmenting a live vocal performance through movement include the "One-Person Choir" interface (Maes, Leman, Kochman, Lesaffre, & Demey, 2011), Knapp and Cook's Integral Music Controller (R. B. Knapp & Cook, 2005), and my own Vocal Augmentation and Manipulation Prosthesis (Jessop, 2009).

It is important to note that each of these gestural models of vocal augmentation presents a very different relationship between movement and the sonic extensions of the voice. Some recognize specific gestures to trigger changes of vocal effects, to bring in synthesized vocal accompaniment, or to record live vocal material and play it back on command. Other interfaces use continuous movement to control live processing and effects on the voice. Some interfaces have a fixed set of behaviors, while others are flexible controllers with a range of effects determined for a particular composition. Clearly, there is no single, obvious mapping that relates movement and vocal processing. However, all of these interfaces intend to support an integrated physical and vocal performance experience, where body and voice act in concert to produce the desired sonic effects.

Blonk and Levin's *Ursonography* presents an interesting variation, in which projected text is synchronized with the performer's tempo and shaped by the performer's vocal dynamics ("Ursonography - Interactive Art by Golan Levin and Collaborators," n.d.). Levin and Lieberman have also incorporated graphics modified by the voice into public installations in *Messa di Voce*, *Hidden Worlds*, and *RE:MARK* (Levin & Lieberman, 2004). In these experiences, the amplitude and spectral content of visitors' voices are used to affect projected graphics. Another public installation where content is controlled solely by the voice is Oliver's "Singing Tree" (Oliver, 1997), with which visitors interact by singing into a microphone. The "pitch, noisiness, brightness, volume, and formant frequencies" of their voices are measured and used in real time to shape the behavior of music and video generation systems.

Other systems for extending performance with the solo voice as input do not necessarily focus on the qualities of the vocal signal, but rather on the text that is spoken. Sparacino's Improvisational TheaterSpace provides an environment for an actor to improvise a scene. The system recognizes simple gestures and certain words and phrases, and determines what text to project in response. Additionally, the projected text can have multiple expressive variations of typography and word movement, which are determined by the system in interaction with the performer (Sparacino, 1996).

Opera is another field where interactive technologies have been used to extend the performer's voice into both audio or visual media. Lepage's *Damnation of Faust* and Ring Cycle, mentioned at the beginning of this chapter, incorporate information from the performer's voices to shape dynamically generated projections (Wakin, 2008; 2010). Machover's *Death and the Powers*, discussed further in Chapter 3, extends the expression of a performer's voice into patterns of light and color on the set, as well as into robotic movement and vocal processing, including the movement of sound in the performance space (Jessop et al., 2011).

2.2.4. Technologically Extended Dance Performance

Extended performance, where the movement of the performers is captured and affects other elements of the performance in real time, has been especially popular in the field of dance. As early as 1965, Merce Cunningham's *Variations V* incorporated photoelectric sensors and antennae to mark the positions of dancers; the data gathered then controlled electronic musical devices (Mazo, 1977). Cunningham also worked with performance capture technologies in a variety of ways, some interactive and some static. He incorporated technology into aspects of the compositional process, the structure, and the content of his pieces. Between 1991 and his death in 2009, Cunningham choreographed the majority of his dances with the support of DanceForms, a program allowing a choreographer to record sequences of movement (frequently using live dancers and advanced motion capture systems), and then manipulate those sequences virtually in a three-dimensional environment ("Credo Interactive Inc.," n.d.). DanceForms provides tools for quickly assembling and reordering sequences of movement, which supported Cunningham's aleatoric choreography practices. Motion capture technologies were also used for creating projected dancers in Cunningham's *Biped* (1999), with visuals designed by Paul Kaiser and Shelley Eshkar. For this piece, Kaiser and Eshkar worked with movement fragments of Cunningham's that had been previously recorded via motion capture technology, created a choreography of those fragments, and transformed the motion capture data into animated three-dimensional figures that performed on scrims as a counterpart to Cunningham's live dancers (Dixon, 2007).

The work of the dance company Troika Ranch is dedicated to the integration of technology and dance. In various performance pieces, Troika Ranch has used dancers' movements to shape visual and sonic elements. The actions of the dancers are detected by laser beams crossing the stage, impact sensors on the floor, or computer vision systems that track points on a dancer's body (Stoppiello & Coniglio, 2003). In these works, the interactions between dancer and media are frequently related to a dancer's location on stage or directly to the values of a sensor on a particular limb. The performer is not given an instrumental level of control over the media. Mark Coniglio and Dawn Stoppiello, the creative directors of Troika Ranch, have developed the mapping software Isadora for controlling live video mixing and video effects in performance. Isadora is capable of taking in movement input via bend sensors on the performers' bodies and external sensing systems ("TROIKATRONIX | live performance tools," n.d.).

A system similar to Isadora that has been used frequently in interactive performance environments is EyesWeb (Ricci et al., 2000). This modular system allows a user to plug in tools for capturing dance movement, analyzing that movement through a variety of techniques, and using that movement to control some output. EyesWeb will be discussed in more depth in Section 2.6.2.

Yamaha's Miburi system (Vickery, 2002), Aylward and Paradiso's Sensemble (Aylward & Paradiso, 2006), and the Danish Institute of Electronic Music's Digital Dance Interface (Siegel & Jacobsen, 1998), have also been used for the real-time generation of music to accompany dancers onstage. All of these systems constrain the possible mappings between movement and digital media through the systems' description of movement, 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).

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 (Winkler, 2002). The Very Nervous System has also been used by Rokeby in a variety of installation scenarios, where different areas of the camera screen are mapped to different instrumental controls. A user's activity and movement in those areas shapes aspects of a music-generating program ("David Rokeby: Very Nervous System," n.d.). Stichting Eleckro-Instrumentale Muzeik (STEIM) has developed another camera-based performer tracking system called BigEye ("BigEye | STEIM," n.d.), often used for performances where performers trigger sound or music events by moving into particular areas of the stage (Siegel & Jacobsen, 1998). 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 (Dixon, 2007).

Sparacino's DanceSpace (Sparacino, Wren, Davenport, & Pentland, 1999) is a space for extended performance and interactive installation that allows both novices and experts to generate musical and graphical output through their movement. Using computer vision techniques to track a performer, DanceSpace creates a representation of the performer's body and associates different instrumental sounds with different parts of the body. By moving different parts of her body, the performer can trigger some particular sounds and shape the pitch of others. This system does not require any special sensors to be worn, and is designed to focus on tracking the movement of specific body parts rather than metrics of "overall movement."

Sparacino's work in dance and theater also includes the concept of Media Actors, software agents to recognize gestures and text from live performers and respond via expressive text, video, and/or sound. In this model, the agents' behavior is driven by a combination of information gathered from external sensing technologies such as microphones and camera systems, and by the agent's own internal motivations and programmed behaviors. There are no direct mappings created between a live performer's movements and the resulting media, since the media manipulations are controlled by an individually-acting software agent.

Marc Downie's work at the MIT Media Lab also incorporates artificially intelligent software agents that are not only influenced by live performance data but also shaped by their own motivations and patterns of perception (Downie, 2005). Downie's interactive agent paradigm sidesteps the question of mapping input data to output media results, taking a different approach to the relationship between live performance and digital media. Biologically-inspired, artificial intelligences can algorithmically generate visual and sonic elements of a performance: these systems perceive live performance information, but have autonomous goals and behavior. These goals can be shaped by an artist creating a piece with these agents, but the specific mappings between the agents' perceptions and their reactions are determined and learned by the agents. For example, in the collaboration between choreographer Trisha Brown and the OpenEndedGroup (Downie, Kaiser, and Eshkar), *How long does the subject linger on the edge of the volume*, interactive agents generate visualizations that are linked to the bodies of particular dancers and projected on top of each dancer. The dancers wear retro-reflective markers picked up by infrared cameras, and the movement information of each dancer is segmented. An agent has the goal of moving across the stage, traveling along on the body of the dancer. While this overall goal for the agent has been determined by the artists, the agent itself is responsible for figuring out when to attach and detach itself to and from different dancers to move across the stage. As the agent learns the choreography, it also learns how to predict more accurately which of its actions will help carry it toward its goal (Soerensen & Lyng, 2005).

2.2.5. Categorizations of Technological Systems for Performance and Interactive Installations

One dimension along which we can examine technological or algorithmic systems in performance or installation contexts is in terms of their relationship to a live performer. Robert Rowe outlines classification categories for interactive musical systems in (Rowe, 2004), which can be extended beyond 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 used to control an output that goes beyond the traditional response of an instrument but still feels like it stems from a human's live performance. 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 system observes Lewis's live performance on the trombone, but itself decides how and when to use that information in determining what it is going to play for its part of a duet (Rowe, 1993).

I categorize the relationship between system and performer into four different models:

- Static systems
- Stage management systems
- Agent systems
- Instrumental systems

The behavior of a *static system* is not influenced in any way by the input of a live performer. This includes systems where the output is fixed and unchanging, such as tape music or a pre-edited video projection. This category also includes systems where the output is stochastic or probabilistic, and systems where the output is influenced by external, non-performer-related data (about the weather, the stock market, etc.). In all of these cases, while the actual output of the system may vary every performance, there is no relationship whatsoever between the live performance and the behavior of the system.

In *stage management systems*, the digital material is often static and is not influenced by any performer onstage, but aspects of the material may be controlled by technical staff. In this model, cues may be triggered at particular times by a live technician, such as a stage manager calling “go” on a lighting cue, a sound effect, or a preprogrammed rigging cue. Material may also be manipulated live by other technicians: for example, a light board operator changes lighting palettes as he is inspired by a live rock show and a sound engineer mixes a production based on the levels he hears. These cues and shaping may be completely dependent on the behavior of performers. The stage manager calls a lighting cue when a performer says a particular line, or crosses to Stage Left. The sound engineer adjusts the level of a particular singer’s microphone from moment to moment based on the singer’s volume. However, in these systems, the technology is not seen as an extension of the performance. Those who might be said to be “performing” with the technology (the theatrical technicians) are typically offstage and are not generally seen by the audience as a component of the performance. The precise timing and shaping controlled by technicians can play a large part in the audience’s overall experience of a production. However, these tasks are more likely to have limited expressivity and limited creative input.

In *agent systems*, input from or analysis of a live performer is used to feed and inform a system’s own generative processes, though that input is not the only thing used by the system in determining its output. These systems are perhaps better seen as agents that interact with a live performance, but have their own goals and behaviors. Downie’s interactive agents for *How Long Does the Subject Linger on the Edge of the Volume* are an example of such a system in the domain of dance performance (Downie, 2005). The autonomous behavior of agents such as Downie’s and Sparacino’s (Sparacino et al., 1999) avoids naïve mappings between the live performance and the media elements, but the resulting interactions are generally not reproducible and do not give the performer a sense of control.

A final category of interactive system is the *instrumental system*, where the behavior of the system is under the direct control of a live performer. An instrumental system has as input some elements of a performer's behavior and uses those elements to shape its behavior in ways that are sufficiently learnable, repeatable, and perfectible by the performer. This is the category of system that is most examined and explored in this dissertation.

These categories are, of course, not completely separable; instead, there is a continuum of all of these modes of interaction between performer and system. For example, an instrumental system where a performer's behavior shapes generated music might have a layer of stochastic behavior, such that a performer might be able to control the general timbre of the sounds produced by the system, but not be able to intentionally trigger specific individual sounds. A given performance piece might also have many different kinds of systems overlaid, or different system models at moment to moment.

Another dimension along which we can examine systems for technological performance extension is the expected expertise of the performer. This dimension is inspired by Wanderley's division of gestural musical interfaces, which incorporates three different types of gestural systems (Wanderley, 2001):

- Digital musical instruments (experts play, generally tactile interfaces, intended for specific sonic results, the audience listens to the performers)
- Interactive music installations (generally free-movement, extreme novices use, exploration rather than sound may be the main goal, the performers are also the audience)
- Dance-music systems (the expressive goal is not only the music output but the choreographic movement used in producing that output)

Experiences such as the Brain Opera (Paradiso, 1999) fall on one side of the spectrum; visitors to the Brain Opera had no prior experience with the novel instruments they encountered, and were given no explicit instructions about how to play those instruments. The instruments had to be designed so as to create a compelling interaction without practice or known technique. On the other extreme are solo performance systems from more traditional extended instruments such as the Hypercello (Machover, 1992) to novel interfaces such as the Lady's Glove (Bongers, 2000) or the Hands (Waisvisz, 1985). In these cases, the performer is expected to be a virtuoso, either in the traditional instrument that serves as a basis for the interaction or through long practice with the new digital musical instrument. Intermediate conditions include the Toy Symphony instruments (Machover, 2004), where amateur performers study and rehearse with novel instruments. One interesting design goal for interactive systems is that they have a "low floor and high ceiling"; it is quick to do something that is reasonably interesting, but additional practice and repetition will allow for finer and finer control of new layers of detail. The framework and systems described in this dissertation are designed to create interfaces for both the novice and expert sides of the scale, as the concept of expressive qualities is equally relevant for both. This work includes examples of both public installations for novices and interactive performance extensions designed for experienced performers.

2.3. Existing Frameworks for Analyzing and Describing Movement

2.3.1. Delsarte's System of Oratory

In the late 19th century, the former actor Françoise Delsarte attempted to create a comprehensive theoretical framework of movement in performance. This was one of the earliest (and, in fact, one of the only) efforts at creating such a framework. Delsarte sought to develop a theory of oratory and aesthetics based on the inflection of the voice, the movements of the body, and the content of speech. His theory explores how these elements conveyed aspects of the speaker's "life," "soul," and "mind," which he saw as three separate but connected entities (Delaumosne, 1893). In Delsarte's framework, gesture serves to convey a person's "soul," that is, their sentiment and emotion, and was the most powerful of these elements of oration. Delsarte states, "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" (Delaumosne, 1893).

Delsarte proposes ideal forms for conveying desired sentiments to an audience, using a particular gesture and stance. Each emotion is linked to a particular set of positions and movements of the eyes, arms, hands, and 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 described to represent different states of the speaker's mind, from vehemence to terror. Additionally, Delsarte believed that gestures should be limited, controlled, and focused on individually.

Delsarte also notes a crucial point in the definition of gesture, that gesture is not merely represented by static poses and postures. For Delsarte, the "dynamic" of gesture is contained in the inflictions and rhythms of a movement. This concept of dynamic movement is an important characteristic of Delsarte's analysis of gesture: he describes that movement is communicative not only through the performer's stance and pose, but also through the way that the shape of the body changes over time.

2.3.2. Dance Notation and Description Systems

Contemporary dance performance analysis and notation systems are a valuable resource for exploring methods to define movement and expression in movement. Contemporary dance has a tremendous range of choreographic styles and movement vocabularies, ranging from formal ballet technique to pedestrian movements like walking, running, and jumping. As the majority of the various forms of dance focus on expression conveyed by the body, we can ask what qualities of movement have been identified in prior models used for the study of dance performance and how those have been related to the expressivity of the movement. Some prior models describe movement with either parametric or semantic spaces, forming higher-level descriptions of movement than basic physical details. Such descriptions of movement qualities and their identifying features can serve as an interesting context for creating frameworks for recognition of expression.

While there exist many frameworks for notating dance movement, the majority of these frameworks do not incorporate notions of movement quality. Such frameworks primarily describe which body parts are involved in a movement, the direction of movement, and the amount of time taken for the movement (potentially with some notion of where the accent falls in the movement). While this is sufficient information to represent some form of the movement or choreography, it does not provide a conceptual framework for understanding expressive qualities, or the dimensions along which a given choreography can vary. Ann Hutchinson-Guest lays out the aspects of movement that are typically analyzed by dance notation, including the timing of actions, the parts of the body used, the spatial variation, and the quality of the movement. She refers to 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.)" (Guest, 1984 p. xiv).

Different dance notation systems have been shaped by the complexity and key aspects of the forms of dance that they represent. Early dance notation systems, beginning in the 1500's, were designed to represent social dances. These forms of notation defined sequences of step patterns, where a given letter was used to represent a particular (known) sequence of steps (Guest, 1984, pp. 42-46). As social dance routines became more elaborate and more focused on movement patterns around the floor, dance notation systems evolved to feature description of spatial movement. In the 1700's, Feuillet's notation system presented bird's-eye views of a dance, notating a dancer's relationship between the music, the steps, the arm movements, and the position in the floor pattern at any point in time. As theatrical dance became more prominent and developed a larger gestural vocabulary, notation systems such as Feuillet's became insufficient to capture the variety of movement that had to be represented (Guest, 1984, pp. 62-67). However, new movement description systems still focused on linking specific movement patterns to specific points in a musical score, and found limitations when the vocabulary to be described did not come from a specified set of movements (such as ballet).

Rudolf Laban's framework of dance notation is likely the most popular system in use today. Labanotation consists of a complex set of symbols that are combined in sequences (read bottom to top) to represent sequences of movement. These can include information about travel patterns, directionality of the body, actions of each part of the body, interaction between dancers, positions of dancers in relation to one another and to a space, relative timing and spatial relationships between movements, relationship of movement to a specific meter, and specific movements such as jumps, turns, contractions, and extensions of different parts of the body (Brown & Parker, 1984). Labanotation can also combine into a single notation symbol several different features of a particular movement, including its direction (represented by the shape of the symbol), its timing (the symbol's length), its level (the symbol's shading), and the body part used in the movement (the symbol's location on a staff) (Guest, 1984, p. 84).

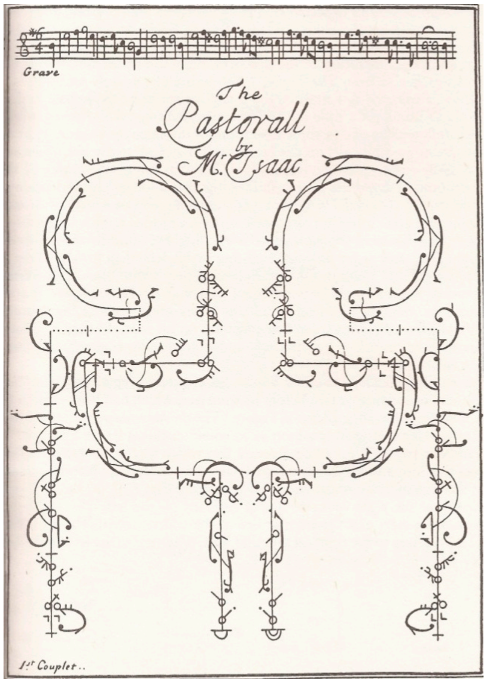


Figure 2. A dance represented in Feuillet's notation system
Patterns through space are a major component of early dance notation systems. Image from (Guest, 1984).

Different variations of Labanotation can define movement sequences with varying degrees of specificity and precision. For example, a notated sequence could describe a dancer traveling forward briefly to center stage, then moving on a slow, curving, and indirect path to stage left. There are clearly many ways to perform this sequence of actions that would correctly follow the directions given by the notation. Laban also developed a system to define the kind of muscular activity and movement quality used in performing a movement. This system, Laban Effort Notation, will be discussed in the following section.

Benesh Notation plots movement left to right along a five-line stave, following the format of music notation. Patterns of travel, directionality information, information about the location of a dancer in a performance space, and relationships of groups of dancers are notated beneath the staff. The positions and movements of key points on the body and limbs, changes of level, and changes of weight are marked with simple lines on the staff, with different positions and movements delineated with a specific symbolic vocabulary. Information about rhythm and phrasing, such as the location of musical beats, the tempo, and the continuity or separation of movements is marked above the staff (Brown & Parker, 1984). This notation system is concerned with creating a flexible linguistic structure for dance notation. 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 (Guest, 1984, 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.

While the modern dance critic John Martin did not develop a particular system of notation, he carefully analyzed modern dance, describing a variety of features that span both quantitative and qualitative aspects of movement. Martin divides the features of modern dance into four categories: *space*, *time*, *dynamism*, and *metakineses* (Martin, 1933). In Martin’s definition, space includes the features of the volumetric space formed by the dancer’s body, by parts of the body, of bodies in relation to a performance space, and of bodies in relationship to other bodies. Martin states that a dance can also be described by temporal elements: its patterns in time, speed, duration of parts of movement, rate of succession of movements, and “regularity or irregularity of intervals between stresses” (Martin, 1933, p. 55). Stress and dynamism are defined as the levels and variations of intensity in the movement. Finally, metakineses describes the “overtones” of movement that convey intention and emotional experiences.

2.3.3. Laban’s Theory of Effort

Rudolf Laban, best known for his work in dance and movement notation was attempting to define and categorize movement qualities as early as the 1920’s. His original model includes four qualitative states, all of which are interconnected: *force*, *time*, *space*, and *flight*. In Laban’s view, movement starts in stillness and ends in stillness, and during the movement each of these four parameters can be increasing or decreasing. Movements that stay the same from beginning to end are seen as mechanical. Laban considers dance not only as a sequence of fixed states, but as the process of transformation between states (McCaw, 2011).

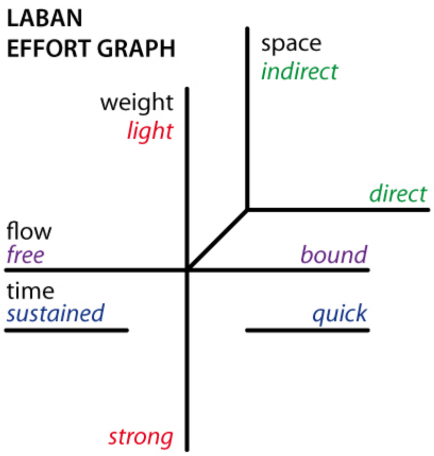


Figure 3. Laban's effort space
The four dimensions of effort identified by Laban. Image by Raphaël Cottin, reprinted from (Cottin, n.d.)

In 1947, Laban wrote *Effort*, presenting a Theory of Effort for analyzing movement dynamics including strength, control, and timing. This framework was primarily constructed for the analysis of the types of movement used in industrial contexts in order to increase efficiency, but Laban believed it would be applicable to many domains (Hodgson, 2001). Laban describes the quality of a movement using a continuous "effort space" with the four axes of *weight*, *time*, *space*, and *flow*. Weight (measured on a scale from strong to light) describes the amount of energy and intensity in the movement. Time (sudden to sustained) describes the speed of a movement. Space (direct to flexible) describes the way that a movement travels through space around the body. Flow (bound to free) describes a movement's smoothness of energy and tension. It reflects the degree to which a person is struggling against a movement or giving into a movement. Movements with free flow cannot be stopped suddenly or easily interrupted, while movements with bound flow are easy to stop at any point.

Laban uses combinations of weight, time, and space to characterize eight basic different categories of movement or *Effort Actions*, each of which can be performed with either free or bound flow. These include punching (direct, strong, quick), pressing (direct, strong, sustained), slashing (flexible, strong, quick), wringing (flexible, strong, sustained), dabbing (direct, light, quick), gliding (direct, light, sustained), flicking (flexible, light, quick), and floating (flexible, light, sustained) (McCaw, 2011). Among the four effort dimensions, it is also possible for different actions at the same point on the spectrum to have different dimensions as a focus. For example, "crushing fruit" and "cutting leather" are both seen as direct, strong, and sustained, but the former places an emphasis on weight as the primary dimension of interest, the latter on direction (McCaw, 2011).

It is important to notice that these effort dimensions, as well as those of other frameworks like Martin's, are subjective and not defined by any particular quantitative values. What is the "quickest" movement or the "most sustained" movement? The descriptions of a dimension may have an intuitive sensibility, but actually identifying a given movement as having particular effort values will require consideration of the context. For example, the range of *time* values with which one particular action can be performed may be different than the range of *time* for another particular action.

Laban's framework has been adapted or used as inspiration by many researchers examining affective and qualitative movement information, including Fagerberg et al., Volpe, and Zhao and Badler (Fagerberg, Ståhl, & Höök, 2003; Volpe, 2003; Zhao, 2001). In looking at Laban Effort Notation, Volpe identifies that expressive content is likely to be contained in the trajectories of a movement through this effort space over time, rather than of the particular values at any point in time.

2.3.4. HCI Approaches

For computer processes to identify information about movement and voice from a stream of data over time, we can apply techniques from machine learning. *Pattern recognition* is a machine learning technique where a computer is trained to discover a function that maps between input examples and output labels, with the goal of generalizing from known examples to appropriately handle new inputs. If the desired output of the system is discrete categories, such as the identification of a specific gesture, this process is *classification*. If the desired output is real values, this recognition process is *regression*. Regardless of the algorithms used, the process is similar: input sensor data undergoes *feature extraction* to obtain a set of features that may be particularly descriptive of the input; selected examples of this processed data are used to train a model that represents a best guess at the relationships between the input feature vectors and the specified output values; finally, that model is presented with new inputs and tested. Depending on how much is known about the desired relationship between inputs to and outputs of the pattern recognition algorithm, the training process may be *supervised*, which means that input data provided for the training process is labeled with the desired corresponding output values. Alternately, the process may be *unsupervised* and given unlabeled data when the goal is figuring out how to categorize or extract features from the data when the desired output labels are not known beforehand. An example of an unsupervised learning algorithm is the K-means clustering algorithm, which has as its goal figuring out how to group a set of data points into K clusters. This is unsupervised learning, as we do not know which cluster a given data point should belong to at the beginning. In *semi-supervised* learning, the algorithm can be provided with not only a set of labeled data, but also a set of unlabeled data that can be used to improve the algorithm's classification assumptions.

In the field of Human-Computer Interaction, a significant amount of research has gone into pattern recognition techniques for movement capture, particularly in the field of gesture recognition. Gesture recognition in HCI has been performed using a variety of input technologies, including computer vision systems (Ko, Demirdjian, & Darrell, 2003; Starner, Weaver, & Pentland, 1997), handheld devices (Bahl, Jelinek, & Mercer, 1983; Schlömer, Poppinga, Henze, & Boll, 2008; Strachan, Murray-Smith, & O'Modhrain, 2007), wearable systems (Benbasat, 2000), and EMG sensors (Zhang et al., 2009). Additionally, this research has used and expanded a variety of pattern recognition and machine learning algorithms, such as Hidden Markov Models (Eickeler, Kosmala, & Rigoll, 1998; Starner et al., 1997; Zhang et al., 2009), Principal Component Analysis (Billon, Nédélec, & Tisseau, 2008), Dynamic Bayesian Networks (Avilés-Arriaga & Sucar, 2002), and Neural Networks (Modler, Myatt, & Saup, 2003). Many gestures and poses with applications for HCI, as well as a number of gesture recognition technologies, are summarized in Saffer (2008). However, there are limitations in the adaptation of HCI technologies for performance contexts. Typical gesture recognition systems work best for applications where there is a predetermined gestural vocabulary and all movements made by the user are expected to fall into that set 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 other types of movement.

Another limitation with most standard gesture recognition tasks in expressive contexts is that they are best used in situations that expect a one-to-one mapping between input and output. A specific gesture is recognized and used to trigger a specific result. These discrete trigger gestures need to throw away the majority of information about gestural variability in order to perform well at recognition. If a system has to recognize when you raise your right hand with a flat palm, and perform some action based on the recognition of that action, it needs to compress many variations of the same basic gesture (the hand raised at different speeds, with different acceleration curves, with a direct or slightly curved path, with different amounts of rotation, with different amounts of tension in the hand and arm, etc.) into a binary value: is this the desired gesture, or is it not the desired gesture? This single bit of information removes the majority of expressive information in the movement.

One example of a system that illustrates this limitation is the g-speak system developed by Oblong Industries ("g-speak - oblong industries, inc.," n.d.). In this system, individual gestures are recognized and mapped in software directly to keystrokes and mouse movement and clicks. This standard computer interaction model can then be used in writing g-speak software programs. While the system allows spatial interaction with large projected displays, the real "interaction" with the system can be no more complex or interesting than that which could be performed with a keyboard and mouse. This significantly limits g-speak's use in expressive contexts. For example, in a prototype g-speak audio application developed by Adam Boulanger, the user could "pluck" strings with g-speak's mouse click and release gestures, and quiet the system by raising both hands, palms to the screen. The use of discrete gestures for plucking discrete strings appeared effective. However, the result of the gesture to quiet the system was not a decrescendo, but a sudden drop in volume in the middle of the gesture being performed at the moment when the system recognized the gesture. The system's inability to capture continuous information about movement was not an acceptable relationship between gesture and result in this context.

Other HCI researchers have focused on recognition of particular movement parameters or qualities. Fagerberg et al. propose a three-dimensional space (the *affective gestural plane model*) consisting of *shape*, *effort*, and *valence* for analysis of emotional movement. This model has been used to help users express emotion in text messages. These authors' principles for the design of interfaces include embodiment (focusing on the connection between body and emotion); "natural but designed expressions" (a specifically-designed set of interactive gestures, but inspired by natural behavior so as to be easy to learn); the affective loop (having users perform expressions of emotion and have their emotions shaped by those movements and by the results of that movement in the application); and ambiguity, allowing for personal interpretation of emotions (instead of having labeled emotion buttons, for example) (Fagerberg et al., 2003). This model blends aspects of Laban's Effort Theory with Russell's Circumplex Model of Affect (Russell, 1980). Russell's studies have shown consistency in people's mental maps about how emotions are distributed in a two-axis space along axes of *arousal* (energy/intensity) and valence (positivity vs. negativity). An important thing to note about Russell's model is that it positions emotions in a continuous space, rather than as a set of discrete categories.

Manfred Clynes's research on "sentic forms," cross-cultural "spatio-temporal curves" of emotions, explores temporal features of movement (Clynes, 1977). The shape of these essentic forms have been determined by many users performing a simple button-press action. The button has two axes of displacement, vertical and horizontal displacement. Subjects push the button while focusing on a particular emotional state, such as joy or anger. Clynes argues that the resulting curves of the button's displacement over the length of the push action, when normalized in time, form similar shapes across many users. An important thing to take from Clynes's research is the use of dynamic and time-dependent shapes, rather than static positions, to describe an emotional space. Taking a single snapshot of the button's displacement would provide very different content than examining the button's movement through time. Clynes' work also suggests some concept of parametric universality: some ability to communicate expression from person to person. One person can do something expressive and someone else can understand something about the expression and emotion being communicated, though the labels for that information may vary.

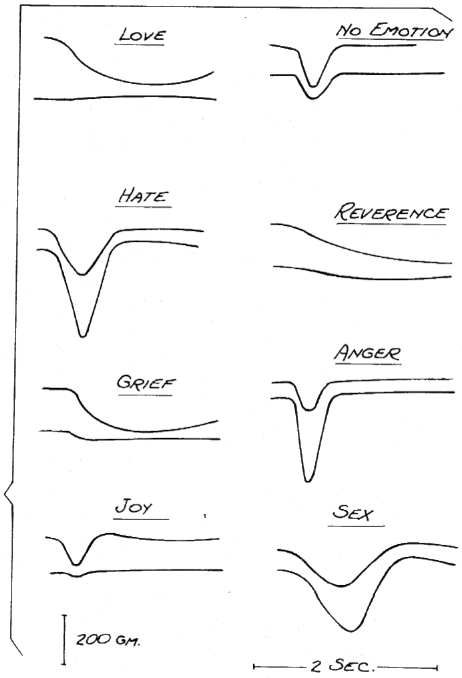


Figure 4. Clynes's sentic forms
The curves in each pair represent the X and Y displacement of Clynes's Sentograph button over time. From (Clynes, 1977).

2.3.5. Specific Needs of Expression Recognition for Digital Music Interfaces

Traditional gesture recognition systems still lack many aspects necessary for expressive analysis tasks, as they generally rely on a known set of sensors and a pre-determined vocabulary of recognized gestures (often trained with samples from many users). As discussed by Gillian, systems for gesture recognition in performance contexts need to handle input and output ambiguity, as well as user-specific vocabularies: users may incorporate a variety of different sensors to detect performance input, choose their own vocabularies of recognition for particular pieces, and have the results of the pattern recognition algorithms control many different kinds of output (Gillian, 2011). To perform pattern regression in the domain of expression recognition, both classification and regression algorithms are necessary. In addition, each of these types of algorithms may need to be paired with labeled, unlabeled, or weakly labeled data.

There has been some prior work in creating systems for flexibly mapping gestures to specific sonic results. Merrill's FlexiGesture is a multi-degree-of-freedom gestural input device that allows a user to associate particular inertial gestures with specific sounds for playback, as well as to map specific degrees of freedom of the device to desired continuous control parameters (Merrill & Paradiso, 2005). While the FlexiGesture's technology is limited to a specific input device, it seeks to address the problem of mapping in digital musical interfaces, where any input data stream can be connected to any output control value. This device provides a methodology for instrument-creators to quickly experiment with different mappings and demonstrate examples of interesting gestures.

Gillian's gesture recognition extension package for the EyesWeb mapping system allows a musician to quickly train a system to recognize a desired vocabulary of physical gestures given a limited number of examples. Gillian states that in the design of a digital musical instrument for a specific performance, it is not as important to have a gesture recognition system be good at generalization across many users as for a system to be adept at learning one user's behavior from a small set of training data (Gillian, 2011). Gillian's system also supports the user in exploring a variety of machine learning algorithms, allowing a user with limited experience in machine learning to experiment with the recognition results of different algorithms.

Fiebrink's Wekinator system allows users to work with supervised learning algorithms in real-time contexts, training mappings from input features to output control parameters on the fly (Fiebrink, 2011). This allows users to rapidly iterate on their mappings by quickly collecting training data examples and modifying their machine learning model by adding new data and retraining. This system also features the concept of "play-along learning": given a score that plays back desired output parameters over time, a user can gesture along with the score to generate labeled training data. However, while the Wekinator can handle both classification and regression tasks, this system has no concept of parameters that vary over time. The Wekinator training examples label a specific set of input features at a particular moment with a set of output parameter values. In order to represent any concept of temporal parameters, the system needs to include computed features that already contain some concept of time, such as derivatives or averages. Thus, it is perhaps incorrect to categorize the Wekinator as a "gesture recognition system": more generally accurate would be the label of "pose recognition system", where the input to the machine learning algorithms at any given moment may not have any relationship to sensor information at prior moments.

A major design decision of the Wekinator system is its choice to have the user directly teach the system mappings between inputs and outputs, with no intermediate steps revealed or accessible. A strength of this model is the ability to discover unexpected intermediate mappings between trained points. For example, a user can label one position of a joystick with a specific set of control parameter values for a sound generation engine, label another position of the joystick with a different set of values for the same parameters, then train the Wekinator on these positions. After the system has been trained, moving the joystick between the two positions will result in a variety of different intermediate values for the sound generation parameters. This is presented as an interesting feature of the system, giving instrument-designers the ability to discover unexpected interactions between their actions and the sonic result.

However, Wekinator's process of directly learning a mapping from input to output values limits the user's ability to later switch sensing systems or output systems, or to add control parameters for multiple output media, as it does not maintain any high-level or intermediate representation of the mapping. A position (of a joystick, of the user in front of a camera, of another sensor) gets mapped to a discrete trigger, or to a position in a set of continuous values. But what if the user no longer wants to use a joystick, but instead a sensor mounted on his arm? Or to have a system whose interaction "feels" similar, but whose output is a different sound generation engine? The process of retraining the system is quick, but the user has to completely start from scratch. With meaningful intermediate mapping stages, inputs and outputs could be more easily switched and added. Additionally, the Wekinator mapping process does not particularly support the user to think about making interesting or meaningful mappings, or *why* a particular position should or should not be correlated with a certain set of parameters.

Most of these systems presented for movement analysis via machine learning only recognize whether or not a specific gesture has been performed, while I also want to recognize time-varying expressive qualities of movement. No system yet exists that allows a user to train a system to recognize specific continuous qualities, rather than classifying movement into emotional categories or into particular labeled gestures.

2.4. Existing Frameworks for Analyzing and Describing the Voice

Both movement and voice are innately personal and expressive instruments, with many similarities in temporal and physical constraints. However, qualitative analysis of the human voice has generally been considered separately from qualitative analysis of movement. Even researchers studying the interaction of gesture and speech frequently focus on the correlation between particular gestures or gestural features and words or vocal features (e.g. Sargin et al., 2006), rather than exploring features that could describe both movement and voice equally. As with the analysis of physical movement, which includes concepts of both gesture and quality, analysis of the voice has also addressed two (generally separated) problems: how to identify the content of the voice (speech recognition) and how to describe the qualities of the voice.

2.4.1. Speech Recognition

Like the standard Human-Computer Interaction model of movement analysis, the typical vocal focus of the field of HCI is on speech recognition. Given a user speaking a particular set of words, can the system classify these into different categories and match them with trained classes of words? This is typically treated as an interpersonal classification problem: a system needs to be able to recognize the same words spoken by many different users. Thus, the focus for speech recognition tasks typically is not on analyzing how different examples of a word differ from some "standard" example, but on how to compress many variations into one category. As with gesture recognition, speech recognition must be able to handle input signals of varying lengths. An input word should be classified correctly regardless of whether the speaker takes a longer or shorter amount of time to pronounce it.

Frequently, machine learning techniques are used to address speech recognition tasks. Hidden Markov Models have been a popular stochastic model for speech recognition (e.g. Bahl et al., 1983; Baker, 1975; Rabiner, 1989). Other techniques for performing speech recognition include dynamic programming (e.g. Itakura, 1990; Sakoe, 1979) and neural networks (e.g. Waibel, Hanazawa, Hinton, Shikano, & Lang, 1989; Wu & Chan, 1993). Many techniques are collected in (Waibel & Lee, 1990) and (Jelinek, 1997). The basic fundamentals of speech recognition via a computational process are described in (Rabiner, 1993).

Rabiner reviews the use of Hidden Markov Models for speech recognition and generalizes the process for performing speech recognition tasks computationally (Rabiner, 1989). In the simpler form of the problem, "isolated word recognition," the challenge is to recognize which individual word in a database a given audio sample is most likely to represent. The audio signal is broken into many small overlapping windows, each of which is analyzed to compute spectral features. These windows of features can then be used as input to a Hidden Markov Model. Generally, one model is trained per word that the user desires to recognize. In cases where the desired vocabulary is long, systems may be trained on individual syllables or sounds that can then be connected to form a variety of words. In “connected word recognition,” individual words have to be picked out from a longer audio stream consisting of many words. This requires additional levels of processing to find the best way to segment an audio stream into words, and can potentially include semantic or syntactic context to predict a most likely sequence of words. In speech recognition, the segmentation problem is less of an issue than in gesture recognition, as breaks between words (silences) can often be identified easily from the vocal signal, assisting the system in knowing which section of audio represents a segmented word.

Traditional speech recognition techniques may be a useful layer in some performance pieces (if the word spoken is X, then extend the expressive performance this way; if the word is Y, then extend the expressive performance a different way). However, for the context of this dissertation, techniques for systems to identify particular words are less relevant than techniques for identifying concepts of vocal quality or expressive variation. The next sections summarize various prior work on analyzing and describing vocal qualities.

2.4.2. Qualities of Speech

A variety of researchers have attempted to come up with frameworks for describing vocal quality, though primarily for discussing vocal dysfunction rather than for exploring expression. Of the research in this area that focuses on expression, most comes from the field of speech analysis, but many of the principles can be generalized to a broader range of expressive vocalization. Scherer identifies a variety of vocal features that he has found to convey expression, including vocal perturbation (short-term variation), voice quality (timbre), intensity, tempo, and the range of the fundamental frequency over time (Scherer, 1986). “Gestures” of the spoken voice have been specifically examined in research such as (Maestre, Bonada, & Mayor, 2006), where they have been represented as trajectories of fundamental frequency and “energy” over time.

Aspects of “voice quality” have also been defined separately from parameters of vocal prosodic contour in analyses such as that of Grichkovtsova et al. (Grichkovtsova, Morel, & Lacheret, 2012). In their studies, *prosodic contour* refers to features such as intensity, fundamental frequency, speech rate, and rhythm of the voice. Their vocal quality metrics, inspired by those defined by Laver (1980), include “phonatory, articulatory, and tension components” of the voice (Grichkovtsova et al., 2012). Campbell and Mokhtari primarily focus on “voice-quality” measured from “pressed” to “breathy,” arguing that this quality should be a major prosodic parameter used for vocal description, along with pitch, power, and duration (Campbell & Mokhtari, 2003).

Vocal quality of spoken words has also been substantially examined by researchers exploring how the voice changes with the speaker’s emotional state (Banse & Scherer, 1996; Fernandez, 2004; Frick, 1985; Ladd, 1985). Vocal aspects found to be relevant in the identification of affect in speech include parameters relating to fundamental frequency range, pitch contour, intonation, loudness, and rhythm. A summary of features and classifiers that have been explored for emotion recognition from speech can be found in (Anagnostopoulos, Iliou, & Giannoukos, 2012). Most HCI studies in these areas focus on classifying emotions from vocal parameters.

The goal of many studies on emotional parameters of the voice is to help create more expressive and realistic synthesized voices through modification of those parameters. Cahn's research addresses affective parameters of speech, which she distinguishes from prosody elements (intonation and rhythmic patterns). She presents modifications to the fundamental frequency and timing parameters as major features in conveying affect (Cahn, 1990). Cahn also presents a model with four different types of features of the voice: pitch (features related to the fundamental frequency including melodic shape), timing (rhythm, word stresses, silences, speech rate), voice quality (describing the voice as a whole, includes "breathiness, brilliance, loudness, pause discontinuity, pitch discontinuity, and tremor"), and articulation (precision of enunciation) (Cahn, 1990).

We see that in addition to features with more standardized definitions such as "loudness," "pitch," and "rhythm," many researchers add a definition of "vocal quality," which can represent a variety of other perceptual parameters. These elements of vocal quality can be seen as relating to a speaker's emotion, as well as to the individual character of a speaker's voice. Specific vocal quality features such as "breathiness" or "brightness" or "timbre" have a variety of different mathematical definitions that seek to capture a perceptual parameter as an equation. There is little consensus between researchers about how to define "vocal quality," either in what parameters should be part of the concept or in how to define particular parameters. However, we can see that vocal quality and expressive aspects of the voice often relate to spectral characteristics of the signal and temporal variation. Additionally, several studies discuss the role of different lengths of time necessary for the analysis of vocal expression, for example, the pitch of the voice at a particular moment versus the contour of the voice's pitch within a word or within a sentence. It is clear that systems for looking at vocal expression require analysis of the signal at different timescales.

2.4.3. Singing Voice Analysis

Other researchers have carried out analyses specifically on the singing voice, often with the goal of developing better algorithms for singing voice synthesis. Kim separates features reflecting an individual's vocal physiology (such as the configuration of the vocal folds and vocal tract) from features reflecting an individual's expressive performance (such as how those features vary over time) (Kim, 2003). D'Alessandro et. al. control synthesis of the singing voice from a database of samples through a sketch-based interface that incorporates dimensions of voice quality including "tenseness," "vocal effort," and "registers" (d'Alessandro et al., 2008). Maestre et al.'s analysis of musical articulation gestures in the voice (transitions from note to note) shows that different articulation patterns can be described and modeled as frequency and energy contours evolving over time (Maestre et al., 2006). This analysis separates the aspects of the vocal contour determined by the musical composition from those added by the singer for expressive purposes and those inherent in the production of particular consonant patterns.

Other researchers have analyzed the singing voice to use that analysis as a rich input to other synthesis algorithms. Stowell analyzes various spectral features of the voice related to timbre, with the goal of mapping those features directly onto similar features in output sound synthesis. In that research, the primary higher-level analysis parameter is stated to be an axis between "brightness" and "dullness" (Stowell, 2010). Mestres's analysis of qualities of vocal gestures includes intermediate parameters such as brightness, articulation (from legato to staccato), dynamics, and pitch (Mestres, 2008). In the work of Ramakrishnan et al., vocal signals are analyzed and turned into a three-dimensional high-level feature space for low-bandwidth communication over a network. The actual dimensions in this feature space are not meaningfully labeled in any way, but calculated using Principal Component Analysis to capture the majority of the variation in the signal. This data, along with low-level features of input vocal gestures, has been used to control parameters of a generative instrument (Ramakrishnan, Freeman, & Varnik, 2004).

2.5. Similarities of Analysis Between Movement and Voice

Movement and voice are both innately physical processes, necessarily continuous in nature, and confined by the mechanics and physics of the human body. Gestures in each modality occur over similar timescales and range from having well-defined semantic meanings (in the case of speech or iconic gesture) to being purely abstract. The definition of gesture as an action that conveys information is equally applicable to both body and voice. Additionally, movement and vocalization have often been used concurrently for communication. I propose that similar frameworks and methods of thinking can be applicable for analyzing and extending expressive performance information from both modalities. Certainly, there are many overlapping aspects of existing frameworks that have been used to describe movement and those frameworks used to describe the voice. As with movement qualities, measurements of intensity and energy have been found to be particularly important in describing vocal qualities. In addition, both modalities highly rely on the shaping of parameters over time (such as dynamics, rate, and rhythm) for communicating expression.

While many different parametric frameworks have been imagined to describe movement or the voice, it is also important to note that no single analytical framework seems to completely capture all aspects of vocal or physical expression. I argue that no particular set of expressive parameters will be universally valuable and sufficient for all performance and installation contexts. Fortunately, with advances in machine learning algorithms, it is possible to surpass one-size-fits-all models in favor of parametric models that can be specific to a particular artist, to a particular piece, to a particular section of a work. These models can be shaped and developed throughout the lifecycle of a performance or installation, during initial experimentation, during the rehearsal process, and even during the run of performances. Additionally, it is now possible to design generalized frameworks that can assist in the development and training of these very specific models. Existing parametric models of voice and movement (such as Laban's theory of effort) can provide context for what types of parameters might be interesting to examine, but are not universally sufficient. While my research includes a set of suggested parameters for description of both movement and voice, these axes serve primarily as a starting point for the analysis of expressive physical performance.

2.6. Mapping Systems and Strategies: Connecting Inputs and Outputs

The majority of the systems described in any interactive performance context come down to, at the core, a question of mapping. How is the physical performance input (from sensors, microphones, video cameras, etc.) related (mapped) to the control parameters of output media? Is this mapping consciously created by a performance-maker, or learned or discovered by a system? How much variation is there in the mapping during the course of a performance?

2.6.1. Mapping Strategies

As discussed earlier, the standard mathematical definition of mapping is an operation or function that associates each element of a given input set to one or more elements of another output set. Mathematical concepts of mapping suggest a variety of different relationships between input and output. In a *one-to-one* mapping, each different input value produces a distinct output: no two input values produce the same output value. An example of this in the performance domain would be mapping the value of a resistive bend sensor to the volume of an output sound. In a *one-to-many* mapping, each input value is connected to many output values. For example, the amplitude of a performer's voice could control several different parameters of a video. In a *many-to-one* mapping, a variety of values of the input will lead to the same output value. In a *many-to-many* mapping, each input value may be related to multiple output values, and multiple different input values may relate to the same output value.

When thinking about creating a mapping for performance, it may be beneficial to introduce multiple levels of abstraction between individual input streams of data and individual parameters of multimedia control. The importance of abstraction in mapping systems has been previously addressed in the context of electronic musical instrument design (e.g. Hunt, Wanderley, & Paradis, 2002; Wanderley, Schnell, & Rovan, 1998). Unlike traditional acoustic musical instruments, where the gestures and actions used to play the instrument are generally explicitly sound-generative, electronic musical instruments allow for the decoupling of performer-controlled inputs from sound generation control parameters. Simple one-to-one mappings between individual inputs and individual sound generation parameters are found to be easy to uncover, but hard to use expressively or to become expert at playing.

Hunt et al. have found that different mapping strategies can have a major effect on the interest and enjoyment of a performer. One-to-one mappings, say from a particular slider to a particular parameter of sound, can be quickly figured out. However, given a multiple-slider interface, users found it challenging to think about performance in terms of individual parameters controlled individually: "Comments abounded such as 'I should be able to do this, technically, but I can't get my mind to split down the sound into these 4 finger controls.' Some users actually got quite angry with the interface and with themselves." A different interface version that incorporated many-to-many mappings was found to be more intuitive, though requiring longer to learn the basics: "At first it seemed counter-intuitive to most users, but they rapidly warmed to the fact that they could use complex gestural motions to control several simultaneous parameters without having to 'de-code' them into individual streams"(Hunt et al., 2002).

A more complex mapping strategy proposed by Hunt et al. involves three stages of mapping: input parameters are mapped to more complex abstract parameters describing the input (such as energy or variation in movement), abstract input parameters are mapped to abstract parameters describing the output (such as pitch and brightness), and abstract output parameters are mapped to specific parameters of sound generation. This frequently results in complex many-to-many mappings that may be harder to master and understand immediately, but provide easier control and expressivity once grasped.

A related mapping strategy has been proposed by Torpey in his work on multimodal scoring systems and mapping systems (Torpey, 2009; 2013). In this model, all input data is mapped into a position in a multidimensional expressive parameter space defined in the system. This position in the abstract space can then be mapped to control parameters for output media. In (Torpey, 2009), this parameter space consists of the emotional parameters of *arousal*, *valence*, and *stance*. In (Torpey, 2013), this parameter space consists of the abstract parameters of *weight*, *intensity*, *rate*, *density*, *complexity*, *texture*, and *regularity*. The goal in this framework is to create meaningful transformations of input to output by going through an abstract intermediate model.

Another important aspect of mapping creation is the distinction made by Teresa Marrin Nakra between continuous and discrete inputs and output responses (Nakra, 2000). In this model, Nakra identifies discrete gestures as "single impulses or static symbols that represent one quantity," including simple actions such as pushing a button or a key on a keyboard, as well as more sophisticated actions such as the recognition of a hand pose. Regardless of whether these symbols are as simple as a keypress or as complex as a recognized gesture, they generate only one bit of information: is this action occurring, or is it not occurring? Thus, these discrete gestures fit a one-to-one mapping relationship, with each gesture mapped to a discrete modification of sound. I agree with Nakra that complex mappings, with inputs that cannot be compressed to a single symbol, are more likely to hold meaningful information in an expressive context. This concept of discrete vs. continuous gesture will be discussed further in Chapter 4.

Volpe, examining the process of creating mappings that relate to a performer's emotional state, presents several layers of simultaneous mapping possible, including high-level mappings from an emotional state and low-level parameter-to-parameter mappings. In Volpe's terminology, an "expressive direct mapping" does not use any high-level information about a performer's emotional state, but instead directly connects input to output parameters. An example of this model is mapping a dancer's position on stage to a synthesis parameter. An "expressive high-level indirect mapping" incorporates higher-levels of analysis in the process of the system's determining how to map inputs to outputs. For example, a system could select a set of functions for mapping based on some rational process. "Expressive mapping monitoring" is the act of the system looking over the other mapping processes and evaluating if they are producing the desired results (Volpe, 2003).

2.6.2. Existing Mapping Systems

Many software tools currently exist to facilitate mapping tasks in performance contexts, including Max/MSP ("Max « Cycling 74," n.d.), EyesWeb (Ricci et al., 2000), Isadora ("TROIKATRONIX | live performance tools," n.d.), and Field (Downie, 2005). Efforts have been made by artists to incorporate machine learning and concepts of expression into some of these tools. For example, EyesWeb recognizes some pre-programmed qualitative features and the SARC EyesWeb catalog incorporates gesture recognition (Gillian, Knapp, & O'Modhrain, 2011). However, there is still ample room to develop mapping tools that simplify the process of working with higher-level expressive information. As I believe that creating meaningful mappings becomes easier when the inputs to that mapping become more intuitive to the composer or choreographer, it is vital to have mapping systems that thoughtfully define expressively meaningful qualities of movement and voice. Additionally, pattern recognition techniques can play an important role in helping systems learn how to transform sensor data into expressively meaningful inputs to the mapping process.

I will discuss the mapping system EyesWeb in more depth, as it has some goals that are similar to those of the work in this dissertation. EyesWeb (Ricci et al., 2000; Volpe, 2003) is a modular, freely available software program intended for live movement capture and analysis, with a focus on encouraging high-level views of gesture. It has a variety of tools for processing input video streams and other sensor information. While EyesWeb can accept various sensor inputs, it is primarily designed for calculating movement parameters from camera input. EyesWeb also has a set of machine learning add-ons for gesture recognition, the SARC EyesWeb Catalog developed by Nicholas Gillian (Gillian, 2011).

Within EyesWeb, there are also modules that incorporate concepts of physical "expression" such as the amount of space taken up by the body, the dancer's physical stability, and the movement's rhythm in space. EyesWeb calculates twelve such "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" (Camurri et al., 2008).

Camurri et al. have also explored the concept of KANSEI (emotional) information in dance (Camurri, Hashimoto, Suzuki, & Trocca, 1999). Some of EyesWeb's movement analysis is inspired by Laban's theories of movement, including Laban's concept of effort.

While EyesWeb does attempt to incorporate some concept of physical expression and "movement quality," this system has some major limitations in the extent to which it is able to recognize expressive details about movement. The majority of its expressive parameters (such as the "quantity of movement" and "contraction index") have been pre-calculated according to its creators' theories about movement, which does not allow for particularly flexible exploration of these parameters. The definition of these parameters is fixed in the system, as is the specific set of parameters. While users with a background in programming can create their own modules for EyesWeb, the system does not support easy definition of new expressive parameters, or include any machine learning techniques available to the user for defining and constructing expressive parameters.

Additionally, the EyesWeb system primarily has been used for classifying movement fragments that are intended to convey a specific emotion, so its frameworks of expressive information have been designed based on what parameters were found useful in that context. The primary study exploring affective and expressive movement information examined whether the system could acceptably recognize four basic emotions (fear, grief, joy, anger) expressed in short excerpts of dance performances (Volpe, 2003).

2.7. What Can We Learn? What Is Still Missing?

In this chapter, we have presented a wide variety of prior work in the areas of technologically extended performance and installation design. We have examined how movement and the voice have been typically analyzed in theatrical and musical contexts, as well as in Human-Computer Interaction contexts. Through these analyses, we have found some aspects where existing systems are insufficient. We have also found that there is a need for more flexible performance extension through high-level definitions of physical and vocal expression.

We have explored a variety of models for representing physical and vocal expression, both computationally and theoretically. Some of the key lessons from our analysis are summarized here:

- There is a distinction between simple categorical classification tasks (gesture recognition, speech recognition) and evaluating continuous expressive parameters (qualities of movement, qualities of voice).
- Expression and emotion are frequently represented by the changing shape of parameters over time.
- Physical and vocal expression is conveyed via features that take place over many different timescales.
- Although movement and vocalization are both innately shaped by the physical properties and timescales of the body, these performance media have not been considered together in frameworks for an expressive performance context.
- For both body and voice, metrics of energy, rate, and scale are relevant descriptors.

Current systems for performance extension do not allow users to flexibly define their own meaningful parameter spaces for vocal and physical description. Given the wide variety of parameter spaces that have been found to be interesting by different researchers and performance-makers, ideal performance extension tools should allow users to explore many of these kinds of parameter spaces. There is not likely to be one new set of parametric axes that will be the most appropriate for analyzing the movement and vocal qualities of every piece in a way that is meaningful to the performance-maker. Even if a system uses one particular fixed set of abstract parameters, it ought to be flexible enough to allow the user to define those parameters in variable ways, with different sensing strategies and different ranges.

This dissertation research includes the development of tools for defining and working with such flexible parametric spaces. The systems discussed here, such as the Expressive Performance Extension System, are designed to allow performance-makers to incorporate into their existing practices higher-level abstraction of movement and the voice, as well as machine learning tools for analysis of expressive quality spaces. The next chapter discusses some of my initial research projects in extended vocal and physical performance and explores the ways in which these projects inspired many principles and features of the proposed workflow for performance extension and of the Expressive Performance Extension System.

3. Early Studies and Development of Principles

A variety of my previous research projects at the Media Lab have provided useful testbeds for the design of theoretical and computational frameworks for expressive movement and voice capture and analysis. These projects include: the Vocal Augmentation and Manipulation Prosthesis, a gesture-based wearable controller for live vocal extension in performance; the Gestural Media Framework, a system for abstraction of gesture recognition and Laban-inspired movement analysis in the context of a live dance performance; and the Disembodied Performance System for the opera *Death and the Powers*, a sensing and mapping system for extending the voice and physicality of an opera singer into the behavior of an entire theatrical environment. This chapter outlines and analyzes these early projects, and explores how they have inspired some of my principles for expressive performance extension.

3.1. VAMP: The Importance of Evocative Gestural Mappings

3.1.1. An Overview of VAMP

The Vocal Augmentation Prosthesis (VAMP) is a wearable controller in the form of a long glove that allows a singer to manipulate her voice in live performance, so she can simultaneously serve as performer and conductor. Wearing VAMP, a performer can capture and harmonize with notes that she sings, extending her voice purely through free gesture (Jessop, 2009). The distinctive interactive characteristic of VAMP is the singer's use of a pinching motion to sample and capture a moment of her voice or other audio material. As she sings, she can "grab" a note from her mouth and hold it beyond her own physical abilities. The captured note is then extended for as long as she keeps her fingers pressed together, allowing her to sing other material and thus harmonize with herself. Other gestures can affect the processing of the captured note.

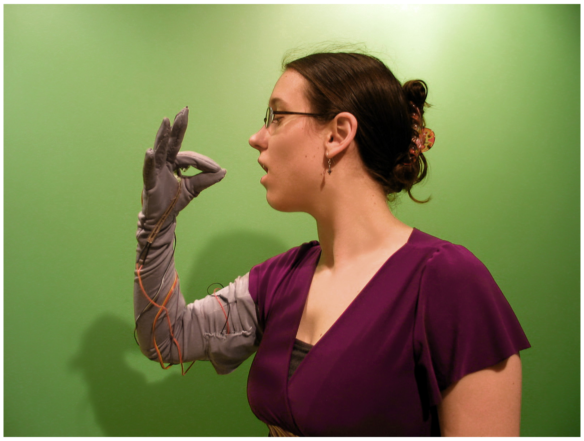


Figure 5. The Vocal Augmentation and Manipulation Prosthesis (VAMP)
The author pinches her fingers by her mouth to "grab" the note she sings.

This controller was originally inspired by the character of Nicholas in Tod Machover's opera *Death and the Powers*. While VAMP was not used in the final production of *Death and the Powers*, its genesis within that context inspired many of the design choices used in its development. In the opera, Nicholas, the research assistant, is scripted to have one prosthetic arm that gives him abilities beyond those of a normal arm. We decided to explore the ways in which an augmented arm might be able to give the character additional musical abilities. In particular, given the operatic context of the production, we sought to find a way to use a special arm to extend the performer's voice, to give him abilities that he could not achieve without the technology.

As the performer playing the character of Nicholas was an opera singer and not an instrumentalist, it was important that whatever procedure we developed to augment his voice through the use of his arm needed to be clear and simple. The performer would be singing a fairly complex score, and would not have much spare mental processing power available to simultaneously play a complex instrument, or one whose use was not particularly intuitive. Additionally, it was necessary for the movement vocabulary to be quite visible and direct for an audience that would likely be more familiar with opera than with interactive music performance, and that would only be observing the gestures from a distance. How could it be obvious to such an audience that the performer's gestures actually were affecting his vocal performance? Small movements such as pushing a button or fiddling with a knob would not necessarily read clearly to the audience as intentional for vocal manipulation. Instead, we needed a form of interaction that could have an obvious connection between the performer's actions and the resulting sonic behavior.

Given these guidelines, I designed a vocabulary of gestures based on choral conducting, as well as a core gestural metaphor of grabbing and extending a note by pinching two fingers together near the mouth. I also designed the mappings to associate these gestures with particular sound manipulations: pinching together the thumb and forefinger to capture a note, extending the arm away from the body to crescendo, raising the hand to add a harmony note, shaking the hand to add vibrato and overtones, and beating time with the arm to "pulse" the captured note rhythmically. Given gestural associations from standard conducting, certain relationships between a performer's action and the sonic reaction may seem familiar to an audience and a performer. We are used to seeing a conductor gesture more broadly and openly to encourage a louder sound, or more closely to her body to indicate a softer response. Similarly, we are used to the gesture of a conductor raising her hand to bring in another section (another note), and of her keeping a beat through rhythmic gestures. This desired movement vocabulary was developed before the technical implementation of the instrument, and strongly informed the choice of sensing and sonic manipulation techniques.

3.1.2. Technology and Implementation

The Vocal Augmentation and Manipulation Prosthesis takes the form of a long glove that stretches past the performer's elbow. The VAMP glove is custom-constructed from stretch velvet fabric, with an assortment of sensors sewn onto the glove to measure various aspects of the performer's gestural behavior. Wearable sensors were incorporated rather than off-the-body sensing (such as computer vision techniques) to be most flexible in a variety of performance situations and not require external hardware to be incorporated into the stage setup. The specific sensors used were chosen in order to recognize major movement components of the predetermined gestural vocabulary.

Two 4.5" flex sensors are sewn onto the glove, one located on the outside of the elbow and one on the outside of the wrist. When the sensors are used as variable resistors, voltage measurements correlate to the amount of strain placed on the sensor. 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 (though these directions are not differentiated in the output, only the amount of distortion from center). The sensors are affixed to the glove at several points along their length, to make sure that their flexion corresponds as closely as possible to that of the glove itself and, thus, the wearer's arm. The glove is also 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. An accelerometer on the wrist picks up movements of the hand. 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.

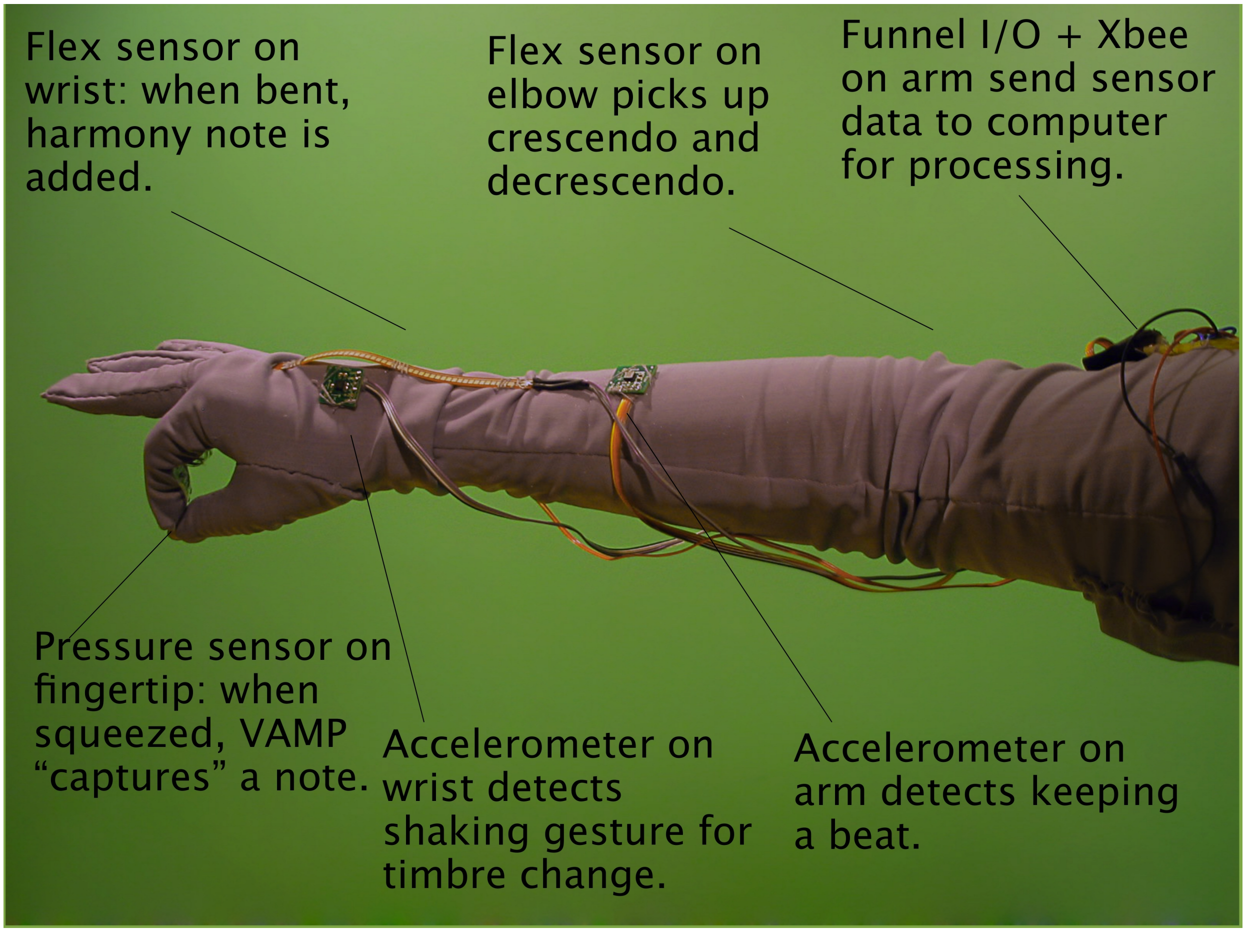

Flex sensor on wrist: when bent, harmony note is added.
Flex sensor on elbow picks up crescendo and decrescendo.
Funnel I/O + Xbee on arm send sensor data to computer for processing.
Pressure sensor on fingertip: when squeezed, VAMP "captures" a note.
Accelerometer on wrist detects shaking gesture for timbre change.
Accelerometer on arm detects keeping a beat.

Figure 6. Sensing system for VAMP

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 modules to a Java applet. This Java program utilizes the Processing API and Processing's Arduino libraries to enable communication with the Funnel I/O board. In the Java program, the sensor information is collected, smoothed, 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 via Open Sound Control ("opensoundcontrol.org," n.d.). 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, while the sensor input and analysis are carried out using Java and Processing.

The implementation of the "frozen" note processing uses the Max pfft~ subpatch `solofreeze.pfft` designed by Jean-François Charles (Charles, 2008). This subpatch uses matrices to perform spectral processing on a Fast Fourier Transform of the audio signal, which allows the necessary computation to be done in real time, and provides a richer sound quality by blending frames together in a stochastic process to avoid an obvious "looping" sound.

3.1.3. Analysis and Observations

While VAMP was not incorporated into the final design of *Death and the Powers*, I have used it in a wide range of demonstrations and a variety of small performances, including a few solo pieces for myself with the glove; an improvisational duet for the glove and an augmented guitar; and a small chamber ensemble piece for soprano, VAMP, cello, tunable kalimba, daxophone, oboe, and piano (composed by Peter Torpey). The connection between the gestural vocabulary and the sonic manipulations have been found to be quite direct and compelling.

One key lesson from VAMP is the power of a strong connection between movement and sonic result in evoking a sense of liveness and magic in an interactive instrument. In the design of VAMP, the interactive vocabulary of the glove was determined before the physical technology was built. This separation of the desired gestural vocabulary and control mappings from the technology needed to implement them helped create a system that could have intuitive and expressive interactions between movement and music, rather than sensor-driven interactions. When the gesture and the sonic result were closely coupled in metaphorically or emotionally resonant mappings, as when the wearer pinches her fingers together to capture a note or when she stretches out her arm to build a crescendo, the resulting interaction proved compelling and interesting.

This early instrument was a major influence for my thought process about performance extension technologies. While different effects were technically linked to values of different sensors, the design process did not create the interaction by envisioning that technological implementation. One could imagine an alternate methodology in which a set of wearable sensors would be first selected and then the values of each of those sensors could be used to control a particular sonic transformation: the amount of bend in the elbow linked to the amount of reverb on the sound, for example, or the amount of bend in the wrist used to control the pitch of a held note. While such a set of manipulations might be clear, it could also easily appear that the connections were chosen at random, without thought for the semiotics of a particular gesture or kind of movement.

It is also important to note that the majority of the recognition process of particular gestures in VAMP has been simplified to detect key elements of a gesture and capture those through the selected sensor set. For example, detection of the pinching gesture to capture a note is performed solely by the pressure sensor on the tip of the forefinger. When this sensor's smoothed values pass above a specified threshold value, a trigger is sent to capture a new note and the note extension is turned on in the Max patch. When the sensor values drop below the threshold, the note extension is turned off. From this example, it is clear that a note could actually be captured by a finger press subtly made with the hand at the side, or from pressing the finger against a surface, or any number of other motions that bring the fingertip pressure sensor over its threshold. Thus, the action of pinching the fingers near the mouth (or near wherever the microphone is held) to capture a note is only partly a technical requirement for the system, and, more importantly, a deliberate performance gesture.

This balance between the aspects of a gesture that are technically necessary for recognition and those that are unnecessary but compelling in performance is a important lesson from the design process of VAMP. One might ask, why wouldn't we want to use more complex gesture recognition and make sure that a performer had to position his hand by his mouth when pinching his fingers in order to "capture" a note? However, an argument against such a system would be the additional complexity required to identify all aspects of the desired behavior. Given that the VAMP system has no absolute position sensing, additional sensing devices would likely need to be added. The programming would also become more complex and risk missing desired gestures through over-specificity. How close to the mouth does the hand have to be to identify the pose correctly? Does the hand have to be at a specific angle? Do the non-active fingers need to be in a specific position? What if another performer with different physical proportions wears the glove? Is there a desired temporal pattern of bringing the hand to the mouth? Capturing a note is a simple trigger, not a continuously changing parameter, so it only requires a simple action to identify a desired trigger time. The performer should not have to think about the system's accuracy in the moment when his action triggers the effect. If this glove were used in a different context, where pressure on the finger could occur in many different conditions but should only trigger capturing a note when the hand is held to the mouth, this would require a more complex sensor system; however, the VAMP implementation is a simpler case. By combining a strong performative behavior (bringing the hand to the mouth to pinch the fingers together) with a simple sensor and analysis system (a thresholded pressure sensor on the forefinger triggering the note capture), we can create the desired effect with minimal error in performance.

Additionally, new performance behavior grew out of the unintentional affordances of the system, such as the ability to "grab" a note from another performer or instrument. In a series of performances, I have explored the effects of capturing other musical content: for example, in a 2009 duet with Rob Morris and his Wii-augmented Gesture Guitar (*Robert R. Morris: Gesture Guitar*, n.d.), I used VAMP to grab notes and chords from Rob's guitar and control them through the same sonic manipulations as originally designed for my voice. We discovered that the pitch-warping algorithm used for bringing in a harmony note that produced a clean fifth when used on pure-tone vocal content created some fascinating timbral variations when used on a complex sonic source such as a guitar chord.

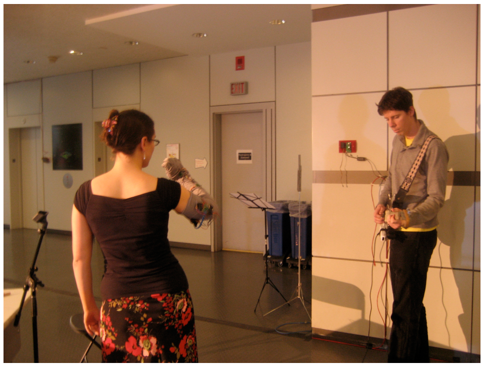


Figure 7. Performance with VAMP and the Gesture Guitar
A duet between the author and Rob Morris. Photo by Peter Torpey.

One primary challenge with the VAMP system was the difficulty of creating new gestural behaviors or of varying the sonic manipulations after the full system had been developed. It was designed as a specific instrument with a specific set of interactions, and was not constructed with the aim of being easy to manipulate or to create new mappings from movement to sound. In designing later systems, I have attempted to balance the needs of developing a strong gestural vocabulary and being able to flexibly create new vocabularies with the same sensor setup.

3.2. The Gestural Media Framework: Abstraction of Gesture for Flexible, High-Level Mappings

3.2.1. An Overview of the Gestural Media Framework

Building on the importance of high-level movement analysis in mapping, in 2009 I began developing systems to abstract raw sensor data into more meaningful movement descriptions. What if I gave you a glove that knew when you flicked your hand, and how hard you were flicking it? That knew when you waved your hand, or when you squeezed your hand, and how tightly you were squeezing? What if this system provided you with information about recognized gestures and descriptive parameters of those gestures, and I asked you to use that information to control the generation of a reactive visual or musical experience? How might you think about designing those mappings? Most interestingly, how might that imaginative design process differ from the more standard one, where I would give you a glove and tell you that it had a bend sensor on one finger and a three-axis accelerometer on the back of the hand, and ask you to use that information to control the visual or musical experience? The way that you might envision mappings given the former set of information would likely result in very different interaction models than those you would envision given the latter set.

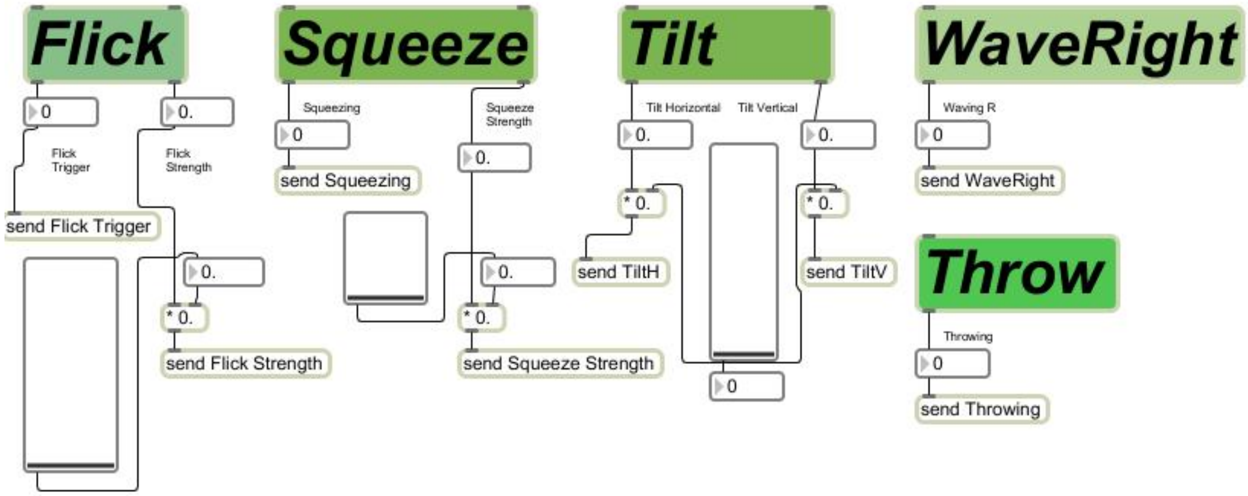


Figure 8. Initial Gestural Media Framework implementation in Max/MSP
Max/MSP objects received identification and modification parameters for detected gestures and allowed a user to scale parameters as desired.

Initial explorations of this concept took the form of a glove augmented with a few sensors and hand-coded recognition algorithms to detect a small set of desired gestures. Data from this glove was encapsulated in Gesture Objects that hid the raw sensor data and produced output only about whether the represented gesture was occurring and of any descriptive parameters related to that gesture. Sensor processing and gesture recognition was performed in Java and sent to Max/MSP objects for mapping to sound manipulation parameters and parameters of visualizations. These

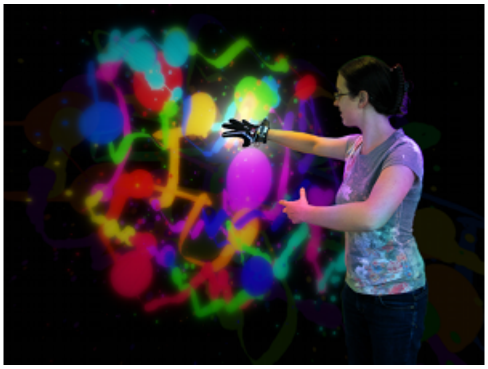


Figure 9. Gestural Media Framework glove
(Rendering by Peter Torpey)

mappings could be easily modified and scaled without the user having to be aware of the original sensor information or how the gestures were recognized. I developed several small interactive visual experiences based on one gestural vocabulary: a "splatter painting" program where paint was rolled around through tilting the hand and splattered with a flick of the hand; a fluid dynamics simulation where forces in the fluid, the color of the fluid, and splashes of particles were controlled by gesture; and a gesture-operated slideshow program.

The core concepts of this work were extended into the Gestural Media Framework, a system built as part of my master's thesis work at the MIT Media Lab to recognize a vocabulary of trained gestures from continuous streams of movement data using Hidden Markov Models. This system also incorporated hand-coded concepts from a modified Laban Effort Space to characterize qualities of a performer's movement (Jessop, 2010).

In the Gestural Media Framework system, a user can specify his or her own important gestural vocabulary, labeled by whatever names are intuitive and clear to that user, rather than being constrained to any sort of pre-determined gestural vocabulary that attempted to be generic or iconic. If I were to train a system on a basic vocabulary of gestures, that vocabulary would immediately be constricting to anyone who tried to use the system, including myself. As this system had the goal of flexibly developing and exploring mappings between gestures and output control, it seemed appropriate that it also should have flexibility in the input gesture set. 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. It was important in this system to integrate temporal pattern recognition techniques rather than simply static pose recognition. A given pose might be part of a wide variety of gestures and movements; it seemed overly limiting to force a choreographer to only pass through a particular pose as part of the gesture intended for recognition.

In order to incorporate temporal pattern recognition techniques, this system uses the Georgia Institute of Technology's Gesture and Activity Recognition Toolkit (Westeyn, Brashear, Atrash, & Starner, 2003), a set of libraries to support the use of Hidden Markov Models in Java. Using GART, one can 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 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 guess is accurate. In the GMF system, if a gesture is identified with a probability of accuracy greater than an empirically determined threshold, the gesture is labeled as currently occurring.

In addition to identifying particular gestures, this system sought to also have some sense of quality of movement: *how* gestures (both those recognized by the system and other gestures) were being performed. While specific gestures are recognized via machine learning techniques in this system, the qualities of movement are calculated and programmed manually. Aspects of particular sensor data streams are correlated to the Laban-inspired axes of *time*, *weight*, and *flow*. Each parametric axis is manually scaled to the range -1.0 to 1.0.

The *time* axis describes the speed at which a particular movement is being performed, from very fast and sudden to very slow and sustained. For implementation of this axis, the "speed" of a motion was a measurement related to how quickly the body is changing its position and orientation. As the sensors used for this project measured acceleration and the bending of joints, speed was correlated with the overall rate of change of the sensor data. The amount of change in each sensor's value over a short time window is summed, with the contribution from each type of sensor weighted empirically to balance the contribution of rapid changes in joint position and in acceleration. This parameter of the amount of change over all sensors is mapped from -1.0 (very rapid movement) to 1.0 (no movement or exceedingly slow movement).

In Laban's system, the *weight* axis describes movement on a scale from firm to gentle. Firm movements are forceful, strong, resisting, and heavy; gentle movements are relaxed, unresisting, light, and weightless (Laban, 1980, p. 73). While the performance of this quality seems intuitive, it was not immediately clear how to derive it from data from the given sensor set. However, given an alternate definition of weight used by Laban, where weight is a measurement of the amount of energy put into the movement, it seemed possible to link weight to the amount of current acceleration. The total acceleration on the performer's body is measured empirically between gentle, still movements and strong, forceful movements. This range is mapped from -1.0 (intense, energetic, heavy movement) to 1.0 (light, low-energy, gentle movement).

The final quality of motion discussed in Laban's Theory of Effort is *flow*, which describes the amount of freedom of energy in a particular movement. Flow is a measurement of how smoothly and continuously the movement is changing, described using an axis from "fluid" movement to "bound" movement. If the movement is changing smoothly and evenly, continuously, and uninterrupted, it is considered to be more fluid; if the movement starts and stops, changing jerkily and unevenly, it is considered to be more bound (Laban, 1980). This parameter was determined to correspond to the amount of change over all sensor values, as examined over longer timescales. A running value for flow was calculated that would be increased or decreased at each interval depending on whether the performer's movement was currently changing rapidly or slowly. At any point in time, the value of flow (from -1.0, bound, to 1.0, fluid) reflected the overall trend in the change of the motion.

It is important to note that these mappings between Laban's qualities of movement and a particular sensor data set were created and tuned empirically, inspired by potential information given by specific sensors and their variance over time. Other qualities of movement, such as Laban's concept of *space*, were not implemented due to the difficulty of associating them to data gathered by the existing sensor set.

3.2.2. *Four Asynchronicities on the Theme of Contact*

As part of the evaluation of the Gestural Media Framework, I choreographed a suite of performance pieces, *Four Asynchronicities on the Theme of Contact*, that used this system in the performance and rehearsal process to map dancers' movements to the manipulation of projected visualizations, sound, and theatrical lighting. This work consisted of four connected movements that explored different ways that people try or fail to connect with one another, fragmenting interactions in time and space. All performers wore sensor-enhanced shirts and gloves, with accelerometers on their hands and arms and flex sensors to detect the position of their wrists, elbows, and forefingers. Sensor data was collected via Funnel I/O microcontrollers and sent wirelessly to the computer running the analysis software.

Each movement incorporated a different kind of technological extension of the performance. In the first movement, a duet, the qualities of the performers' movements affected the intensity of the theatrical lighting and transformations of projected washes of color, while particular key gestures (such as reaching out to one another) shifted the color palettes. In the second movement, a solo performer controlled a soundscape through the qualities of her movement. Different kinds of sounds were played when the performer's analyzed movement qualities fit into different regions of the Laban-inspired three-dimensional quality space (time, weight, and flow). Other sounds were triggered by specific gestures. The third movement was a duet where each performer's movement affected the generation of a different instrumental part, each playing a semi-random walk between notes in a selected scale, with the key and scale switched by recognized gestures. In the final movement, a quintet, the quality of each performer's movement affected a different region of a fluid dynamics simulation projected behind the performers, highlighting rapidly shifting groupings and moment-to-moment dynamic variations among the performers.

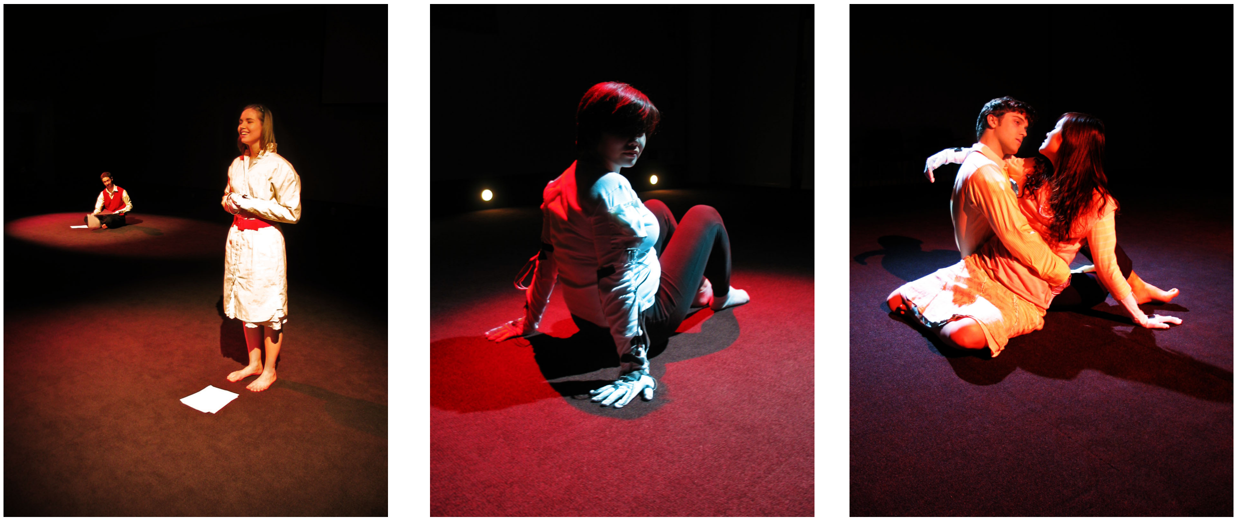


Figure 10. Images from *Four Asynchronicities*, Movements 1-3
L-R: Kevin Burchby and Lisa Smith; Danbee Kim; Noah Jessop and Xiao Xiao. Photos by Peter Torpey.

3.2.3. Analysis and Observations

One of the key principles discovered in the process of developing and working with the Gestural Media Framework was the different role of movement information that was represented in a discrete manner ("Is a particular gesture occurring? Yes or no?") than that of information that was represented continuously ("How fast is the current gesture being performed right now?"). I found that the recognition of particular gestures to discretely trigger events was often not particularly interesting; much more important in conveying expression was the subtlety of how a movement was performed. The recognition of a gesture actually compresses a tremendous amount of detail about a movement into a yes or no question: did the system just recognize that gesture? If a gesture recognition system is working well, many variations on the same movement will be grouped together and the expressive variation in different performances of that movement will be lost. While some interactive situations strongly need these sort of simplified triggers, it is important not to lose too much information about the details of a live performance. The use of continuous vs. discrete analysis of movement and the voice will be discussed further in Chapter 4.

Another relevant lesson from this project was that different qualities needed different scales of time to be properly represented. In this implementation, *weight* and *time* were qualities that could change rapidly within the window of a few sensor readings, reflecting the immediate state of the performer's body, while *flow* needed to change more slowly over the course of a gesture or a sequence of gestures. Thus, systems working with qualities of movement may need the ability to adjust the window of time over which the system analyzes input data to feed into the qualitative analysis.

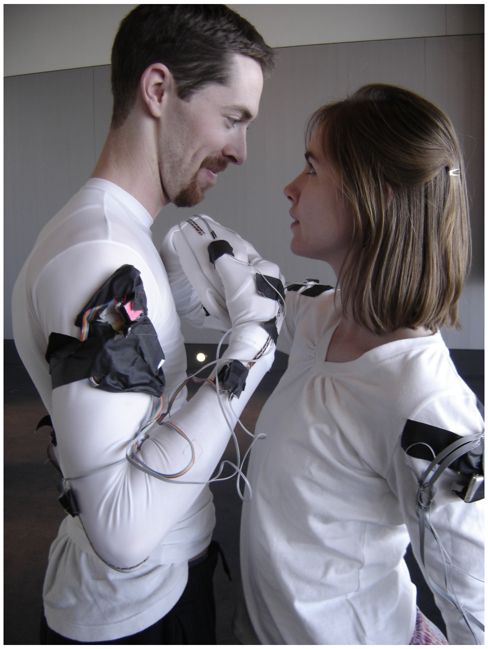


Figure 11. ***Four Asynchronicities* performers** Lisa Smith and Kevin Burchby in wearable sensor arrays, with accelerometers and bend sensors on their arms.

Another important lesson was the challenge of segmenting meaningful gestural data from an unlabeled stream of movement information. When a trained system is performing recognition on live input, there is no immediate way for a system to know what segments of movement it ought to analyze to detect a gesture. There is not necessarily a rest state between key movements, and a significant proportion of the movement input may not be related to the gestures that need to be detected. In *Four Asynchronicities*, I chose to combine two different modes of movement segmentation. In the first mode, the Hidden Markov Models continually processed several different window lengths of historical data. In the second mode, a manual trigger marked the start of a gesture: a performer wore a sensor on his fingertip that he would touch to indicate to the system that he was about to perform one of the key gestures. While the manual trigger segmentation was more accurate, it added choreographic restrictions and another aspect of which the performers needed to be aware.

In this system, the gesture recognition was performed using Hidden Markov Models. The training process used separate programs for training the system and for recognizing data given a trained model. The gesture recognition was therefore trained offline, in separate portions of rehearsal dedicated to capturing training data, and run online in the performance context and while rehearsing choreography. This separation of programs between the training system and the live system, while more modular for development, led to the gesture recognition process being less smoothly integrated into the rehearsal process. In general, capturing new training data examples may need to be a separate rehearsal event from running a particular choreographic fragment, depending on what aspects need to be captured from that training data. However, if two entirely separate computational systems are required to perform each of those tasks, this does not support a process of easily iterating on gesture recognition for the production, either experimenting with learning new gestures or modifying training data for existing gestures.

Given the hand-coded relationships between qualities of movement and a selected sensor set, the system was also not easily adaptable for a variety of movement qualities. Different sensing setups or the decision to use different parameters for quality of movement would require significant experimentation and careful thought to develop and code the relationships between sensors and movement qualities. While this piece used a specific set of qualities of movement, the system was less generalizable than desired. I realized that it was necessary to have systems that could more flexibly work with qualities of movement.

These aspects of the gesture recognition and movement quality analysis system made it more challenging to use the system fluidly in a rehearsal process. Many rehearsals experimented with pure choreography, rather than trying to integrate the technology from the very beginning. I believe that *Four Asynchronicities* would have had a stronger relationship overall between choreographic content, story, and digital performance extensions if the process of adjusting qualities of movement and gestures had been simpler and quicker to perform in the software, allowing for more experimentation in rehearsal rather than outside of rehearsal. The smooth integration of technological systems into a rehearsal process is a key element of performance systems, as will be discussed further in following chapters. The rehearsal process is a crucial test for any system: if it is useful in the piece and sufficiently easy to work with, it will be used in the final production. If not, the demands of a performance and rehearsal setting will generally lead to its removal from the piece.

3.3. The Disembodied Performance System: Extension of Virtuosic Physical and Vocal Expression

Some of these lessons from the Gestural Media Framework were quite influential on my work developing performance capture technologies for the Disembodied Performance System (DPS), created for Tod Machover's opera *Death and the Powers* (*Death and the Powers DVD (in progress)*, 2014; Machover, 2010; Torpey, 2009; Torpey & Jessop, 2009). This system, designed in collaboration with Peter Torpey, addressed how to map an expressive performance from the human body to other modalities including non-anthropomorphic visuals, sonic transformations, and robotic movement. The process of working on *Death and the Powers* significantly informed my thoughts on capturing and extending expression, as well as broader principles for integrating technology into live performance.

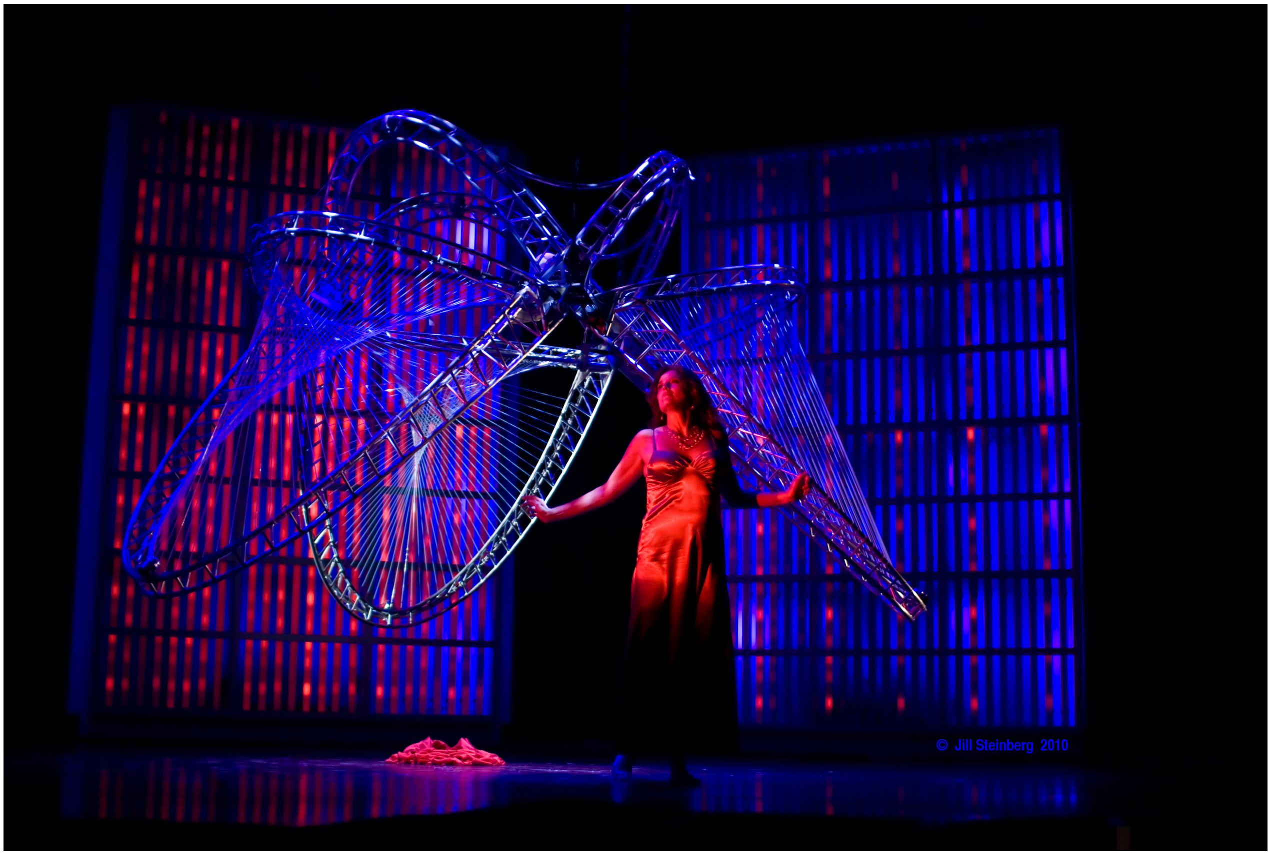


Figure 12. The set of Tod Machover's opera *Death and the Powers*
In *Death and the Powers*, the main character uploads himself into his house and communicates with his family through the scenic environment. Photo by Jill Steinberg.

3.3.1. *Death and the Powers*

Death and the Powers tells the story of a rich and powerful businessman and inventor, Simon Powers, who finds that he is nearing the end of his life. He seeks to extend his ability to experience the world, interact with his family, and carry out his business dealings by uploading his consciousness (his memories, emotions, behaviors, personality, everything that makes him Simon Powers) into a computer system integrated throughout his house. In the first scene of the opera, Powers uploads himself into the System, says, "See you later!" and disappears. The remainder of the piece focuses on Powers' family (his daughter Miranda, third wife Evvy, and part-cyborg research assistant Nicholas) and the world at large as they seek to figure out how to interact with Simon in his new form. They must figure out whether they believe that their husband, father, and mentor is indeed still present as the bookshelves or chandelier or walls. If he really is still in this new form, is he the same man or has he changed? How can they connect to him now? As the show progresses, Simon seeks to persuade his family that his new form of existence is substantially better than being a body "of flesh and blood," and attempts to convince them all to upload themselves into the System to join him. Each of the family members in turn has to make their choice between going into the System or remaining in their human bodies (Machover, 2010).

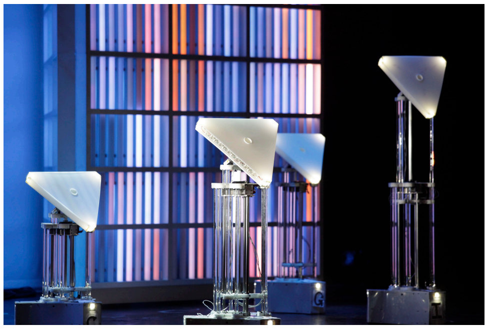


Figure 13. The *Death and the Powers* Operabots
The Operabots serve as both characters and scenery, commenting on the action of the opera. Photo by Jonathan Williams.

This storyline of the inventor Simon Powers is presented as a pageant play put on ritualistically by robots ("Operabots") in some future time when humans no longer exist. In a prologue and epilogue, we discover that the robots are scheduled to periodically retell the story of the man who uploaded himself into the system. However, they do not fully understand the purpose or the importance of the story, as they do not have a conception of death. They keep putting on the show and hoping that they will find some more clarity through their retelling. Four selected robots "transform" into the human characters to begin the show-within-a-show; others appear as characters and as scenic and lighting elements throughout. Overall, the robots serve somewhat of the role of a Greek chorus, commenting on the action while also being incorporated into it.

Death and the Powers was composed by Tod Machover, with a libretto by the poet Robert Pinsky. The initial production of *Powers* was directed by Diane Paulus of Harvard's American Repertory Theater, who also has significant Broadway experience. It was choreographed by the contemporary ballet dancer and choreographer Karole Armitage and designed by Hollywood production designer Alex McDowell. The first performances of *Death and the Powers* took place in Monte Carlo, Monaco in September 2010, with additional performances in Boston in March 2011, Chicago in April 2011, and Dallas in February 2014.

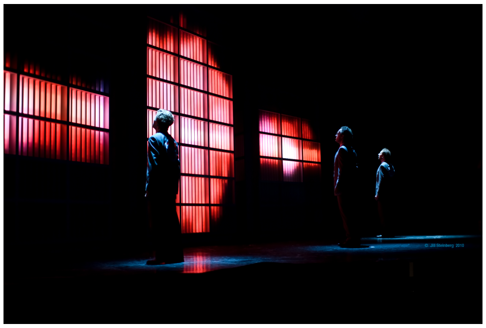


Figure 14. Simon Powers speaks through The System
Powers communicates with the Delegates from the Outside World through a language of color, light, and movement on the LED periaktoi. Photo by Jill Steinberg.

Theatrically, this plot presents a major technical and creative challenge, as the main character is first seen portrayed by a live opera singer (originally baritone James Maddalena), and then, for the majority of the show, embodied by the entire theatrical set and the performance space (Torpey, 2012; Torpey & Jessop, 2009).
The stage must breathe, react, be emotionally expressive, and be as compelling as a human performer. It must be able to convey the character of Simon Powers in a non-anthropomorphic form. Designed by production designer Alex McDowell, the main scenic elements that represent the character of Simon Powers are a set made of three periaktoi, each with low-resolution LED displays on one or two faces. While in the System, Powers primarily expresses himself through a non-anthropomorphic language of light and color on these walls. Occasionally representational video content appears to represent Powers' memories or when he conjures a set of images of himself to confuse Delegates from the Outside World. At other points in the show, Powers embodies himself in the Chandelier (a scenic piece and lighting element that reveals itself to be a musical instrument), in the Operabots, and in the movement of his processed voice around the theatrical space.

While it was a challenge to represent an expressive, emotional character primarily as a language of light and color on bookshelves, even more important was the need for this representation to not be static from performance to performance. One can imagine Maddalena performing the first scene and disappearing offstage, having technicians hit a few buttons to start a pre-programmed video routine, and requiring the remainder of the opera to be performed in fixed time to a click track synchronized with the video. However, that method loses the beauty and variability of a live performance. Therefore, we decided that everything that happened onstage to portray the character of Simon Powers, from the behavior of the lighting patterns on the walls, to the lighting and movement of the robots, to the movement of the surround sound audio in the theater, ought to be able to be shaped in real time by Maddalena's live performance. The media in the production should be different every night, able to vary fluidly and expressively. In order to achieve this goal, Peter Torpey and I developed the Disembodied Performance System, which incorporated wearable sensors, movement and vocal analysis strategies, a node-based mapping system, and systems for visual and sonic manipulation.

3.3.2. The Disembodied Performance System

In our production of *Death and the Powers*, the performer leaves the stage after he is "uploaded" in the first scene, but continues giving a live performance offstage. He enters a sonically isolated booth in the orchestra pit, and continues to sing the role of Simon Powers, actively performing, moving, and behaving as if he was still onstage. Parameters of his movements and voice are measured via wearable sensors and microphones and used to control the theatrical environment. I was responsible for designing and creating the sensors and sensor data processing that we used in the production, which included physiological and gestural sensing and analysis as well as vocal analysis. I also collaborated in designing the mappings from expressive performance information to output media.

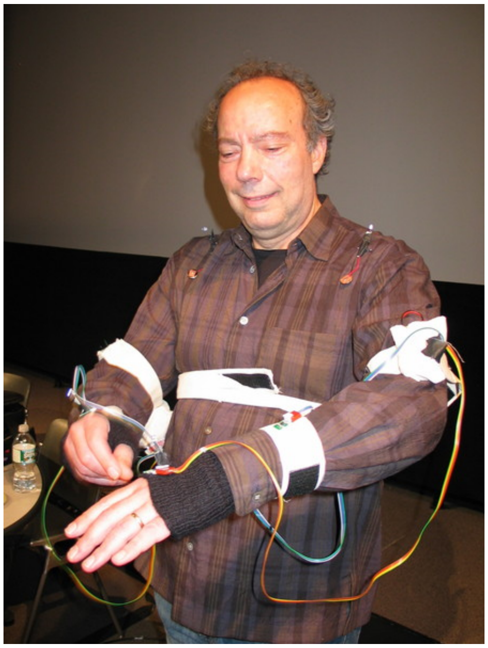


Figure 15. James Maddalena in prototype Disembodied Performance sensors
Maddalena wears accelerometers on his wrists and the backs of his hands, and a band to measure his chest expansion with breath.

We sought to find a minimal and reasonably unobtrusive set of sensors, in order to allow the performer to behave as naturally as possible and with as few physical limitations as possible. Importantly, we did not want to create a new gestural instrument that our performer would have to learn to use; instead, we sought to capture the performer's existing physical and vocal expressivity. As a professional opera singer, the performer playing Simon Powers uses a wide range of behaviors of his voice and body to communicate the experience and emotions of the character. We wanted to study and capture his pre-existing vocabulary so as to transform what he did naturally in performance into expressive gestures in the scenic environment.

In deciding on an ideal sensor set for this production, we found that one of the major aspects of the performer's physical presence is his breath. The rhythm and shape of a performer's breath reveals phrasing and emotion, as well as providing a sense of life. Thus, a wearable breath sensor was incorporated into the Disembodied Performance System to capture this expressive information. This sensor consists of a fabric band tied around a performer's chest, located at the point of maximal chest expansion during inhalation (this location varies from performer to performer). Incorporated in the fabric band is a flexible portion with a sensor that varies its resistance based on how much it is stretched. This sensor thus detects the performer's inhalations and exhalations. This information is captured by a Funnel I/O microcontroller board and transmitted wirelessly via XBee radio modules. This simple sensor was found to detect information about the breath of the performer and about his vocal phrasing that was more detailed than the information obtainable from audio or the score alone.

Additionally, three-axis accelerometers on the arms and the backs of the hands are used to obtain information about Maddalena's movement as he sings. Two separate accelerometers are used on each arm in order to capture the distinction between small movements made with the hands alone and larger movements that incorporate the entire arm. The accelerometers are sewn onto gloves and wristbands that can be easily worn in a known orientation. Data from the accelerometers is captured by a Funnel I/O microcontroller board worn in a pouch attached to the upper arm. That data is then transmitted wirelessly.

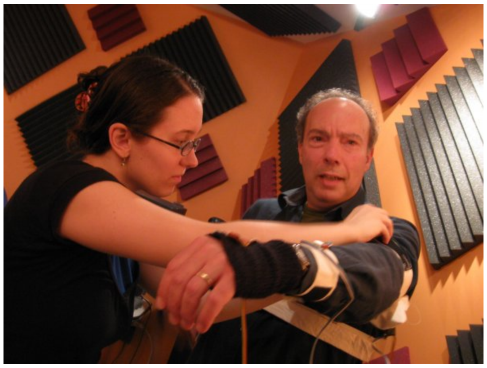


Figure 16. The author fits Disembodied Performance sensors on James Maddalena Accelerometers on the hands and forearms measure qualities of Maddalena's movement. Photo by Tod Machover.

Our choice of accelerometers as the primary sensors for movement capture was influenced by our goals of an unobtrusive system that could capture the performer's natural expressive movement. We determined that it was not necessary to capture specific gestures from Maddalena, as one might do in a more directly instrumental model; more important was the overall character and expressive quality of his natural motion while singing with emotion. We generally did not work with the raw accelerometer data, but instead computed features of the accelerometer data that reflected overall energy, temporal variation, and rate of change. In the Disembodied Performance System, I manually created algorithms to map this movement input to qualities in a modified Laban Effort Space of *weight*, *time*, and *flow*, abstracting the movement away from specific sensor values into a higher-level quality framework.

In our early work with Maddalena, we noticed that he was very expressive in the way that he shifted his weight from side to side and back to front. I developed pressure sensors that could attach to the front and back of his shoes to capture some of that variation in weight shifting. Each sensor consists of a pair of pieces of conductive foam with a circuit on either side. The foam reduces in resistance as more pressure is put on it. Thus, the sensor can respond to changes of weight (though with a slow recovery rate). In the production of *Death and the Powers*, we did not use these particular sensors, as the sound isolation booth in the pit was structured such that Maddalena preferred to sit during the majority of the production.

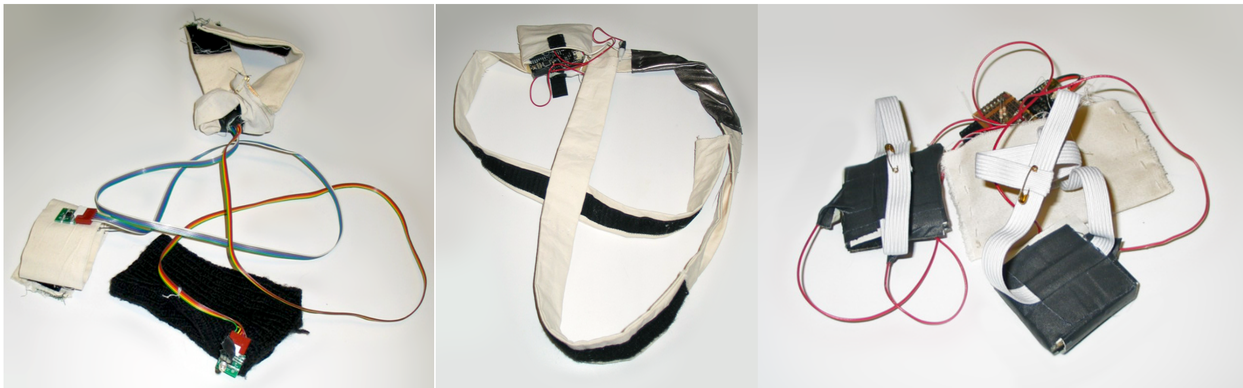


Figure 17. Prototype Disembodied Performance sensors
L-R: Armbands and glove for movement sensing, breath sensor band, and shoe pressure sensors for detecting shifting of weight

I also developed a system to examine expressive qualities of Maddalena's voice, including intensity, frequency, and timbre parameters (such as vocal harmonicity and dissonance). His voice is captured via a microphone in the booth and analyzed live via a Max/MSP patch. These analysis parameters can then be used as inputs for mapping. The analysis is performed over a short sliding window, so that the results of the analysis feel instantaneous and any visual response controlled by this analysis feels immediately synchronized to the live voice.

Using the Disembodied Performance Mapping System (described further in Chapter 5), we then mapped the resulting qualities of movement and voice to control a variety of output media. This mapping system communicates via the Open Sound Control protocol ("opensoundcontrol.org," n.d.) with a variety of other show control systems, particularly the `RenderDesigner` system created by Peter Torpey to create generative visualizations on the walls. It is important to note that the mapping between input performance information and output control parameters is not constant throughout *Death and the Powers*. All of our systems incorporate the concept of *cues* (or *modes*), different collections of settings for a particular point in the show. In the case of the Disembodied Performance System, the system has a collection of different mappings from input parameters to output control values. These mappings may change from mapping cue to mapping cue because different media needs to be controlled or the media has different control parameters (in the case of the visualizations on the LED walls, different visual cues might take a different set or different ranges of control parameters), or because the desired relationship changes between the performance data and the control parameters. When the system is given a particular mapping cue number, it switches to the desired mapping.

Importantly, this process did not rely on a specific pre-composed movement or vocal vocabulary, but rather on intuitive aspects of the performer's virtuosic expressive performance. We had several research sessions with James Maddalena before the start of *Powers* rehearsals to explore his natural movement vocabulary for performance and try to figure out how to capture his innate expressiveness as a performer. We did not want to create a system that was an instrument that Maddalena would have to play while he sang, or a model where he explicitly tried to control the multimedia elements. In fact, we actually did not give Maddalena a view of the stage, so he would not get into a feedback loop by trying to effect a specific change on the set. Instead, we asked him to perform as he naturally would if he were onstage (moving, gesturing, being vocally and physically expressive) and extended that behavior through the Disembodied Performance System.

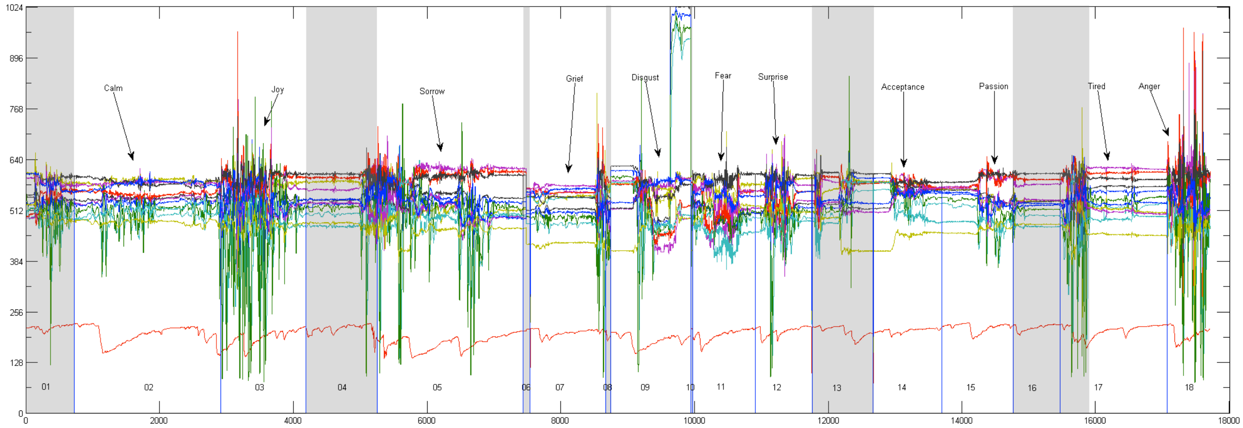


Figure 18. Trial data from Disembodied Performance sensors
In May 2009, we captured sensor data from Maddalena performing vocal exercises with a variety of emotions. Different emotions clearly showed different types of movement information. Graph by Peter Torpey.

During the rehearsal process for *Death and the Powers*, we constructed, shaped, and refined the mappings between Maddalena's performance and the multimedia results. We thus designed all of the *Powers* systems to be as flexible as possible during the rehearsal process, not requiring recompilation of code or stopping any system in order to adjust a mapping, change a visualization on the walls, tweak robot choreography, adjust sound manipulation, etc. In this fast-paced professional rehearsal process, there was no time to take a break and rewrite a code file for a mapping. Everything had to be able to be changed on the fly, so that we could see the immediate results while we were still running the same scene, and without the walls going dark in the meantime.

3.3.3. Additional Observations on *Death and the Powers* and the Disembodied Performance System

An interesting consideration for interactive performance systems that is demonstrated in *Powers* is the distinction between *modes* and *triggers*. A *mode* is a state of a system: a collection of continuous parameters or system settings. A mode defines the current rules and structures of the performance system, and persists until a change in modes. Sequences of modes create the larger structure of a performance piece. Each mapping in the Disembodied Performance System is actually a mode, as it defines the current interactive behavior until the mode is changed. A *trigger* is a discrete, momentary event that initiates a particular action. A trigger may change modes, or cause a certain event to play out in its entirety. Playing a specific video once on the walls, or a sample from the keyboard, is initiated by a trigger.

In *Powers,* the changes between different mapping modes are controlled by the second keyboard player, who plays a notated part that incorporates not only triggers for sound samples and sonic textures, but also triggers that change cues for the visuals on the walls and the connected performance interaction mappings. The Disembodied Performance System is passed its cues via the `RenderDesigner` system for the visuals on the walls, such that the changes of mappings for different visual looks is kept in sync with the changes of visual cues. Particularly as the `RenderDesigner` system incorporates fading behavior between visual cues, it was important to switch the DPS mapping cue only when a `RenderDesigner` fade was complete. Once the system has been put into a visual cue and associated performance mapping cue, it remains in that mode until the next trigger. This use of modes and triggers in the mapping system allows the performance extension to remain connected to the performance in multiple ways: triggers (commands for changes of state) are defined in the score and carried out in synchronization with the music, while each mode (a state) determines the input-to-output mapping and how the interaction is carried out moment-to-moment within that mode.

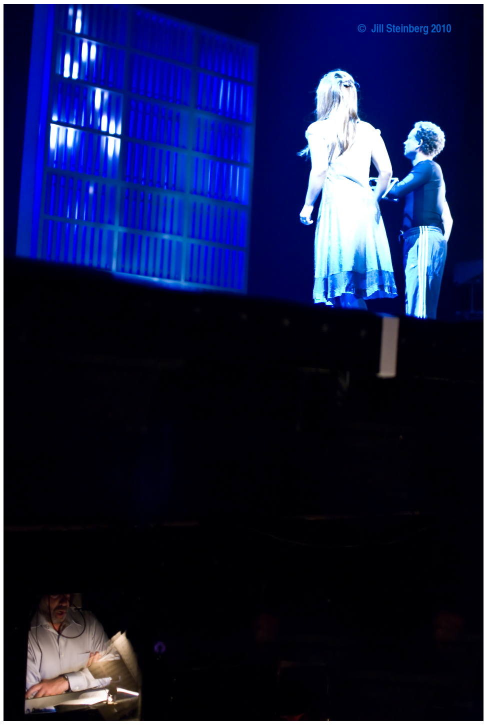


Figure 19. Simon Powers in the orchestra pit
James Maddalena performs as Simon Powers from the orchestra pit while the expressive qualities of his performance are translated to light on the LED walls on the stage above him. Photo by Jill Steinberg.

One key aspect of the performance extension system for *Death and the Powers* that differs from many other performance extension systems is that the live performer is not visible to the audience. The extended performance is visible and audible, but the human being creating that performance disappears after the first scene of the show. We were faced with the challenge of making the experience feel live and responsive even without directly showing the connection between the live performance and the expressive multimedia extensions of that performance. In early discussions of the system, we wondered whether we would have to put the actor playing Simon Powers somewhere visible or partially visible to the audience, perhaps in a semi-transparent box to one side of the proscenium. However, the libretto for the opera makes very clear that Simon Powers has "nothing like a body." He "cannot hear with ears, he cannot speak with breath" (Machover, 2010). We decided that we would be undermining the story and our intention in the design of the show if we revealed Powers and showed exactly how the behavior of his real, physical body affected the media elements. Instead, we decided that we had to truly transform his presence into something completely non-anthropomorphic.

Our goal was a sense of human liveness, even when one could not see the human that was generating the live behavior. In a way, it did not matter whether the audience was aware that the visuals they saw and the sounds that they heard were shaped and controlled by a live performer, as long as they bought into the premise that the set was the main character. If the scenic environment felt sufficiently live, reactive, responsive, and connected to everything else that was going on in the action onstage and to the music, we decided that it didn't matter whether the audience understood exactly how that behavior was technologically constructed. Indeed, if the audience was too concerned with the technologies behind the interaction, or with attempting to figure out "how it worked," we would have considered our work to have been unsuccessful. The point was the story, not the implementation.

Another key aspect of this virtuosic performance extension system was our decision not to create a new instrument for our Simon Powers to learn. We sought to abstract James Maddalena's physical and vocal performance away from his body in a way that allowed him to still perform as he would if he were onstage. The musical score for *Death and the Powers* was sufficiently challenging that it would have been cognitive overload to ask the performer to simultaneously perform a sequence of specific instrumental gestures. Instead, we chose to capture what Maddalena did naturally. As a professional performer, he is exceedingly adept at using his voice and body to represent the emotional state of a character. We chose to leverage his existing virtuosic skillset rather than to create a brand new expressive interface at which he would be a novice. This choice informed our sensing strategies, as well as our analysis techniques.

While the levels of abstraction in this system proved useful for creating meaningful mappings, the system was still tied to a particular sensor set, performance scenario, and set of expressive parameters. I realized that a more generalized system would be needed to quickly learn the relationships between different sensor inputs and desired qualitative spaces. This need for a more flexible system was a major impetus for the work described in this dissertation.

3.4. Conclusions and Performance Extension Principles from Prior Work

Through discussion of this set of prior projects, this chapter has presented the foundations of my work on voice and movement extension. Additionally, these prior projects have served as a framework for discussion of several principles for integrating technology into performance contexts in compelling ways, as well as of some of the needs of technologies designed for performance.

The Vocal Augmentation and Manipulation Prosthesis demonstrated the power of strong connections between physical actions and sonic reactions. By imagining gestural mappings at a high level (such as the pinching finger gesture capturing a sung note), the interaction design of this system was completely abstracted from any implementation details, allowing the development of compelling mappings between movement and digital vocal extension. The specific gestural vocabulary of interaction guided the choice of sensing systems, movement analysis techniques, and specific mappings. This instrument used a mix of discrete gestures for triggering specific actions (e.g. capturing a note) and continuous control over expressive parameters (e.g. shaping the amplitude of the held note).

The Gestural Media Framework and *Four Asynchronicities* demonstrated the expressive utility of high-level definitions of movement qualities and gestures, and the use of these abstracted parameters and gestures for creating mappings. Continuous movement qualities were also found to be more expressive than discrete gestures for augmenting a performance. By looking at the rehearsal process for *Four Asynchronicities,* we also see the need for more flexibility in fluidly experimenting with qualities of movement and gesture recognition while in rehearsal. Other principles that can be seen in analysis of the Gestural Media Framework include: discrete versus continuous actions; the challenge of gesture segmentation from a stream of movement; the necessity for different timescales of qualitative analysis; and the desire to have a unified system to handle both movement analysis and training as well as the expressive mappings of inputs to outputs.

The Disembodied Performance System and *Death and the Powers* also showed parametric definitions of movement and vocal qualities to be useful in an extended performance context. In this virtuosic performance, focusing on qualities of movement and voice rather than a particular pre-determined vocabulary of gesture allowed the performer to be naturally expressive in the ways he would in a standard theatrical context, while having that expression extended into a variety of media. The performer's feedback was carefully modulated so that he did not overly focus on his control of other media elements. Other relevant principles for developing technology in performance seen in *Powers* include: flexibility of mappings to allow development and exploration during rehearsals; the distinction between triggers and modes; and technology designed with a focus on a sense of liveness and storytelling rather than on "how the system works."

In analysis of the extended performance work in *Powers,* we also see the need for flexible systems that can easily adapt to a variety of input sensors and output media, as the Disembodied Performance System had only hand-coded input sensing and sensor analysis. For a more flexible or easily modifiable system, a user would want to be able to switch sensor setups, or adjust qualities of movement rapidly in the middle of a rehearsal or performance process.

More discussion on VAMP, the Gestural Media Framework, and the initial stages of the Disembodied Performance System can be found in my master's thesis (Jessop, 2010). Some of the principles inspired by these systems will be discussed further and expanded in the following chapter, as we explore the expressive elements of performance and interactivity, the development process of a performance or installation, and the incorporation of machine learning techniques into an interactive performance process.

4. Fundamentals of Designing Extended Live Performances

This chapter presents a set of principles for designing systems to support and extend live expression in performance and interactive installations. It also includes a set of the key principles, guidelines, and necessary questions that should be considered by practitioners seeking to design extended vocal and physical performances, particularly those that incorporate machine learning and analysis of expressive qualities. Through examination of the concepts of expression and liveness, as well as discussion of additional projects I have done at the Media Lab, I propose a set of questions and guidelines both for practitioners seeking to create extended live performances and for those seeking to design systems for extending live performance. I outline the Expressive Performance Extension Framework for incorporating interactive technologies, particularly machine learning of high-level expressive qualities, into performance and rehearsal contexts. Key aspects of this discussion include the expressive role of time, the use of regression rather than classification algorithms for analysis of expressive qualities, analysis of the relative skills of humans and machines in performance systems, and the definition of a framework that can support analysis and extension of both movement and the voice.

This chapter is divided into several sections:

- Elements of live performance and interactive installations that are sources of expression, particularly the role of different timescales in expression
- Mapping live performance into digital media while keeping a sense of "liveness"
- The development stages of a performance or installation work, with a focus on how technologies can support existing creative workflows
- Integration of machine learning techniques into the creative process: a suggested framework and workflow
- Goals of a system to support performance extension

4.1. Expressive Elements of Performance

Expression is one of the most challenging aspects to define of any kind of performance (dance, theatrical, musical, etc.), and yet it is one of the most significant. Juslin states in his psychological exploration of musical expression:

"…expression is largely what makes music performance worthwhile. It is expression that makes people go through all sorts of trouble to hear human performances rather than the 'dead-pan' renditions of computers; it is expression that makes possible new and insightful interpretations of familiar works; and it is on the basis of expressive features that we prefer one performer rather than another"(Juslin, 2003).

Before we further discuss practices and methodologies for extending live expressive performances through technology, it will be useful for us to examine in more depth some of the aspects of performance that inform a sense of "expression." It is important to note again, inspired by Juslin (2003), that "expression" is a multi-dimensional phenomenon, rather than a single entity of which a performance can have "more" or "less" or that it can be "lacking" or "full of."

4.1.1. Script and Score, Production Vocabulary, Interpretation, Improvisation

Expressive elements of performance can be seen at three different levels. The first of these I will call *score-level expression,* the expressive content inherent in the script or score, the fixed text or musical notation. This is emotional and expressive information conveyed through a score's content and structure. This layer of expression is generally consistent across different productions of a particular score or script. The second level of expression is *directorial-level expression*, and comes from the choices made for a specific production, such as the directorial decisions, choreography, and design elements. These elements of a production are the same for each performance of a specific production. They may support, complement, or counter the expressive content of a script or score. The final layer of expression is *interpretation-level expression*, and comes from performers' individual variation around the directorial content and the content of the score. This layer differs with every individual performance instance and is seen as different kinds of variation around the "set" elements of the piece (such as temporal or timing variation, dynamic variation, accentuation, articulation, variation in force or energy, etc.).

Different expressive performance and installation contexts have different balances of what content is static or "set" and what content changes with every performance. For example, one production might have the direction for a performer to "enter from stage right and cross to center, humming." Another production might have the direction, "enter from stage right looking back over the left shoulder, take ten beats to cross to center, stepping rhythmically in an even tempo, and cheerily humming the first bars of Beethoven's Symphony #5." In the first instance, the expression in the performance of the events may be said to come more from the individual performer's interpretation of those directions and the way that he chooses to carry them out on a specific evening. In the second instance, more of the content of the piece is set by the director. The performer's variation will come in more subtle differences in timing, emotional content, and articulation of the actions.

Alternately, a piece might be completely improvised, with potentially all content and structure being developed on the spot by the performers. Other improvisational pieces could have a predetermined structure and improvisation within that structure, or a specific vocabulary (such as a sonic, melodic, or physical vocabulary) that performers use for their improvisation. All of these will create different relationships between what types of expression arise from score-level, directorial-level, and interpretation-level content.

With regard to an individual performer's expression, much of the expressive content is communicated by how the performance varies moment-to moment in the context of the constraints of the script-level and directorial-level content. Juslin identifies factors that relate to expression in musical performance, which can be extended to expression in broader performance contexts. He proposes a psychological model of describing musical expression through five separate aspects of performance. *Generative rules* are the transformations of performance (through changes in dynamics, tempo, and articulation) that help the listener understand the musical structure. *Emotional Expression* relates to the variations in performance that are used in order to convey a particular emotional experience to a listener. A performer can remain true to the score-level information in a piece while still having the freedom to shape the overall mood of the piece. *Random Variability* describes the variation in acoustic parameters that comes purely from the limitations of our biological motor systems. This kind of variation, particularly in timing, contributes to the sense of "humanness" of a performance. *Motion Principles* refers to the shaping of dynamic and tempo patterns either to correspond to human movement patterns (a ritardando generally having the same shape as de-acceleration from running, for example), or due to human movement patterns. Finally, *Stylistic Unexpectedness* addresses the ways in which expression can be created by a performance violating expectations of a particular style of music or performance conventions for a particular section or moment. While all five of these factors are intended to reflect different psychological principles and different neurological pathways, it is important to note that they all refer to variation in timing, dynamics, articulation, etc. More generally, it is the small variations from the "score" or the "expected" performance of a piece and the actual details of a specific performance that help give rise to expressive content. It is also important to note that this model of expression presents expression as a concept that relates both to the performers (in their intention and the details of their performance) and to the audience (in their interpretation and their experience of the performance).

4.1.2. The Role of Different Timescales

As was explored in Chapter 2, the different features of a physical performance that convey expressive content are likely to have layers of expression at many different timescales. A note or a particular body shape can be experienced at a precise instant and described with certain parameters. However, the majority of expression is not reflected in a static image, but instead comes from how all the parameters vary, grow, develop, or stay fixed over a variety of timescales from a brief moment, to a phrase, to an entire piece. Similarly, the expression in a piece comes not only from the fixed parameters of a piece as a whole (such as its pace, or smoothness, or dynamics), but also from the momentary variations and deviations from those defined standards, and in the ways that those standards change throughout a piece.

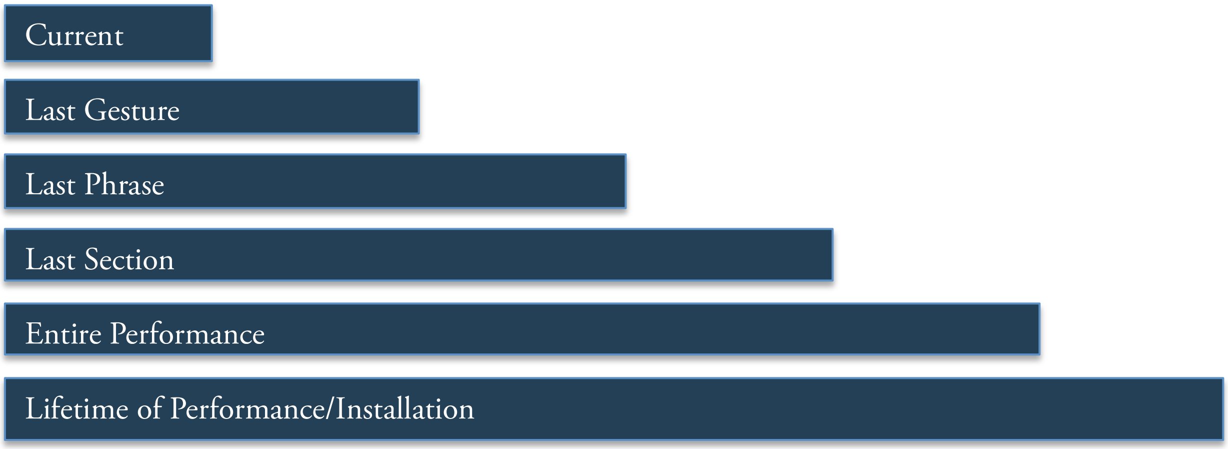


Figure 20. Relevant timescales of expression in performance

For example, a particular dance performance might have a quick, intense rhythm throughout; however, the moment-to-moment expressivity in the dance comes from how the movement deviates from that standard through slower or faster subdivisions of tempo, the amount of fluidity versus rigidity in a motion, or the amount of energy in a particular movement. If a performer is moving glacially slowly and then rapidly flicks his hand, that flick conveys a different kind of expressive content than the same flick after he has been flicking his hand repeatedly and swiftly. In addition, the expressive norms established in a piece may change from section to section, or from one moment to another.

It is important that a system can handle definitions of and recognition of expression over multiple different timescales. To extend the possibilities of expression recognition systems even further, systems should analyze temporal behavior not only to determine the performer's current point in an expressive space, but also to analyze particular shapes or features of trajectories through expressive spaces. A system to appropriately process expressive information should therefore recognize not only the general expressive parameters of a piece, but also when and how the work varies from that baseline. These various temporal aspects of expression should be included in their definitions of a multi-dimensional, continuous expressive space.

In addition to the different scales of time that may need to be considered within a particular piece, it may also be important to take expressive measurements across different points of the life cycle of a performance or installation. How is the perceived expression of one visitor to an interactive installation affected by what other participants have done earlier? If many participants have been observed making very small movements, the impact of one participant who makes a very large movement should perhaps be highlighted. As another example, one of the ways in which "expression" is visible in a performance is the ways in which that performance differs from night to night. If one sees a single instance of a performance, it may not be clear how much of the performer's behavior is scripted and completely set, and how much variability is possible and "special" for that particular presentation of the show. Particularly in a performance that incorporates some layer of improvisation, the expressive variation in a given single presentation may not be clear. What if some notion of how a performance differed from night to night could be preserved and those differences could be seen in a specific performance?

4.1.3. Interaction Between Performers

Another key aspect of how expression is established and communicated in a performance piece is the interaction between performers, if there are multiple performers. This is an aspect that is frequently explored by choreographers and performance-makers, but less often by researchers studying interactive or controllable systems. When multiple people are onstage, we will interpret movement differently depending on the performers' physical relationships to one another. For example, imagine a performer who starts with her right hand by her side and raises it slowly, palm up, while looking out past her hand. If this performer is alone onstage, there is a certain layer of expression that is conveyed by this gesture. However, place this performer onstage with another dancer to her right side, and her raised arm will appear to be reaching out for the second dancer. Additional levels of metaphorical and symbolic context are established by the presence of the second performer.

Brown, in her discussion of various aspects of Labanotation, presents a set of movement relationships between performers: "Partners may approach, meet, move together, part, dance near, by the side of, behind or in front of each other; lead, follow, move together, in canon or in opposition; address, touch, support, surround, grasp, carry each other or an object" (Brown & Parker, 1984, p. 26). There is a tremendous amount of content that is contained in these relationships between performers, which may be relevant to consider when designing extended performances that include multiple performers.

In developing an interactive extended performance or installation that features multiple performers or participants, one question to explore is to what extent the interaction is shaped by aspects of each participant's individual behavior, by aspects of the overall stage picture, and by aspects specifically related to the interaction between participants. For example, let us imagine an interactive installation where the participants' movement is used to affect a soundscape. In one version, each person moving in the space controls a different instrument or sound type in the soundscape: this person affects the low bowed drone, that person affects the high marimba-like melodies. As each participant moves more rapidly and with larger movements, their associated instrument is brought out and given a denser texture. In another version, the overall activity of the soundscape is controlled by the shared energy of both participants together. In a third version, melodic fragments arise from the drone as the two participants move closely together and start to come into rhythmic synchronization, moving at a similar speed and scale of movement. In all of these versions, the overall impression of interactivity will be shaped by the combined behavior of all participants, but the experience of each mapping may be quite different. In an interactive installation, these mappings will draw out different kinds of interaction between participants. In a performance, these mappings will draw attention to different aspects of the behavior of the performers.

This dissertation primarily explores single-person experiences, including both performances and installations. Even in situations such as *Death and the Powers*, which included a several-person cast, the performance extension technologies were primarily designed to extend the performance of a single actor in the cast. While the majority of the actors have standard roles, the actor portraying Simon Powers has his expressive movement and vocal behavior captured and used to control and shape aspects of the visualizations on the walls, the movement of the robotics, and the movement of the sound in the space. An interesting point to note in the case of *Powers*, however, is the behavior of the Operabots, which are actually controlled by multiple performers and technicians, as well as have autonomous behavior that has been scripted by the director and choreographer. Aspects of the singer's voices control lighting and small movement patterns on the Operabots; simultaneously, major movement patterns of the robots are puppeteered by technicians, and the current cue state (and thus autonomous behaviors) of the robots is determined by an overall system operator. In a way, an individual robot thus serves as an extension of several people's performances. To refer to the terminology of types of performance defined in Chapter 2, these robots are a combination of instrumental, stage managed, and static performance systems.

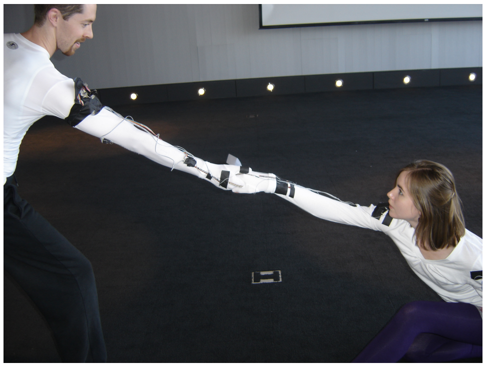


Figure 21. An evocative gesture between performers in *Four Asynchronicities*

Four Asynchronicities on the Theme of Contact is one set of performance works described in this dissertation that explored extending multiple performers simultaneously. These dance pieces included computer recognition of some gestures that were specifically designed because of their metaphorical content in the interaction of two performers (such as a hand grasp and spin). This piece also included other performance extensions that reacted to multiple performers. For example, the fluid dynamics visualization in Movement 4 had different colored points responding to each performer so the patterns of the visualization were affected by each performer's individual movement but the overall impression was determined by the combination of all the performers.

4.1.4. Interaction Among Performers, Space, and Objects

Another important category to address in discussion of expressive aspects of a piece is the interaction between performers and space, as well as between performers and objects. As with multiple performers on the stage, other scenic elements and objects that share a space with a performer will change the way in which the performance content is interpreted and seen as expressive by an audience. Semantic and metaphorical content may vary significantly with changes in the performer's spatial relationships to elements of the performance space and to the space as a whole.

The nature of an object can influence our interpretation of a performer's movement and the goal of that movement. A gesture that may appear abstract when performed in free space may be quite clearly instrumental when performed in relationship to the object that it manipulates. For example, in a production of *Our Town* performed at the Huntington Theatre in 2013, the actors performed on a bare stage with a stylized pantomime vocabulary for the majority of the production. The actual gestures were not all clearly connected to specific actions, but created an overall impression of work and activity. However, in the climax, a curtain was raised to reveal a highly detailed, period-accurate set, on which the final scene took place. The very same gestures that had seemed abstract throughout the show suddenly took on very specific meanings when combined with physical objects: pumping the water for the sink, flipping the bacon, opening the cabinet (Cromer, 2013). As expression is partially related to the audience's experience and interpretation, the presence or absence of objects as part of a movement can strongly affect expression.

Similarly, the nature of an interaction with an object, whether that interaction is through a performer's focus on the object or through actual handling of the object, changes the way that we interpret particular physical or vocal gestures in performance. The hand raising gesture described in the prior section would similarly take on additional layers of metaphorical and communicative meaning if the performer raised her hand slowly to an object in the space.

For the purposes of this dissertation, I have primarily limited the movement space examined to the space of free movement, where a performer's movement through space does not include the manipulation of physical objects. However, a performer's interaction with objects is important to consider in a general model of expressive elements of performance.

Spatial relationships between a performer and the performance space also shape the expressive content of a piece. As discussed in Chapter 2, one of the primary components of early dance notation systems is the notation of movement patterns in space, or "floor plans." In these notation systems, a dance is defined not only by its steps, but also by information about where dancers should stand and face and travel. The scale of a performer's movement in relationship to the scale of a space, a performer's orientation toward a particular direction in the space, the amount of the space used by a whole performance, all are relevant and potentially expressive.

The proximity or distance of a performer to scenic elements can also convey emotional or expressive content and can influence the details of a performance. For example, imagine a series of movements of the upper body, while a dancer remains standing with her feet in a fixed position. Imagine a performer carrying out this sequence in the middle of a large, bare stage. Now imagine the same sequence of movements while the performer is balanced on top of a high, small platform. Now imagine the same sequence performed within the confines of walls close to the performer on three sides, or in the corner of a room. The sequence of actions is identical. Indeed, the specifics of the performer's actions could be identical. Yet, these three performances will still convey different kinds of emotional content through the spatial relationships that are established.

Another potentially expressive aspect of performance is the performer's spatial relationship to the audience. Is the audience far from the performer, sitting in a theater watching a performer on a standard proscenium stage? Is the audience seated just a few rows from a performer? Are performers moving through the audience, or is the audience free to move through and around the performers? The impact of a dancer, singer, or other performer's expressive behavior may differ depending on this spatial relationship to the audience. For example, in a large proscenium hall, a tiny movement or a whisper may not be perceived by the audience and thus will not convey expressive information. In a tiny room where the audience can stand right next to the performer, every subtle detail of her movement or voice can be perceived.

To extend this line of questioning about audience/performer spatial relationships even further, what about when the audience is not physically in the same space as the performer, as is now possible through technology? What aspects of expression can be conveyed when the audience is even more physically distant? The *Death and the Powers* global interactive simulcast, discussed in Chapter 6, explores this question.

4.1.5. Interaction in Installation Spaces

In the context of an installation (particularly an interactive installation), these questions become more complex. Generally, the audience members in an interactive installation serve as the "performers," shaping the behavior of a system or environment through their actions. However, typically these performers do not have prior experience with or knowledge about controlling the interactive system. Frequently, they are given limited instruction and allowed to find their own patterns of interaction with the system. Certain levels of expressivity are set up by the construction of the system and the space of the installation while others are shaped by how the participants behave (which is, in turn, influenced by the design of the system and the space).

An interesting aspect of interactive installations is the degree to which visitors may be influenced in their beliefs about the nature of the interactivity by the behavior of others interacting with the system. A kind of meaning is often created by how you see someone else interacting with a system. Another's interpretation or misinterpretation of how a system works can inform your own beliefs and experiences of that system.

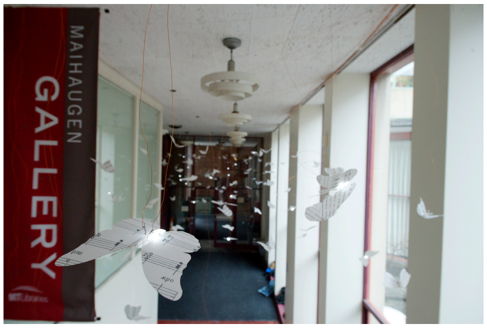


Figure 22. Bibliodoptera installed at MIT
The vellum butterflies are printed with musical scores and text from books. Photo by Andy Ryan.

As an example of an interactive installation where different meanings were constructed through solo or group interactions, I present Bibliodoptera, an interactive installation created with Peter Torpey for MIT's 150th Anniversary Festival of Art, Science, and Technology (FAST). This installation's reaction to visitors was quite subtle and open to different interpretations about the primary cause.

This installation, originally located in a corridor between MIT's Hayden Library for the sciences and humanities and the Lewis Music Library, consisted of a cloud of vellum butterflies hanging from the ceiling. These butterflies were printed with musical scores and text from books in the libraries, forming an unobtrusive and beautiful symbol of the knowledge of the arts and humanities that have been developed and pursued at MIT. A subset of the butterflies also contained individually addressable LED lights. Trajectories through the cloud illuminated to guide passersby along the length of the corridor, triggered by proximity sensors at each doorway. When a person entered either side of the hallway, one of several paths of illuminated butterflies came on one by one down the hallway to the opposite side, and then slowly decayed. This subtle interaction was designed to create a sense of life and activity in the installation that was connected to visitors to the hallway. Bibliodoptera was installed in February 2011 and remained in place until June 2011. During the months that the installation was in place, it transformed the hallway from a simple passageway to a space for a experience.

This installation's simple and subtle interaction pattern was designed for the relatively sparse traffic present at a normal moment in the hallway, allowing a single participant to create a hallway-wide effect. However, a very different experience was created in the last weekend of FAST, when installations around campus were open and the general public was invited to come see the installations. As thousands of people passed through the installation space over the course of a few hours, the hallway was continually packed with people. The resulting behavior of the installation was that the LED trajectories continually flickered, as an individual trajectory would not have time to complete and fade out before being retriggered repeatedly. Given this rapidly changing LED behavior, visitors attempted to determine what was activating the butterflies. Many would wave at an individual lit butterfly, to see if their movement affected the butterflies individually. Perhaps the most interesting group interactivity effect, however, was the moment that any visitor thought that the

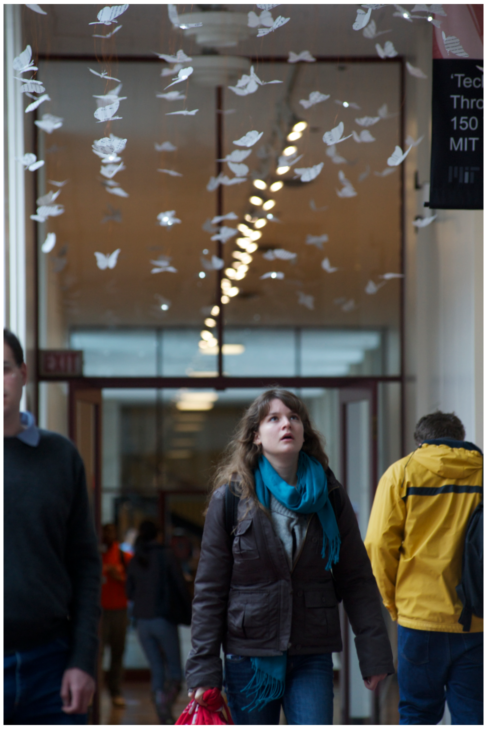


Figure 23. Visitors observe Bibliodoptera
Photo by Andy Ryan.

butterflies were lighting up in reaction to sound. Once one visitor began to clap, or sing, or call to a butterfly, that interaction pattern would rapidly propagate down the hall as other visitors assumed that sound was being sensed by the installation. This is an example of how visitors' perception of interactivity of an installation can be shaped by what they perceive others doing around them.

4.2. Extension of Performance: A Discussion of "Liveness" and Mappings

In exploring the needs of technologies that extend live performances, it is necessary to explore what is essential to a "live" performance, and how to extend that sense of liveness. This section explores the concept of "liveness" as well as how different mapping and control strategies can affect the sense of liveness of performance extension systems. This section also includes case studies of two of my prior projects, the *Sleep No More* Extension and the Chandelier, a Hyperinstrument designed for *Death and the Powers*. These projects bring up some relevant issues for designing interactive technological systems for performance extension, particularly the question of how to strike a balance between human and computational control of these systems.

4.2.1. How Do You Know It's Real?

One of the key issues in the practice of technologically-extended performance is the concept of liveness. As Mark Coniglio says in *The Importance of Being Interactive* (Mark Coniglio, 2004), "What we love about digital media was precisely what made it inappropriate for use in a live performance—it is indeed always the same." A live performance changes along myriad dimensions night to night and moment to moment, subject to the state of the performer, the relationships between performers, even the relationships between performer and audience. Timing varies along both large and miniscule scales, moments of emphasis and focus vary, details of position and shape vary, qualities of movement and voice vary. In contrast to this, standard digital technologies (such as recorded or pre-composed digital music, digital video and projections, preprogrammed sequences for lighting or scenic movement or robotics) are generally stable, ideally identical on every repetition. What characteristics do digital technologies need to have in order to integrate smoothly with live performance, enhancing rather than diminishing the ephemerality and variability that makes live performance compelling?

The question of liveness is always present in an interactive work. Particularly in pieces where digital media responds to or is shaped by live movement, we must ask the question: how do we know it's real? How is the piece different because of the interactivity? For example, in the field of dance there is a long tradition of choreographing works to accompany a piece of music. The musical piece is now generally played back as a recording, and the performers synchronize their movements to the music. The piece can be constructed and choreographed such that the movement corresponds in intuitive ways with the sound that is heard; the performer raises his hand sharply on a particular beat of the music, or circles his torso to a certain swoop of sound. Audiences are used to this music-movement relationship: if the movement corresponds to the music at a given moment, typically this is because specific movement was choreographed as to correspond to that music, and rehearsed and precisely performed to line up temporally with that music. When we switch to interactive dance-music systems, we flip that situation on its head. A particular beat in the music may only occur *because* the dancer raises her hand sharply. In what cases should we make this transformed relationship between performer and digital media visible to audiences (and how should we do so)?

The sense of liveness can also be closely related to a performer's experience of control over the system (or a visitor's experience, in the case of an interactive installation). How direct is the relationship between what the performer does and some manipulation of the result? Does anything the performer does have some result? Is it possible for the performer to have control over when something should change? A strong and direct sense of control is exceedingly helpful in giving an installation a sense of liveness, and is beneficial in a performance context as well. For example, in the second movement of *Four Asynchronicities on the Theme of Contact* (discussed in Chapter 3), while in actuality the content of the soundscape was determined at every moment by the qualities of the dancer's movement, the transitions between different "regions" of the soundscape did not always feel under the performer's control. It was not clear when, in the course of a movement, the dancer would trigger new sounds to come in and out of the soundscape. Due to this, the music and the movement felt less closely coupled, and thus less clearly interactive.

One of the strongest signifiers of liveness through immediate control is a system's reaction when the performer does nothing, and its reaction when the performer begins and ends units of activity (either physical or vocal). Does the system resolve into one mode when the performer stops moving or singing, and immediately react again as soon as the performer begins a new gesture or phrase?

As discussed in Chapter 2, technological systems can have many different relationships to a live performer. In some systems, the output behavior is not influenced by a live performer's input. In others, the systems follow Rowe's "player" paradigm, incorporating information from the live performer but behaving according to their own goals and intentions (Rowe, 2004). In other instrumental models, the system follows the player in a repeatable, learnable, and controllable way. The sensation of "liveness" of a particular technological system will vary depending on where it falls in this space of reactivity and interactivity.

In systems that primarily have their own independent behavior, where the input of the performer is not clearly influencing the behavior of the system, it may be challenging to see that the system is changing its behavior live. For example, take the case of a system that is randomly generating sonic material, without any external input or control from a performer. Despite the fact that the sound is indeed being created "live," differing from performance to performance, it is not intuitively clear how the experience of hearing one particular iteration of this material generated live would differ from hearing a pre-recorded version of this accompaniment. An exception to this situation might be if there are human performers completely improvising their own material based on what they hear; in this case, the difference in sonic accompaniment from performance to performance might evoke a greater variety of reaction and focus from the human performers. However, even in this case it may not be obvious that the computational system is changing live as well as the human performers.

4.2.2. Complexity in Mapping and Liveness

A performance or installation's sense of liveness is closely related to the mapping strategies that are used to connect input data gathered about the performance to the resulting output media. In particular, there is a tension between the simplicity or complexity of a mapping and how clear it is that the output behavior is controlled live by a performer or visitor. In a very simple one-to-one mapping, the connection may be clear, but may not be expressive. In a very complex many-to-many mapping, there may be room for significant expression, but the relationship between action and interactive response may not be clear.

In addition, even simple mappings require many questions to be considered to create interesting connections between a live performance and the digital extensions of that performance. For example, take the basic premise for *Death and the Powers*, that the performance of a live opera singer needs to influence the visuals on the stage, the movement of the robots, and the sound in the space. We might first start thinking about very basic mappings, tying the input from one sensor to the control for an output parameter. Perhaps we connect the instantaneous height of the singer's right hand to the height of a glowing element on the LED walls. This is a straightforward one-to-one mapping that, while perhaps not particularly interesting or expressively meaningful, is immediately responsive and is directly tied to the live performance. It is important to note that even though there is a simple mapping in place, there are still a wide range of mapping decisions being made. First, we are choosing to connect this particular input parameter (hand height) to one particular output parameter (the height of a visual element). Second, we are explicitly or implicitly defining how we think the ranges of those two parameters should be related. What are the lowest and highest inputs we think we're going to have? How high and low do we want the patch of light to move? Do we want an increase in hand position tied to an increase in height, or to a decrease in height? (That is, what direction is the mapping?) Are the two numbers to be mapped linearly or exponentially or with some other function? Do we always want the input to be used, or if the hand height is below a certain level do we want to treat that as a "hand down" baseline? What amount of variation in the input comes from noise in the sensors that perhaps we might not want to directly translate onto the walls?

4.2.3. If a Stage Manager Can Do It, She Should

A major factor to keep in mind when selecting methods of technologically extending a performance is which kinds of interactive performance extension technologies are required to achieve the desired performance effect. In particular, we should take care to select the simplest technological systems possible. In this process, we should remember that artistically skilled humans are extraordinarily capable at performing certain kinds of sensing, recognition, and control tasks, and use technological systems for sensing and control only when that will enlarge the capabilities of the performance extension system in a positive and necessary way.

Suppose we have a performance piece in which we have decided that every time a dancer raises her hand, the sound of a bell should be heard. We could approach this as a straightforward gesture recognition problem, outfit the performer with wearable sensors or incorporate a computer vision system, train a system with many examples of hand-raising gestures, and set up the system to play back a prerecorded bell sound upon recognition of the trigger gesture. An alternate method to achieve the same performance goal is technologically easier, quicker to implement, and more reliable: have a stage manager or other technical operator push a button to trigger the prerecorded bell sound when he sees that the dancer is raising her arm in the desired gesture. People are extraordinarily good sensors; they can anticipate movement, correctly evaluate the precise temporal relationship necessary between a gesture and the resulting sonic action, and quickly pick up on many variations of a gesture. Particularly if the choreography is pre-composed, a human operator can know when the desired gesture is about to occur and react with great precision in timing. Even in an improvisational situation, if the human operator only has a few elements to track, this situation is easily achievable.

However, suppose we wanted to know more about the movement than simply that the dancer had raised her hand. What if we wanted to track where she was in the process of raising her hand, and how quickly she was raising her hand, and how fluidly or jerkily she was raising her hand? What if we wanted to use this kind of information to change what kind of bells were played, or how long a note would be struck, or how much a sound would be distorted? Suddenly, we need to keep track of different kinds of continuous information, not only know a simple yes or no about whether the gesture has been performed. We could give the stage manager a set of sliders that he could move to reflect his perception of these parameters. However, as the number of variables increases, the stage manager is not going to be easily able to visually track and physically model all of those variables. Layers of the nuance of the movement may be lost. Here is a situation where it makes sense to introduce a computer system to track the movement, since it has the capability for analytical precision and temporal specificity.

Similarly, even if we were only using specific gestures to trigger specific sounds in a constant relationship, what if we wanted to use a fairly large vocabulary of movement triggers and to have several performers improvising with that movement vocabulary? This wealth of information would also quickly overload the cognitive processing capabilities of a human sensor (system operator), and might be another situation where a technological recognition system would be useful.

Let us return here to the distinction between "gesture" and "quality," as defined in Chapter 2. A gesture is *what* is done physically or vocally, while qualities are *how* things are done. In exploring mapping strategies for extension of physical performance, specific gestures are more likely to be mapped to discrete actions, while qualities have the ability to be mapped to continuous behaviors. Especially in the continuous capture of high-level physical parameters such as movement qualities, a computational system can give us more flexibility and precision than a human could accomplish.

4.2.4. *Sleep No More* and the Operator Model of Human-Machine Task Sharing

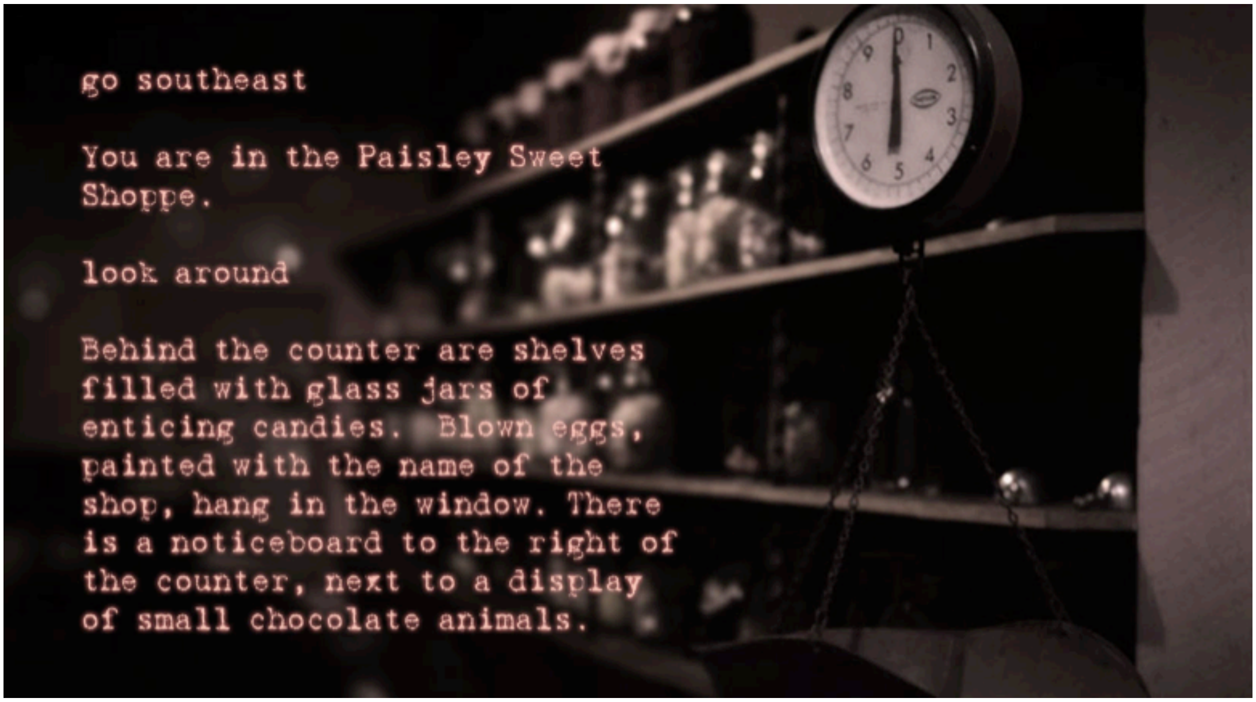
I argue that a basic idea behind the balance of human and machine interaction in the extension of a performance piece is that we should let humans do what humans do well, and let machines do what machines do well. In combination, this strategy has the potential to create the most robust and sophisticated systems. I came upon a more complex example of this principle in my work on a digital extension of *Sleep No More* (SNM), the hit NYC show developed by the British theater group Punchdrunk. This online extension of *Sleep No More* required a complex combination of computer systems (good at working with large amounts of data and complex rulesets) and human operators (good at improvisation and storytelling) to produce interesting performance results.

In *Sleep No More*, the audience dons masks and enters a 100-room warehouse space in Chelsea, meticulously decorated with a filmic level of detail and filled with a pervasive soundscape of music and audio. As individual audience members make their own way through the space, they encounter performers telling a story primarily through dance and stylized movement that combines Shakespeare's *Macbeth* with a variety of elements from Hitchcock movies. A key aspect of this performance is the audience's autonomy. Each masked audience member is free to wander around the space as he chooses, to follow actors, to open drawers or read books, and to encounter his own subset of the show. Since many different scenes take place in the building simultaneously, no one audience member can experience the whole show, only his own path.

Punchdrunk came to us at the Media Lab in the fall of 2011 and proposed a collaboration. They wanted to experiment with new ways for people to experience their production. Since the setup required for the show is so elaborate, it is very challenging to tour the production. Punchdrunk had tried filming their work in the past, but a static video was never able to capture the experience of what it feels like to be at a Punchdrunk show. They wanted to try a new model, where we would pair remote audience members with audience members at the actual show, giving both people access to new story elements and a different way of experiencing the piece. Working closely with Punchdrunk, we developed this new experience and ran a pilot version of it for five shows in May 2012, connecting several pairs of audience members per show.

We decided to create an online analog to *Sleep No More* to give remote audience members a way to have their own SNM experience. We decided that this should not take the form of a head-mounted video from the onsite partner, as this would not give the online partner the feeling of autonomy that is such a crucial part of going to a Punchdrunk show. Instead, we decided to create an online world with a variety of rooms, some of which had equivalents in the real space and some of which were spaces only hinted at by the live show. We also chose not to represent these spaces by computer animation or detailed pictures. When you are physically at *Sleep No More*, the space feels dark and infinite. You could go anywhere and do anything. There are no obvious limits and boundaries. In attempting to figure out how to translate that impression of space onto a computer screen, we decided to take our inspiration from old text-based computer adventures. With a black screen, a blinking cursor, and the sound of a river in your ears, you don't know the boundaries of the space. You have to learn how to interact with the world you are experiencing. We created this online world using sparse but evocative text, a continual soundscape experienced through headphones, occasional imagery, and moments of live and pre-composed video content. The spaces and what could happen in them were influenced by the actions of the online participant, the behavior of the onsite participant, the two participants' interactions, and the timeline of the real show.

We also explored a variety of ways to connect the two partners, without disrupting the normal show; the other 395 audience members present every evening needed to have no idea that something else was happening to only a handful of participants. The rules of the show therefore could not be changed: no speaking out loud, no pulling out a cell phone. We decided to incorporate the onsite participant's mask as one mode of communication, by enhancing the mask with bone conduction transducers to send messages and vibration into the participant's head while leaving his ears free. The mask was also augmented with various sensors (such as audio sensors, galvanic skin response sensors, and heart rate sensors) to detect aspects of the onsite participant's experience that could be communicated to his online partner. The position of the onsite participant in the real world was also tracked. Additionally, at times each pair of participants would be connected through special objects in the physical space ("portals"), such as a typewriter in a private room that could type what the online participant was writing, a robotic Ouija board that would spell out messages from the online participant while the online participant observed via a webcam, and a mirror that would write messages from the online participant in a ghostly hand.



go southeast

You are in the Paisley Sweet Shoppe.

look around

Behind the counter are shelves filled with glass jars of enticing candies. Blown eggs, painted with the name of the shop, hang in the window. There is a noticeboard to the right of the counter, next to a display of small chocolate animals.

Figure 24. Web interface for the *Sleep No More* Extension
Through sparse imagery and evocative text, participants were drawn into a virtual, text-based version of the world of *Sleep No More.* Graphics by Peter Torpey.

My own work for this project primarily focused on the content of the online world and developing an interactive fiction engine to set up rules for how the online world would be shaped by the online participant's actions, the show's timeline, and the onsite partner's actions. As part of this, we turned our narrative and descriptive content and rule systems into a 5000+ line script file in a custom markup language (JEML, created with Jason Haas) that held the descriptions of the spaces in the world, the items in the world, the characters that could be found there, and most of the logic about how online participants could interact with different parts of the world and what would happen when they did. The script file also contained the rules of what imagery and videos would be shown when, in what style the text would appear, the material used for the sonic experience, and how one could move from location to location. This script file additionally held information about when to connect participants to a particular portal and what actions would take place in the online world given the behavior of the real-world participant. More broadly, this script laid out the rules of the story: the results of specific actions, with the preconditions and current states that would shape those results. All of these things were considered to be part of the script rather than of our more generalized story system, since they were specific to the SNM storylines and to the material created to convey those stories. Our story system knew how to parse the rules and the descriptive content, but that parsing and rule-following was generalized away from any of the specific story behavior.

One of the major challenges in this process was that we did not want the system to reveal itself to be a limited computer system. We set the goal that we never wanted the system to say, "I don't understand you," or, "Please type another command using one of these keywords," or "You're not allowed to do that." However, there do not yet exist any systems that can perfectly parse unconstrained natural language and intelligently respond to anything that the user entered. We thus decided to make a system that used a combination of computer and human intelligence in its responses. This enhanced interactive fiction system incorporated a language parsing system and was programmed to know how to interpret a variety of commands and statements according to the file defining the specific world of the show. Then, if the system did not know how to handle a specific statement made by the online user, it would send that statement onto a human "operator," who could determine how to parse and respond to the statement, by giving the system commands that it knew how to interpret, and/or by writing responses live to the online participant. Thus, the text world experienced by the online participant was a mix of pre-composed text stored with the rules of the system and additional bits of content improvised by an operator to keep the experience flowing smoothly. The amount of interaction that an operator had with any particular online participant varied from participant to participant and throughout the course of the experience.

I served as one of the two primary operators for this system during our pilot run. The operator served as an improvisational actor, needing familiarity with the story content of the real and online experiences, the online world, and the actions available in the system. Being an operator was a fascinating experience, as I was both moderator and storyteller, both helping to guide people in learning how to uncover existing narrative in the world and improvisationally creating additional narrative in the world as necessary (for instance, scripting a conversation when a user chose to talk to the lawyer in the bar, a character who briefly appeared in a video clip).

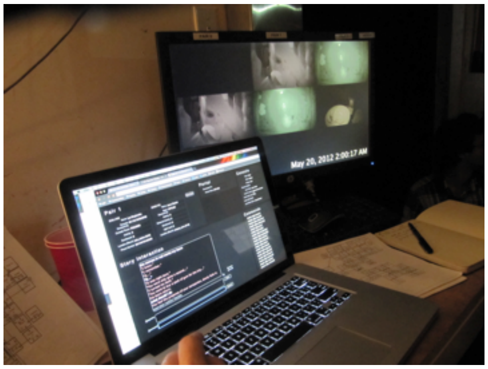


Figure 25. The *Sleep No More* Extension's operator interface
During the run of the experience, the operators could see what video was sent to each pair of participants. Through a web-based interface, they could interact with each individual pair through entering story commands, writing text to the online participant, and sending audio and visual content.

The overall feedback on this project was highly informative, though also mixed. While some online users found themselves deeply engaged by the experience (often more than they had anticipated), others did not feel sufficiently connected to their onsite partner or did not realize they had one. The latency for the user in receiving certain human-generated answers was found frustrating by some. Some onsite users were excited by the moments of special interaction and additional storylines, others were not sure they knew what was going on. While the connection between the partners was not found to be as strong and clear as we had hoped, the relationship of the online partner's experience and the human operator's improvisational storytelling through the system was an unexpected and exciting discovery.

This project is especially interesting as a model of allowing both digital and human systems to play to their strengths in telling a story. The digital systems could handle the overall story arc and world, as pre-programmed in JEML. They could instantly recall long text descriptions, keep track of a complex model of the online world's state, and display the state of the world to the online participant through a variety of media. These systems did most of the "heavy lifting" of creating and modifying the online space. This freed up the human operators to step in as necessary to parse complex natural language, to observe the engagement levels of the participants, and to help shape the story for an individual participant. A human operator could not have created such an experience in real time without any pre-existing content, and the digital systems could not have run the experience by themselves without having to set obvious boundaries on what the online participants could do. Through the combination of the two, a new kind of compelling story experience was created.

4.2.5. The Chandelier: Continuous Control and Triggers

The Chandelier from the opera *Death and the Powers* is another example where a combination of machine and human sensing proved to be both accurate and expressive. This Hyperinstrument also serves as an example of combining continuous control of parameters with discrete triggering of events. *Death and the Powers* features a scene where Simon Powers, having uploaded his consciousness into his environment, inhabits his large chandelier and in that form has a romantic and erotic encounter with his third wife, Evvy. For the first half of the show, this chandelier hangs above the stage serving only as a light fixture. In the duet between Simon and Evvy, it descends to the stage and closes around her. When she touches its strings, she finds that it is a musical instrument. As she caresses the strings, plucking and strumming and damping them, she controls layers of the sound in the scene.

The scenic design of the Chandelier was created by Alex McDowell and Steve Pliam (Pliam, 2007). Several prior Opera of the Future students had explored ways the strings of the Chandelier could mechanically generate sound, from bowing to robotically plucking to resonating with electromagnets at different frequencies. However, the mechanical actuation was deemed impractical for a moving object. We determined that the most important element of the Chandelier scene was how the character of Evvy could physically interact with it; we needed her movement to be clear and sensual, not constrained by the needs of playing a complex instrument. We thus turned the Chandelier into a giant controller, with the emphasis on how it was played rather than how it could physically create sound. My role was in designing the sensing and interaction for the instrument.

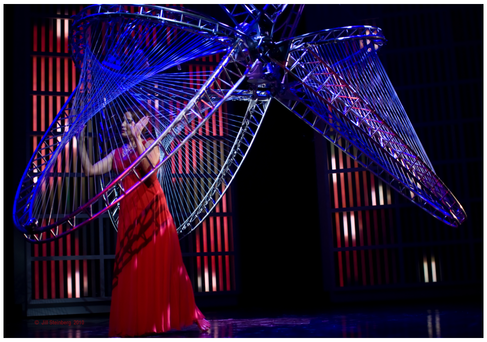


Figure 26. Evvy plays the Chandelier
Patricia Risley as Simon Powers' wife Evvy has a physical duet with Simon as the Chandelier, manipulating his voice and layers of sound by how she touches the strings. Photo by Jill Steinberg.

In the original version of the Chandelier used in the premiere performances in Monaco, stretch sensors around groups of strings were used to detect Evvy's interaction with the instrument. As she strummed the highly elastic Teflon strings, they vibrated and stretched the sensors. Evvy's activity interacting with the strings shaped the levels of a pre-composed musical track, the sound of the

Chandelier. In addition, we determined that we wanted Evvy's deliberate plucking of strings to trigger special samples. Initially, I attempted to detect these special plucking gestures from the pure stretch sensor data. However, I found that the sensor data was not sufficiently differentiated between plucks and other kinds of interactions as to reliably pick out the desired gesture infallibly and never have a false positive result. Additionally, recognition of a plucking gesture, and thus triggering the associated sound, only could occur after the gesture had been completed. The gesture was too short to predict given only the data about the string movement. For such a swift action, this was not a sufficient reaction time; it introduced a momentary latency. We needed a system that could not only reliably recognize a plucking gesture and distinguish it from other kinds of actions, but also one that could anticipate the plucking gesture so as to trigger the sound as the string was being plucked, rather than when it had been plucked. The solution we ended up using was to add a human into the sensing loop. The continuous dynamic movement was still detected by the computer system and used to shape the audio at each moment. However, the key "plucking" gestures were identified by an outside technician who could then push a key to trigger the appropriate samples in sync with the performer's gesture. Given a technician with sufficient experience anticipating and following improvisatory movement, the resulting interaction is much more accurate and more immediately linked with the performer's movement. This instrument thus illustrates another combination of human and machine sensing to take advantage of the skills of each. The nuanced detail of the playing was best captured by a technological system, but the recognition of a specific key gesture was best achieved by a human.

4.3. The Creation Process of a Piece

Another point of focus in designing technology for performance extension and for creating a performance that incorporates performance extension technologies is how those tools can integrate into existing processes for creating performances. Ideal systems and methodologies should fit into the initial ideation process for a piece, be useful throughout the rehearsal process, and integrate into the final performance or installation. The development process of a particular piece may proceed linearly through these stages, though frequently the sequence is more complex. Ideas about a piece are refined and changed through the rehearsal process. Initial performances of a piece may then lead back to modification of the ideas and more rehearsals.

4.3.1. Ideation

Often, the first stage of creating a performance or installation work is the ideation stage. How does a production first start? With an idea, with a concept that may be exceedingly specific or quite broad. Often, this idea or set of ideas is explored, extended, and refined even before there are any formal rehearsals.

Early processes of exploring performance ideas may take the form of gathering evocative inspirational material, from sounds to stories to images. In the ideation process of "image banking," performance creators gather a variety of images, guided not by any specific content of the images but by which images they find interesting in relationship to the piece in development. These collections of images can then be analyzed to see what they reveal about elements of the piece (color palettes, structures, moods, spatial relationships, similarities and variations, etc.).

Initial explorations for various kinds of performances or installations may include developing fragments of movement or gestural vocabularies, writing musical phrases, selecting or writing sections of text, designing sounds, testing interactions, sketching designs, storyboarding potential sequences of actions, shaping expressive arcs for sections of the piece, brainstorming structural frameworks, and envisioning particular moments of a piece.

All of these ideation processes create different varieties of "sketches" of ideas, partial realizations that can be analyzed and explored. Technologies for performance extension should lend themselves to such quick sketches in a similar manner. They should support rapid experimentation with ideas to see what ideas remain interesting and what ideas do not feel connected to the goals of the performance or installation. Can one create a quick mapping to test a particular idea about a performance, without having to take much time or put in much effort for an idea that is merely an initial exploration?

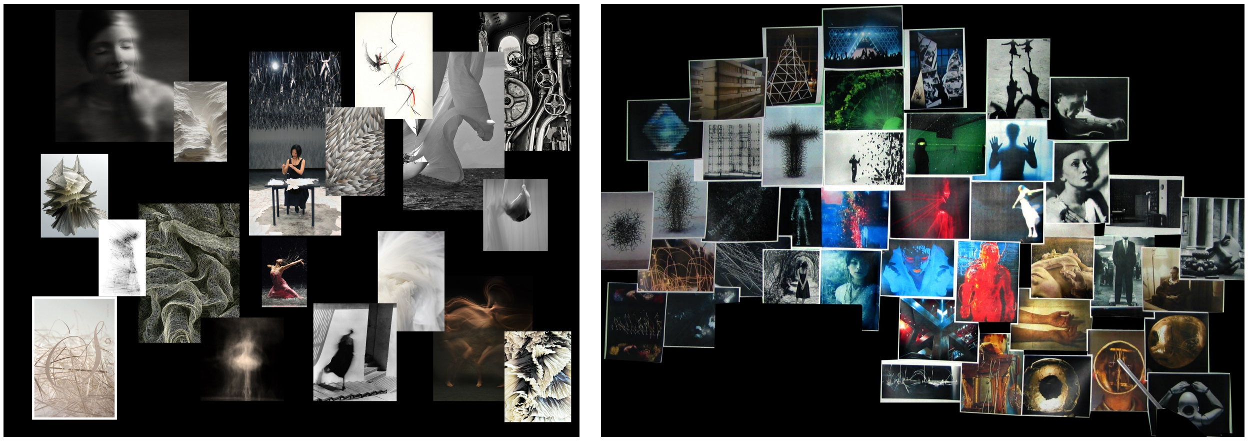


Figure 27. Example inspirational image banks
The image bank on the left was designed by the author in the development process of *Crenulations and Excursions*, discussed in Chapter 6. The image bank on the right was created by the author in collaboration with Peter Torpey and Alex McDowell for *Death and the Powers.*

4.3.2. Rehearsal

At the beginning of the rehearsal stage, a work can be in many different stages of development. For a major opera or theatrical production, the script or score likely exists at the beginning of rehearsal, as well as initial or well-developed scenic, lighting, and costume designs. The director will often have made many decisions about blocking and character development, and developed her overall vision of the production. For a musical production, a musician or conductor will generally begin rehearsal with a musical score. The details of expression and interpretation of that script or score come both from ideas that are brought into the rehearsal process and from ideas that are discovered in the course of the rehearsal process. For a dance production, some choreographers will come into a rehearsal process with particular movement sequences or vocabularies already created, while others will prefer to develop movement content directly on their dancers. Others come in with fewer preconceived ideas about the content, movement vocabulary, or story of a piece, aiming to discover those elements through a process of improvisation with the performers. Rehearsal processes for some experimental musical compositions may proceed similarly: the composer may bring in some ideas for musical content, particular musical phrases, or particular sonic exercises, and develop more material through a process of improvisation and exploration with the musicians and the ideas.

In all cases, the material of the performance will be developed, shaped, and continually refined throughout the rehearsal process. The performance-maker (director, composer, choreographer, installation designer, etc.) must remain open to observing what is actually going on in the space with the performers, and to flexibly adapting his ideas and direction as inspired by the performers. In *The Empty Space*, director Peter Brook relates the story of his first major directing role, as director of *Love's Labour's Lost* (Brook, 1996). Prior to the first rehearsal, he meticulously and painstakingly plotted out paths for the forty actors in a scene where the Court first enters. At the rehearsal, he gave each actor the directions for his or her first stage of the entrance, and instructed everyone to enter as directed. However, Brook immediately saw that the movements of this mass of people were very different from what he'd envisioned:

"As the actors began to move I knew it was no good. These were not remotely like my cardboard figures … we had only done the first stage of the movement, letter A on my chart, but already no one was rightly placed and movement B could not follow … Was I to start again drilling these actors so that they conformed to my notes? One inner voice prompted me to do so, but another pointed out that my pattern was much less interesting than this new pattern that was unfolding in front of me -- rich in energy, full of personal variations, shaped by individual enthusiasms and lazinesses, promising such different rhythms, opening so many unexpected possibilities … I stopped, and walked away from my book, in amongst the actors, and I have never looked at a written plan since" (Brook, 1996).

The rehearsal process for performance works presents particular challenges for incorporating technology seamlessly. There are a number of features of a rehearsal process that are relevant to analyze in this context:

- Creative ideas are tested, developed, and refined during rehearsal.
- Modifications to the show are made rapidly.
- Rehearsal time is precious.

As described above, the rehearsal period is generally the time when the majority of a piece is developed and created. The director or choreographer or conductor may come in with various levels of prior research and development and different amounts of direction already determined. However, during the live process of having the performers together in a space, the performance-creators must be flexible and open to the ideas that emerge during rehearsal, whether that requires slight modifications or complete transformations of previously determined ideas.

In rehearsal, a director or choreographer is used to being able to change many things instantaneously, to give directions about things to try and immediately see the results of those directions. A director should be able to say something like, "Let's start again from Ed's entrance, and this time let's see what happens if you enter from upstage left and come in much more quickly. Sarah, you need to take a little longer before you respond to him, and on your line cross downstage left." In a manner of seconds, the performers reset and begin the scene again with these modifications. For any performance extension technology to be easily integrated into a rehearsal process, it needs to be able to make desired modifications on a similar timescale with similar ease. It would be unacceptable for the director to make a similar request of the interactive technological systems and the technologists to respond, "Of course, as soon as we can stop all of the running systems and change that code, we can have that working for you. Perhaps by tomorrow's rehearsal?"

Of course, even standard technological elements in the theater, such as stage lighting, scenic elements, or audio systems may have several different timescales of response. Making two lights brighter may be a change that can be quickly accomplished before re-running a scene; adding a light in a new location and changing its color may require some time to accomplish (unless the production is equipped with a sufficient number of moving lights). Similarly, different kinds of alterations to an interactive system may require different amounts of complexity and time to accomplish. However, the swifter the modification process can be, the more thoroughly the interaction can be designed in the course of the rehearsal process and integrated into the production.

Additionally, there may or may not be additional rehearsal time specifically set aside for incorporating, tuning, and fine-tuning technological elements. It is highly beneficial for these activities to be done synchronously with other aspects of the rehearsal process. For example, changing the ranges of how much a particular performer's behavior affects a visualization should not require stopping and restarting that visualization, but should be adjustable mid-scene with a minimum of disruption to the rehearsal.

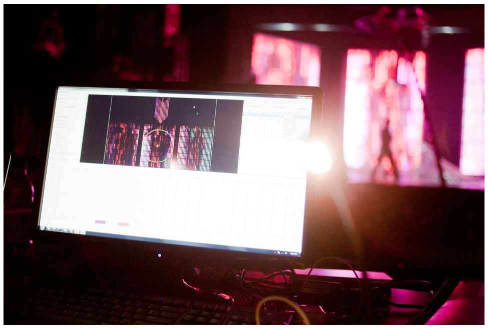


Figure 28. Rendering system for *Death and the Powers*
The visuals and performance mappings were shaped during the rehearsal process, using systems that could be modified and run simultaneously. Photo by Matt Chekowski.

The design of the systems for *Death and the Powers* was highly influenced by the demands of a traditional opera rehearsal process. We designed all systems, including the Disembodied Performance System, to be able to rapidly test and iterate on new ideas. We knew that the majority of our development would be done mid-rehearsal, and thus that our systems needed to have no distinction between a "composition mode" and a "performance mode." All of the opera control systems, from the robot choreography design toolkit to the Disembodied Performance mapping system to the visual rendering system, allowed changes to be made to mappings, visualizations, or robot choreography and have the results be immediately seen without the need for a compilation process.

4.3.3. Performance or Installation

Once a piece has been rehearsed and developed, it will eventually be performed (or, in the case of an interactive installation, opened to the public). There are various properties of a live performance or installation scenario that place particular challenges and demands on any performance extension system.

First, it is necessary that performance augmentation and extension systems run in real time or as close to real time as possible. The time period in which audiences will perceive two events happening "simultaneously" is very brief. For example, Levitin et al. found that the visual appearance of a person striking a drum and the audible sound of the drum needed to be within around 40ms of one another for observers not to detect the asynchronicity between the events (Levitin, MacLean, Mathews, & Chu, 2000). The majority of analysis algorithms must be performed "on line"; there are few performance scenarios where it would be feasible or desirable to capture some performance information, send it to an algorithm that takes a few seconds or a few minutes to do some processing, and then return the results.

Second, systems for a live performance or installation scenario must be consistently reliable. The nature of live performance is that the show begins on a particular day at a particular time. At that moment, all systems must be operational. They must stay fully and reliably operational throughout the entire show. Similarly, in an interactive installation, the experience must be working correctly whenever a new visitor enters the installation. This differs from movement or voice analysis systems that are generally intended for a live demonstration, where there is some flexibility in presentation if portions of the system are not working as intended. In a performance or installation, one does not have the opportunity of going back and trying something again if it did not work properly the first time.

Third, systems must be able to adapt to variation in performance. Humans are not robots; every night's performance will be unique and individual. Performance extension systems generally seek to take advantage of this moment-to-moment variation. However, it is also important to note that systems need to be flexible enough to account not only for intentional, expressive variation but also for mistakes in a human performance. What happens when a performer accidentally skips a portion of the choreography, or enters late, or sings the wrong phrase at the wrong time? What about when a performer momentarily forgets what is supposed to be happening? Are the systems for performance extension flexible enough to still work with this level of variation from the expected performance? For an example, let us imagine a standard score-following system that is playing an accompaniment along with a live singer. What is the behavior of the system if the singer misses a note, or jumps to the next phrase inaccurately? How much can the system compensate with behavior that falls in a range that is desired and expected? "The show must go on," and the technology within the show must go on as well.

Similarly, performance systems and the design of extended performances should be able to compensate for technical problems. How can a system fail gracefully when unanticipated problems arise? For example, what if the wireless wearable sensors on a performer run out of batteries and stop transmitting data? What is the default behavior of the system when it is no longer receiving the live performance data? This is both a system design question (are there default mappings that can be activated, or features allowing human operators to step in as necessary to adapt for an unexpected technical situation?) and a performance design question. What *should* be the default behavior of a system and the digital media extensions if they are not receiving live performance input, or if there is some other technical error? Should some reactivity be "faked" via a human, or is there some non-reactive mode the system should be placed in that will still look or sound interesting? Is there a way to possibly "play back" recorded data from another performance as the output? Does all of this information need to be controlled via the same system that does the interactive mappings, or is there an easy way to “turn off” that system and take over with another system designed for live technician input? For example, in *Death and the Powers*, when the performer is not providing performance input into the visual system, the majority of the visualizations are completely or almost completely static images. We had some external control over parameters of the visualizations via a TouchOSC application on an iPad, so if something went wrong with the data capture in a performance we could switch from live data from the mapping system to manually-manipulated visual control parameters.

A last principle to note about live performance or installations is that the lifespan of the experience may be quite varied. Some works may be presented only once, while others may run for months or years. Concepts of reliability and failing gracefully will also be relevant on these longer timescales. Other important aspects may be how easy a system is to debug or replace components, if something starts to fail, as well as how experienced of an operator is necessary to run the system.

4.4. The Expressive Performance Extension Framework for Machine Learning in Performance Extension Systems

The principles discussed so far for movement and vocal extension in performance and installation contexts serve as helpful guidelines when attempting to integrate machine learning techniques into these contexts. This section lays out a process for creating a work that uses machine learning techniques to perform expressive performance extension, and presents many of the key questions that are relevant both for the creation of a technologically-extended piece and for the creation of systems to support performance extension. The Expressive Performance Extension Framework presented in this section prioritizes the representation of movement and vocal expression as continuous trajectories through continuous expressive spaces defined by semantically-meaningful parameters, rather than as discrete categorization into semantic descriptions of expression. It seeks to locate the majority of the creative process in the mappings between high-level descriptions and output control. It gives the user control over these input-to-output mappings, while providing high-level descriptions of input as more intuitive handles for creating mappings. Additionally, it has an intrapersonal generalization goal rather than an interpersonal generalization goal: the aim is for the framework and systems to be useful for each individual artist in an individual way, rather than creating one fixed set of movement and voice qualities and attempting to apply that set to all creative scenarios.

Two key features of the Expressive Performance Extension Framework are its focus on expressive qualities of movement and the voice, and its use of abstract, high-level parameters for describing those qualities. I agree with Antonio Camurri (Ricci et al., 2000) that the majority of expressivity communicated through movement is conveyed by spatio-temporal characteristics of the movement, rather than by syntactic meanings of particular gestures. In particular contexts, it may be useful to perform classification or recognition of certain movements or vocal gestures either before, after, or alongside the analysis of continuous expressive parameters. This framework will allow for the integration of such classification steps; however, it particularly focuses on the analysis of movement along continuous parametric axes. Regardless of the particular parametric space defined for a particular artwork, the outputs of the system into any particular expressive space are continuous values, not classification; the primary goal is not to label a movement “staccato” or “angry,” but to find a position on a set of continuous expressive axes that are meaningful (in the particular context they are used) for controlling output media. I describe specific sets of expressive axes developed through my research as a useful starting point for movement or vocal analysis, but also incorporate methodologies for selecting a particular set of axes for a given piece.

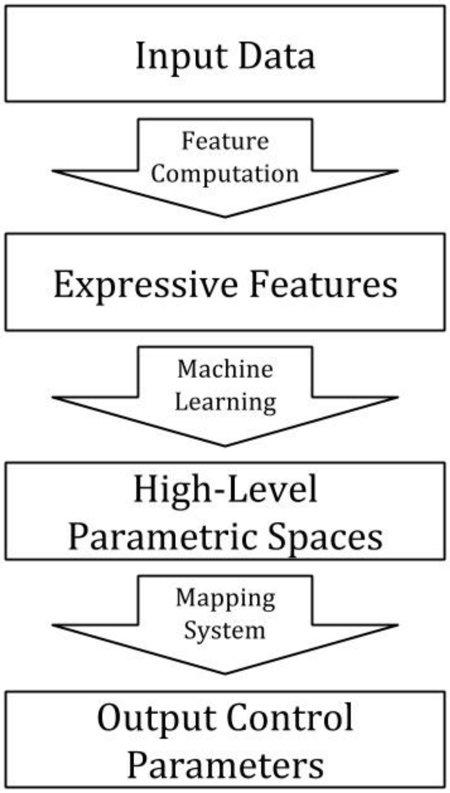


Figure 29. Four-step process for expressive control

This framework incorporates machine learning techniques for identifying a performer's location in these kinds of expressive spaces. While an association from sensor data to a position on or trajectory through a set of expressive axes could be created by hand, there are numerous advantages of machine learning techniques over hand-coding associations. The first of these is speed and flexibility in developing new connections between inputs and an expressive space, changing expressive spaces, or changing sensing systems. Rather than writing a new (and likely not reusable) piece of code for every set of input sensing systems and abstract parameter spaces, machine learning allows the same code structures, given new example data, to work for a variety of contexts. This allows users with less coding background to create these mappings from input data to intermediate expressive spaces. Perhaps even more importantly, using machine learning techniques means that a user does not have to know the exact relationship between the data and the desired expressive space, and does not have to manually determine that relationship through a laborious empirical process. In many expression analysis tasks, it may not be entirely clear which features of the input data relate to the desired expressive space.

In the Expressive Performance Extension Framework, the process of working with machine learning of abstract parameters can be conceptualized as four layers of information: input data, expressive features, high-level parametric spaces, and output control parameters. Input data is transformed into expressive features via feature computation, expressive features are transformed into points in high-level parametric spaces via machine learning, and information about high-level parametric spaces is transformed to output control parameters via manual mapping processes.

The actual process is actually much more complex than this, because this sequence of actions can happen at any point in the lifecycle of a piece. The interactions and associations between each of these four layers are developed throughout the design and rehearsal process of a performance or installation, and can potentially be shaped by later performance input as well.

The remainder of this section outlines a multistep process for creating a performance piece that integrates machine learning techniques for recognition of abstract expressive parameters. The steps of this process include: developing the expressive goals of the performance or installation; choosing a set of expressive qualities that seem relevant to a particular performance work; selecting methods of sensing for data capture; computing potentially meaningful features from sensor data; collecting training data examples; selecting, tuning, training, and testing an algorithm for machine learning; and mapping the analyzed movement parameters to output control. These steps are not necessarily performed linearly, but are generally explored and refined somewhat simultaneously. Developments in one stage may reveal modifications to be made at another stage: for example, if the qualities learned by pattern recognition algorithms are not sufficiently subtle, a sensor that collects more nuanced information may need to be added to the process.

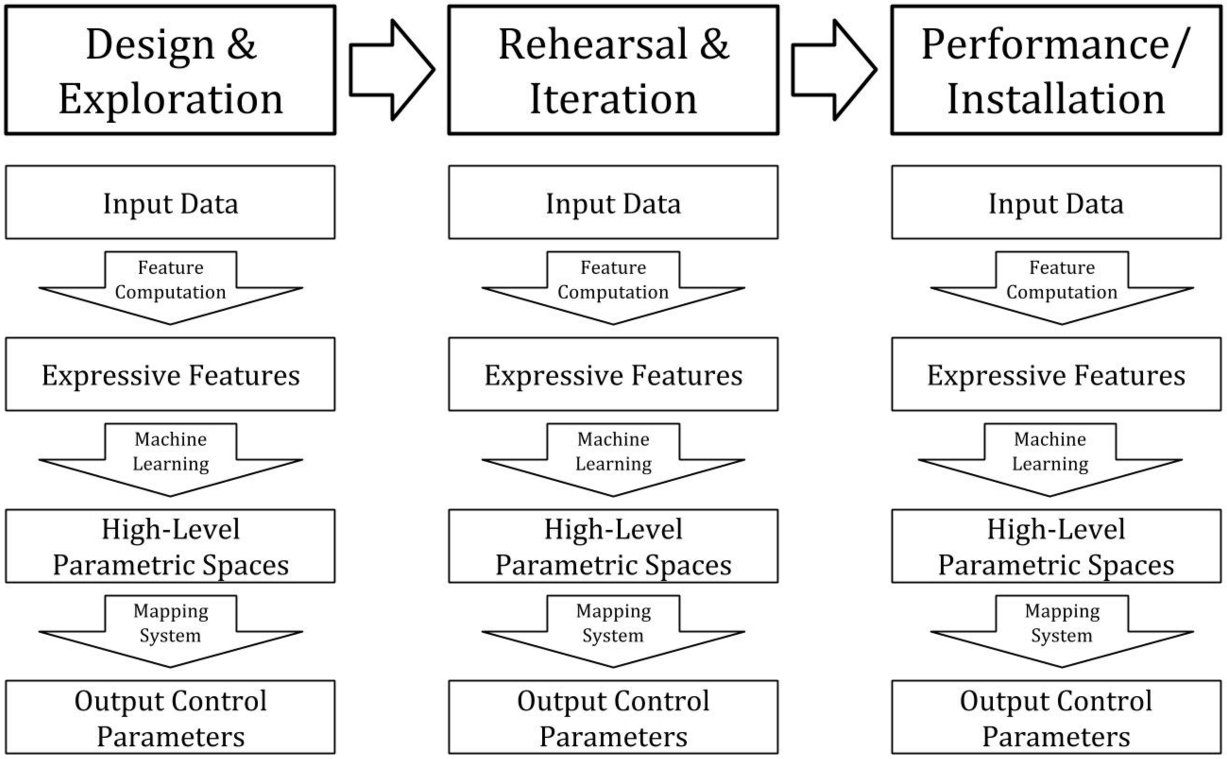


Figure 30. More complex model of expressive control
The four-layer model presented in Figure 29 is relevant at many different stages of the development of a piece.

Step 1: Determine Goals of the Performance or Installation

The first stage in any integration of technology into a performance or installation should not have anything to do with the technology. Before deciding what the technology is or how it should work, it is most important to find some creative goals and core ideas for the work. These ideas will help shape everything that comes afterwards. What is the heart of the project?

In our work in the Opera of the Future group, the heart of a project is almost never a specific technology or use of technology, but an imagined experience that we then try to find the technology to create. For example, the early development of *Death and the Powers* began with the idea of a "choreography of objects," the question of how the movement of objects onstage could help tell a musical story in a new way. The show was to be an opera, defined as a music-driven story. As initial ideation and development of the piece progressed, the creative team developed the story of Simon

Powers and his design of the System as a way to live on in the world after his physical body dies. So, we had this story about a man who becomes his house, which we interpreted to mean the theatrical environment. The technical decisions that came after that tried to be in service of this story. What does it mean for a set or an entire theater to be expressive like a human? Could we extend a live singer's performance to somehow shape the behavior of the whole environment? The Disembodied Performance System was developed in response to these questions and story goals.

Another example comes from the development of the Vocal Augmentation and Manipulation Prosthesis, discussed in Chapter 3. This interactive controller for a singer arose from two core ideas: an opera character who was supposed to have a prosthetic arm that made him specially enabled, and the image of "grabbing" a sung note with the pinch of two fingers. These core ideas guided the technological choices and interactive implementation of the glove. How could the instrument create a sense of wonder about the performer's extended capabilities? What kind of a movement vocabulary would be clear in the context of an opera production? What kinds of sensing and gesture recognition technologies would be appropriate for detecting relevant aspects of that movement vocabulary? How should the sensor information be mapped to sonic control?

A story discussing the opposite kind of creative process comes from a dance and technology user group mailing list in 2001. An announcement was posted about a new performance in which dancers would be located in two different cities, each wearing motion capture suits. The data from their performance would be used to control virtual avatars located in a virtual shared performance space. A response to the post summarized a major potential issue of technology-augmented performances: "I'd rather hear about the artistic content and motivation for using the technology, not just the technology itself. What is the content, exactly?" (Dixon, 2007, p. 6). It is not enough for a performance work to be "about" the use of a particular technology; the technology should be a tool to create a new kind of experience.

Different types of questions may be applicable while developing the core creative ideas and structure of a performance or installation:

- What is the story to be told by this work?
- What is the goal of creating this work?
- What ideas are explored and conveyed with this piece?
- What is the desired experience of the audience or of a participant?
- Are there particular images or moments that are envisioned to occur in the piece?
- What is the narrative or expressive arc of the piece?
- Is there a larger thematic or experiential context for the performance or installation? (What is the space or spaces where it is located? Is it part of a series or presented with other experiences?)
- Who is the expected audience? What is their skill level, in the case of an installation?
- How long is the experience? How might that length of experience vary?

While not all of these questions will be relevant for any individual piece, they are examples of broader conceptual and structural elements that have nothing to do with specific technologies and yet will be key guides for the choice and use of specific technologies.

Step 2: Select Desired Expressive Qualities

Regardless of how movement or voice will be sensed or what pattern recognition process will be implemented, it is necessary to define the expressive space that a system is to learn. A major goal of this research is the transformation of sensor data into meaningful high-level parametric structures incorporating multiple expressive axes, each with a normalized range. The sensor input, ranges of parameters, and desired expressive spaces will vary between different pieces and different performance-makers. Thus, an ideal system should let a user demonstrate examples of movement or voice that form particular points in an expressive space, and automatically model the relationships between input data parameters and that continuous expressive space. A user may also want to define their own high-level axes, in addition to suggested parametric structures.

For different performance-makers, the kinds of aspects of movement that are expressive may be very different, as may the definition of "expression." Modern dance choreographers Martha Graham and Merce Cunningham are examples of choreographers who saw physical expression in very different ways. Graham's technique is about expressing emotion, with definitive gestures corresponding to particular emotional states. Her guiding choreographic principle of contraction and release is designed to show a direct correspondence between body movement and human emotion. For Cunningham, expression comes from the fact that the human body is in motion, but his motion is not designed to express something specific. Cunningham's choreography incorporates aleatoric techniques, attempting to keep a focus on pure movement. Graham's gestures have specific narrative and semantic meanings, while Cunningham's gestures focus on the expression intrinsic to rhythm and movement. Different sets of axes for defining movement qualities might be more or less useful in each of these cases. For a Cunningham piece, it is not so relevant to look at affective parameters describing movement, while for a work by Graham, it would likely be very important to do so.

In determining the quality axes to use for analysis of specific performance input, it is important to consider the relevance of temporal, spatial, dynamic, and emotional parameters in describing that input. As an example, let us say that the input is sensor data from a solo ballet dancer, performing choreography that is inspired by the physical properties of fluids in motion. As this is a non-narrative piece without gestures intending to represent specific emotions, a set of expressive axes to describe this movement might include parameters like *pace* (slow to quick), *fluidity* (legato to staccato), *scale* (small to large), *continuity* (continuous to disjointed), *intensity* (gentle to intense), and *complexity* (simple to complex). These orthogonal axes can be combined to form a high-level expressive space. Positions in and trajectories through that space can then be mapped to control parameters for multimedia. In all cases, the outputs of the pattern recognition algorithms should be continuous values, not classification. To extend the possibilities of expression recognition systems even further, systems can analyze temporal behavior not only to determine the performer's current point in an expressive space, but also to recognize particular trajectories or features of trajectories through expressive spaces.

What kinds of parameters may be most useful in describing the movement or vocal qualities to be used in a piece? I have suggested a set of qualitative parameters that are relevant for describing both movement and vocal quality (*intensity*, *scale*, *energy*, *complexity*, *fluidity*, and *rate*), but there are many other kinds of parameters and models that may be relevant for a particular piece or a particular performance-maker. Given our analysis of different frameworks of movement and voice in Chapter 2, it seems likely that most sets of expressive parameters will at least need some measurements of energy and some measurements of how the input relates to time.

In this framework, I suggest that sets of expressive parameters should be selected to be reasonably orthogonal. This means that the parameters do not depend on one another, that changing the value of one parameter does not necessarily affect the value of another parameter. For example, take *rate* and *fluidity*. It is possible to perform fluid or jerky motions at many different rates. These parameters are likely conceptually orthogonal. If one examines concepts of *rate* and *tempo*, however, it is likely that these parameters represent slightly different features of the metric of "speed"; increasing the tempo of movement will likely affect the rate and vice versa.

While parameters are semantically and conceptually orthogonal, they are likely in practice not completely independent. Values of Parameter A do not depend on values of Parameter B at a given moment; however, the current value of Parameter B may influence future values of Parameter A. Changing one parameter may not directly impact the value of another parameter, but may affect the ease with which that other parameter can be changed in the future. For example, if one is moving with a high amount of energy, it may become more difficult to move slowly while maintaining the amount of energy in the movement. It is still possible to move slowly and energetically, but it is more challenging than moving quickly and energetically. This parameter space can perhaps be visualized as a set of springs pulling on a moving point, where each spring is an expressive parameter and the point is the user's current expressive state. As the location of the expressive state changes in relationship to the spring endpoints, moving in one direction will make it easier or harder to move in other directions based on the relative strengths of the springs.

Step 3: Select Sensors

As one develops a list of expressive parameters about movement and voice, one can begin to select sets of sensors to collect data from the live performance. Different types of sensors and sensing strategies will be more or less useful depending on the desired information to be gathered from the input and the limitations of the particular sensing setup.

Benford's framework of "expected," "sensed," and "desired" actions is an interesting model for exploring sensor-based interactions, especially for applications with more creative or exploratory purposes, rather than applications designed for the user to perform a specific, known set of tasks (Benford et al., 2005). In this framework, expected actions are those which the user is likely to do or might be expected to do. Sensed actions are those that the given sensor setup can properly detect. Desired actions are those that the interface designers want users to perform to control the interface. Branford points out that the combinations of these actions (for example, actions that are desired but cannot be sensed) provide either limitations to be overcome or opportunities for designers to add additional creative functionality. These distinctions can be useful in examination of the capabilities of different sensing systems.

Several different popular performance sensing strategies are presented here with some of their strengths and weaknesses. Additionally, features of two of these sensor strategies, wearable movement sensors and computer vision systems, are discussed in more depth to demonstrate examples of some of the aspects of a performance or installation to be considered when selecting sensors.

Using frames from a video camera as sensor input is a popular technique in gesture recognition systems (e.g. Avilés-Arriaga & Sucar, 2002) and has often been used for using movement to control music (e.g. Modler et al., 2003) or in interactive dance performances (e.g. Ricci et al., 2000). These systems require no technology to be worn by the performers, are unobtrusive, and can be relatively inexpensive (depending on the cameras used).

Cameras that cover the entire stage space can also be useful for obtaining movement information about all of the performers onstage, though it is more difficult to locate specific individual dancers. It is also challenging to follow movement when dancers are temporarily occluded from the camera's view. Similarly, a vision-based system will have difficulty tracking individual dancers through a space if the paths of two dancers cross, unless there are constraints on the costume design (such as distinct colors) that make individual dancers be quite visually distinguishable.

One major issue with video input is that these recognition systems are generally not robust under varying stage lighting or with different backgrounds. As (Wilson & Bobick, 2000) state, "Lighting conditions, camera placement, assumptions about skin color, even the clothing worn by the user can disrupt gesture recognition processes when they are changed in ways not seen during training." While the stage setup can be controlled to attempt to provide optimal separation between dancer and background, this may result in limitations on the stage, costume, and lighting design for the performance, or complexities incorporating the machine learning algorithms into the rehearsal process. If the conditions under which this piece will be rehearsed and under which the expression recognition systems will be trained are not identical to performance conditions (different outfits worn by dancers, a different space, different lighting conditions), this poses a significant challenge to vision-based recognition systems. The only meaningful input features that can be learned may be changes of parameters, rather than particular parameter values.

A different strategy for movement sensing is to detect features of the dancers' movement through wearable sensors, such as Inertial Measurement Units (IMUs) such as accelerometers and gyroscopes, located on several points of the body. Data from those sensors can then be transmitted wirelessly to an expression recognition system for processing. Benbasat describes many of the advantages of working with IMUs over vision-based systems (Benbasat, 2000), including that IMUs need less processing power than camera systems and do not suffer from issues such as occlusions. These wearable sensors also allow for spatial invariance in movement capture, which is an important aspect in movement and movement quality recognition (Nam & Wohn, 1996). A sudden, heavy movement, or a particular gesture, ought to be detected the same no matter where the performer is located on the stage or what direction a performer is facing.

One difficulty with wearable sensors is that they send continuous data whether or not a performer is onstage. For the sections of the piece that consist of a limited number of performers, we may need additional control data to the expression recognition algorithms to say which sensor inputs ought to be used and which ignored at any given time. Additionally, sensors such as accelerometers and gyroscopes do not easily allow us to have a sense of an individual dancer's location on the stage. If that sort of data is needed to be able to position any part of the interactive visual design, there will need to be additional or different systems in place to obtain that kind of information.

In an extended performance context, an interesting distinction can be made between sensors that capture aspects of a performance that are under the conscious control of a performer and those that capture aspects of the performance that are unconscious or outside the control of a performer. The latter category includes the majority of psychophysiological sensors, such as galvanic skin response sensors, heart rate sensors, and brain wave sensors. Measurements such as galvanic skin response rarely reflect the emotional experience that a performer is trying to convey, but are more likely to be affected by the performer's internal experience (stress about a challenging musical passage is coming up or a mistake, for example) (Nakra, 2000). In *Death and the Powers*, this was a primary reason why we avoided psychophysiological sensing. We wanted to capture and transform the experience of the character, the expressive behaviors that our actor was deliberately using to convey the character's personality, emotions, and experience. We did not want to reproduce the emotions of the actor as distinct from the emotions of the character. Our use of breath sensing falls somewhat between these two: in opera, the arc of the singer's breath while he is singing is a controlled and deliberate aspect of his expressive performance. However, when the singer is at rest, his breath may relate to the character's emotional expression, but more likely is shaped by his own physical needs.

In certain performance contexts, shaping multimedia through aspects that are not under a performer's control may be desirable. There is certainly a long history of performance augmented through measuring physiological or psychophysiological metrics of the performers. However, in the majority of extended performance work, performance-makers want a performer to be able to deliberately control the digital media through his conscious behavior, or at least to have the digital media reacting to aspects of the performer's behavior that are under his conscious control (even if that conscious control is directed toward the goal of producing a musical phrase or a movement phrase in a particular way, rather than toward a particular shaping of the digital media). In an interactive installation, having the behavior of the installation remain at least semi-controllable by a participant's deliberate actions will add to a sensation of "liveness." If an environment is measuring my brain waves and responding accordingly, am I aware of my own brain waves? Can I affect my own brain waves enough to experience the interactive nature of the experience?

A few common sensing strategies for movement and the voice are summarized in Table 1 and Table 2 with some of their advantages and disadvantages.

Type of Sensing	Advantages	Disadvantages
Handheld/Stand-mounted microphone	-If highly directional microphone is used, less susceptible to noise from distant sources -Visually signals that the vocal signal is being captured -With multiple microphones, allows separation in sensing from different	-If handheld, limits use of participant's arms -If stand-mounted, requires participant to stay in one location

	sources	
Wearable wireless microphone	-Allows flexibility of movement throughout a space -Comfortable and keeps hands free -Less susceptible to noise, as positioned to capture the desired vocal signal -With multiple microphones, allows separation in sensing from different sources	-Requires participants to be fitted with a microphone -If participants will be moving vigorously, can be constraining and will pick up sounds of movement -May pick up interference from other wireless devices nearby such as radios
Microphones in the space	-Can be located discreetly, participants need not know microphones are in place -Does not require participants to wear any devices -Can cover a wide space with sensing -Keeps a participant's hands free -Can capture information from many participants at once	-Susceptible to noise, will pick up all sound in the space in addition to the desired vocal signal -Signal quality may vary depending on participant's proximity to a microphone -Difficult to distinguish between different sources, if there are multiple participants

Table 1. Selected techniques for sensing the voice

Type of Sensing	Advantages	Disadvantages
Cameras/Computer Vision	-Off-the-body sensing, so does not require outfitting participants with sensors -Potentially low-cost -Unobtrusive -Can cover a wide area for sensing	-Susceptible to changes in lighting -Susceptible to occlusion -Challenging to distinguish different participants -Data varies with a participant's position in relationship to the camera (distance from camera, angle in relation to camera, etc.) -Can require high processing power to do tasks such as figure-tracking
Kinect	-Off-the-body sensing, nothing to wear -Built-in skeleton and hand capture -Separation of body from environment -Separation of multiple participants -Knowledge of participant's distance from camera allows calibration for data variation with position	-Limited field of accurate sensing for an individual Kinect -Sensing distorted beyond ideal range -Primarily gross movement captured (hand not fingers) -Susceptible to occlusion, changes in lighting, and alternate angles
Commercial movement capture systems	-Many points on the body precisely tracked	-Require markers to be worn on the participant -Large amounts of data may be

		challenging to process
Wearable sensors (accelerometers, flex sensors, gyroscopes, etc.)	-Ability to track fine details of movement and small movements (depending on location of sensors) -Resistant to changes in lighting and background, occlusion of performers, and changes in directionality -Can easily associate data with an individual even with many performers -Can require less processing power than vision-based systems	-Require participants to wear specialized garments or accessories (cannot simply walk into a space and be recognized) -Cannot generally track position -Send data continuously -Potentially an obtrusive or limiting design element in costumes
Electric Field/Capacitive Sensing	-No need for wearable items -Cheap hardware -Fast and requires low processing power	-Significant variation in data for each person -Highly variable with user's position in relation to sensor -Ambiguous data: can't be sure what is generating capacitance -Exponential decrease of resolution with distance -Data highly susceptible to influence from other people and conductive items in the space
Interaction with Sensor-Filled Objects	-Allows the use of many sensors without requiring preparation steps for participant (as with wearables)	-Requires the use of an object, which may or may not be appropriate for a given performance or installation context

Table 2. Selected techniques for sensing movement

In deciding on what sensing strategies to use, it may also be useful to consider some properties of the input information that the sensors are desired to capture, as well as the general context of the interactivity. Table 3 outlines some of these properties.

Relevant property of input or interactive context	General questions	Specific examples
Scale of input to be captured	What are the maximum and minimum values of the input? What is the softest/smallest input that should be picked up? What is the largest/loudest input that should be captured?	Can the microphone pick up the softest sound that will be sung? Does the movement sensing system cover the entire space in which sensed movement should occur?

Precision/sensitivity desired from input	How precise does the system need to be? What kinds of input need to be distinguished from one another?	Do we want to capture tiny hand movements, and/or the broad movement of a person within a large space?
Invisibility of sensing mechanism	How aware should the audience/participants be of the sensing? Can there be any preparatory steps to make someone able to be sensed, or not?	Is there a place to hide a camera or microphone? Are wearable items (sensors, identification markers or special colors) appropriate for the performance or installation context, or problematic?

Table 3. Relevant properties of input for selecting sensing strategies

Another important aspect in sensor selection (as well as sensor data processing) is the role of silence, stillness, or “not captured” movement or voice. Is it desired that every action that the user takes within a room is captured and processed, or should there be some region of interest within which the system is active, and other regions where a user can rest?

Multiple sensing strategies can also be combined for greater sensing accuracy or variable sensing modalities in different areas of a space. For example, technologies such as RFID sensing can be used for sensing the presence or absence of a person wearing an RFID tag, but not fine-grained information about the person's movement. In an extended performance, RFID technology may not be particularly informative on its own. However, this technology may be used meaningfully in combination with other sensing. For example, what if a Kinect is detecting elements of a performer’s movement, but we want the digital extension of that movement to vary depending on which performer is standing in front of the sensor? An RFID tagging system could identify different performers so that the Kinect data could be more meaningfully interpreted.

Ideally, systems for working with expressive performance extension will be sufficiently flexible as to allow many of these different types of sensing strategies to be used when relevant for an individual performance or installation. At the beginning of developing a work the creators may have some knowledge of which sensing strategies will be useful or unhelpful; however, the needs of the sensing may change throughout the development of a piece. For example, a particular interactive installation might start with the design of having interaction occur in a small region of the space, suggesting a Kinect as a possible sensing strategy. But what if the performance-makers then decide that, while fine-grained movement sensing is desired in that particular region, some sense of movement throughout the remainder of the room should also be incorporated into the experience? The Kinect may not be sufficient to capture this information and additional sensors (perhaps cameras) will have to be included. Ideal performance extension systems should be flexible enough to easily work with many different types of sensing, to combine sensing strategies, to allow the user to experiment with different kinds of sensing, and to switch sensing strategies at any point in the development of the piece.

Step 4: Select Features for Computation

The next step is to transform raw sensor data into features of interest. The process of *feature computation* is intended to turn raw data into something that is more likely to convey salient information or that is normalized for purposes of easier comparison. For example, if one is using computer vision to recognize a hand position, it might be useful to preprocess the image given to the algorithm to make the hand always the same size. It also might be helpful to give the algorithm a slightly blurred image so that individual pixel differences are less important. The feature computation and analysis stage also allows systems the use of not only immediate sensor values but perhaps information about smoothed average values over time, recent ranges of values, or rates of change of values. In this framework, the specific features of interest to be calculated can be defined for a particular sensor set and performance context, or predetermined general sets of features created by system designers can be used.

The selection of particular features is especially important in movement and vocal quality recognition, as some kinds of features may be more likely to contain expressive information. Meaningful features for both vocal and physical performance likely include the amount of energy in the input, tempo and changes in tempo, and the amount of variation in different input parameters. Temporal features of the movement and voice are also vital.

Sometimes a user may not know what features are likely to capture the majority of the expressive information in a given dataset. Fortunately, algorithms are available for automatic discovery of important features from a dataset, using the process of *feature extraction*. For example, in Principal Component Analysis, a system is trained with a variety of labeled data, where the data is some vector of input features and the label is a discrete categorization or continuous parameter value. PCA seeks to find an function that will map a high-dimensional input space (the vector of features provided as input to the function) to a lower-dimensional space that describes as much of the variation in that input as possible.

Step 5: Collect Training Data

When deciding what kinds of movement or vocal training data to collect for a machine learning algorithm, there are various components that must be considered, particularly in the case of systems designed for performance and installation contexts.

Important questions to consider include:

- What are examples of the extremes of each parametric axis?
- How can measurements of one axis be performed with a variety of values of the other axes?
- What are some intermediate points on each parametric axis?
- What complexities are introduced by a particular sensing setup?
- How much variability is expected in the input?
- How many different people will the system need to analyze?
- How can the data collection process avoid gathering incorrect data caused by the act of data collection?

It may be useful to plan in advance a space of movement or vocal qualities to capture initially, in order to make sure the parametric space is well covered and that the data captured is more likely to well represent the desired information.

A performance-maker should consider the boundaries of each parametric axis and some variations on that boundary. For example, when capturing data to explore the parameter of *scale*, one should record examples of the largest movement expected and the smallest movement expected. In addition to that, one should record examples of “large” and “small” movements that vary in other parameters and body parts. Perhaps the movements in the piece that feel the “largest” include: a broad circular sweep of the arms, a fast run across the stage, and a quick fall. At what different speeds can these movements be performed and still feel similarly “large”? All of these variations will provide different kinds of sensor data (depending on what kinds of sensors are used to capture the data), so may all be useful to capture as examples of “large” movements. The broader the variety of examples of a given quality provided to the system, the more likely that the system will not accidentally learn an overly constrained concept. For example, if all the training examples for “large” movements consist of sweeping arm movements, a “large” movement using only the legs may not be properly identified as such.

As another example, let us assume we are attempting to train a system to identify the complexity of vocal phrases for a certain performance piece. We may want to think about the effects on our perceived *complexity* parameter of different patterns of amplitude (crescendos/decrescendos, soft, loud, moderate, steady, widely variable), different vowels/timbres (as well as the amount of variation in timbre), different melodic patterns, and different pitch ranges. For the particular piece we are creating, we decide that we want *complexity* to primarily express something about timbral complexity and rapid shifting of timbres. We should be careful to capture training examples in this space on several different pitches, so that our system does not accidentally learn patterns that expect a particular pitch and only that pitch. Alternately, let us say we are most interested in the complexity created by a range of variations in pitch, the melodic “shape” of the sample. We may want to incorporate a few different vocal timbres and amplitudes in our training data, so that maximum complexity is not learned as “these kinds of rapid pitch variations, as sung quietly on an ‘oo’ vowel.”

Training data often should also be captured of intermediate qualities, examples of movement or voice with qualities that are not at the extremes of each parametric axis. If only examples of the extremes of a parameter are given, a system may be able to do some interpolation between those extremes, but may primarily be performing recognition of the quality’s extremes. As with the movements or vocal examples selected for the extreme points, it will be beneficial to capture variations that seem to be near the same place on the parametric axis. What are some different kinds of movements that seem midway between the smallest and the largest movements in the piece?

It is also important to take one’s sensing setup into consideration in creating a varied pool of training data. Is it possible for similar movements to be sensed in different ways, depending on their spatial relationship to the sensing system? For example, let us say that part of the system is a webcam tracking activity in a space. A large sweeping hand movement performed very close to the webcam will provide different data than the same movement performed far away from the webcam. However, it is most likely that those two performances of the movement should both be identified as movements near the large end of the parametric axis. If all the training data is captured from performances close to the webcam, the system's analysis accuracy will decrease when the performer moves far away. Better training data would include examples at a variety of distances and physical orientations to the webcam.

Another important question in deciding what training data to capture is how much variability is expected in the input that the trained system will receive. To reframe this in an expressive context, how different is the performance likely to be every night? Is this system being used to recognize expressive qualities in a piece that has fixed choreography, in a piece that will contain improvisational elements, or in a public installation where certain kinds of movement or vocal gestures may be suggested but the input is not constrained? The amount of variation necessary to design into the training data will differ across these scenarios.

As another factor of input variability, how many different people will the system need to analyze? If this system is designed for a solo dancer, it will be desirable for the system to learn qualitative parameters as they are expressed by that dancer in particular; however, for an installation where the system will interact with many different people, care must be taken to make sure that the qualities captured are sufficiently generalizable. For example, in the case of a public installation that analyzes the visitor's voice, capturing training data only from a single voice might have limitations: any individual voice will have a particular range and a particular set of innate qualitative features. Either training data should be captured from multiple individuals or, at least, the system should be tested to make sure that it has not over-fit the data to a particular subject.

Another key factor in training data collection is the length of the data examples collected and the manner in which the examples are segmented. In the Expressive Performance Extension System, discussed in the following chapter, data samples can be collected at any length, and are then normalized into a window size selected by the user. Individual samples are gathered by starting and stopping the system.

It is important not to introduce unintended information into the system through the action of signaling the beginning and end of a data collection window. For example, say you are collecting samples of a performer singing with varying levels of complexity into a microphone, clicking a mouse to mark the boundaries of each sample. That data collection process will likely not affect the training data. However, say you are attempting to collect training samples of yourself moving at different rates. If you have to click a mouse at the beginning and end of each sample, you will be introducing an unintended movement (clicking the mouse) into either end of the sample.

Step 6: Train Model

There are a variety of existing pattern recognition algorithms that may be useful both for learning relationships between particular sensor data streams and desired expressive features and for learning the relationships between these selected features of interest and the desired expressive spaces. It is important to consider several properties when selecting an algorithm to recognize expressive input in live performance contexts. First, how well does the algorithm generalize from its training data? Can it handle the complexities of human movement and voice, given representative examples? Second, how quick is the algorithm to train, and how many examples does it need for training? Third, can it run its recognition process in real time? While training the algorithm can be done with a number of movement examples "offline," the system has to be able to recognize and handle new input "online" in the middle of a live show. Fourth, can it handle temporal variability in its input, either by using algorithms that include a history of samples or by pre-processing input samples to obtain a time-normalized input? Finally, does it work with labeled data (necessary for developing desired expressive spaces), or can it handle unlabeled data (potentially useful for automatic feature extraction)? Can it be trained on a set of labeled data but improve given additional unlabeled or weakly labeled data?

A Hidden Markov Model (HMM) is a probabilistic algorithm popularly used for recognition of particular sequential patterns in time-varying domains such as speech (e.g. Rabiner, 1989) and gesture (e.g. Gillian et al., 2011; Ko et al., 2003; Nam & Wohn, 1996; Westeyn et al., 2003). An HMM is a state machine that models the statistical probability of given sequences of outcomes. It is described by a set of states, a set of transition probabilities between states, and a set of probabilities of outputting a particular observation while in each state. While a standard Markov Model represents a situation where the underlying probabilities are known, such as the probability of observing a particular sequence of heads and tails when flipping a coin, in a Hidden Markov Model the relationship between the underlying states and the observed outcomes is not known.

There are three major questions that can be asked about HMMs (Rabiner, 1989), as well as about most other pattern recognition algorithms:
1) Given a sequence of observations, how likely is it that a given model would produce that sequence? (This is the testing phase of a model.)
2) Given a sequence of observations, what is the sequence of states that best "explains" that sequence of observations?
3) How do we adjust the parameters of a model to best represent the probability of a given observation being produced by that model? (This is the training phase for a model.)

A particular advantage of HMMs in performance contexts is that they can easily handle temporal inputs and inputs of varying lengths, due to their built-in notion of sequences and memory (Rabiner, 1989). However, Hidden Markov Models are generally only used for classification. In order to have HMMs learn the relationships between movement and a continuous parameter space, it would be necessary to adapt these models to produce continuous output. Additionally, the inputs to an HMM have to be discrete integer values, and thus continuous (or essentially continuous) sensor input parameters have to be mapped into a space of discrete input classes. A large number of training examples is also required and it can a long time to train a model (Gillian, 2011). These issues, particularly the large number of training examples needed, may be significant drawbacks in a fast-paced rehearsal process.

Another pattern recognition algorithm frequently used in gesture recognition is the Support Vector Machine (SVM) (Abe, 2005). Some benefits of using SVMs for expression recognition include their strength at generalization and ability to handle small training sets. However, the algorithm generally performs classification rather than regression, and is more complex if more than two output classes are necessary. Additionally, the input to an SVM must be a fixed-length feature vector, so movement data streams of various lengths must be normalized to a fixed period of time. Standard SVM algorithms are generally only used with labeled data, though variations on the algorithm can be used to combine labeled input examples with additional unlabeled input examples.

A third algorithm that may be particularly beneficial for recognizing expressive parameters is the Neural Network (NN). NNs, inspired by models of brain activity, consist of interlinked layers of nodes (or neurons), each of which activates if the sum of its inputs passes a given threshold. Each link between nodes has a weight, and each node has its own activation threshold. When values are given to the input notes, activation thus propagates along the network. In the process of training a neural network, the weights and activation thresholds are adjusted until the network produces appropriate outputs for its example input vectors. While neural networks may take a long time to train, they quickly process new input data for testing and need few training examples (Bishop, 2006). Neural networks can be trained with either unlabeled or labeled data, thus proving flexible for a variety of situations.

Neural networks have been used frequently for the recognition of musical parameters (e.g. Fiebrink, 2011; M. Lee, Freed, & Wessel, 1992), where their ability to perform regression is a major benefit. Frequently, one neural network is trained for each desired output parameter (Fiebrink, 2011). Additionally, the same or separate neural networks can be used for classification tasks, should the choreographer choose to recognize particular movements as well as overall movement qualities. Lee et al. use multiple simultaneous neural networks for identifying discrete gestures along with continuous parameters for sound control (M. Lee et al., 1992). Similarly, Modler and Myatt use the outputs of a Time Delay Neural Network for both recognizing gestures and directly controlling continuous volume levels based on output values (Modler et al., 2003). Another particularly strong point of NNs is their ability to produce outputs for inputs not included in the training set (Fiebrink, 2011). In gesture recognition, this means that NNs can produce outputs for gestures that are not in the training set. For continuous expression, this means that they are good at giving output values for inputs between trained ranges.

However, NNs present the interesting difficulty that the structure of a trained network does not reveal how it has learned to classify inputs or whether it has learned the right thing. The structure of the network does not reveal what it has learned about classification rules. Additionally, neural networks potentially require a long time to train a model, even though they do not necessarily require many training examples. Since the entire network has to be re-trained whenever new examples are added, input feature sets are changed, or new output parameters are added, this length of time for training may need to be considered in particular situations where the learning might need to be continuously performed in real time.

Creating machine learning systems for recognizing continuous expressive qualities also presents particular challenges that differ from the standard gesture recognition process. In gesture recognition systems, a system can wait until a gesture has been completed and then output a single "answer," which is then generally used to trigger a particular process. Systems for the recognition of continuous qualities need to produce meaningful "answers" at every point in a movement, not only once at the end of a particular movement. That richer bandwidth of dynamically-changing information between analysis and output is not only beneficial in an artistic context but also necessary, as some outputs may be triggered on or off but many others need richer, more sophisticated control.

Step 7: Test and Post-Process

Once a system has been trained, it can be tested to see how well it has learned the desired concepts. The process of testing generally involves presenting the system with new input examples and seeing how well the output results line up with the desired predictions. In testing, the system is being analyzed for how well it has generalized from the training examples. Given complex inputs such as movement and vocal data, it is impossible that a system can have been given input data examples from every possible case. The key question is how well does the system perform on new inputs that it hasn't seen before? If these inputs are similar to those that it has been trained on, does it recognize that similarity? Another question that is important in performance contexts is the question of what the trained model outputs when it is given input data that is nothing like anything it has seen before, such as new kinds of movements, new voices, etc. Given the behavior of the system after testing, it may be necessary to add additional training examples to help refine its accuracy.

Additionally, it may be beneficial to do some post-processing stages on the output data from the machine learning algorithms. Such post-processing techniques might include such tasks as checking or thresholding an algorithm's confidence values (how sure of this answer is the algorithm?), or combining the results of multiple models (which model is most likely?).

Step 8: Map to Output Control Parameters

Given all of this expressive input at various levels of abstraction (direct sensor data, computed features, high-level parametric representations), the next question is how that input should be mapped to control parameters of output systems. What kinds of performance extension make sense, in the context of a specific piece? The process of creating mappings is at the core of any interactive performance. For the experience of the audience, it does not matter *how* the input about the performance is gathered (through what set of sensors) or how it is processed (through what feature computation algorithms or machine learning algorithms). The core question that defines the experience of an interactive piece is the result of a particular action or type of action. The same sensing, analysis, and output systems can be given drastically different characters through the definition of the mappings between input and output.

Systems for developing mappings need to be flexible enough to allow for both exploring mappings empirically and implementing mappings previously imagined. In the development of an extended performance or installation, it is possible that the creative team may go in with certain ideas about how the expressive control should work, what connections between movement or voice and output media will be interesting or relevant to the piece. However, those original ideas need to be tested quickly to see how they actually feel, especially as the remainder of the content of the piece develops. In other rehearsal processes, there may be little initial sense of how the mappings should work:

perhaps there is a vocabulary of sound that is desired as output, and some knowledge of the movement content of the piece, but the actual relationships between movement and sound will be found through empirical exploration.

The Expressive Performance Extension Framework prioritizes the hand-crafted mapping of abstract parameters to output control, with the machine learning techniques being used to associate sensor data to abstract parameter spaces. This differs from work such as (Stowell, 2010) and (Fiebrink, 2011), where machine learning is used to directly associate movement or vocal inputs with desired output control parameters. Focusing on the intermediate layers and keeping those layers accessible allows for a variety of benefits to the creative process. First, I believe that the development of mappings to output control parameters is a vital part of an artist's creative process in working with an interactive system. Directly using machine learning of mappings may not allow the artist sufficient control and room to experiment in this process. Second, input and output modalities can be more easily changed without requiring the entire system to be rebuilt, because the relationship between sensors and intermediate expressive models is separate from the relationship between intermediate expressive models and output control parameters. Third, meaningful intermediate layers make it easier to define the control of multimodal outputs by one or multiple inputs. Finally, different aspects of the input-output relationship can be created and experimented with at separate times in the rehearsal process. For instance, one could work with a vocalist in a studio to develop an interesting set of abstract parameters to describe her voice and then later explore how those parameters could affect the control of a sound generation system.

Step 9: Refine and Develop over Time

As mentioned earlier, these steps are not necessarily linear, and aspects of many of these steps will grow and develop simultaneously during the course of development of a performance or installation. Systems to support this framework for performance extension need to allow for synchronous shaping and refinement of different layers. For example, one should be able to experiment with ideas about how movement might control a soundscape without having fully committed to a sensing system. As those relationships are explored, important details about what kind of sensing system is necessary may be revealed in the process.

Nor are these steps only completed once in the development of a performance or installation. Just as the content of a performance piece is developed, modified, experimented with, refined, changed, and re-imagined during the rehearsal process, the content of an expressive performance extension will also be continually shaped and invented during the rehearsal process. Development of the story of a piece may demand new interactive mappings, which may in turn require new representations of performance and new sensing modalities. For example, imagine a particular duet dance performance in which the movement of two performers controls generative visuals on the floor all around them. Initially, both dancers' movements are measured via a set of cameras mounted above the stage. In rehearsal, as the story and the choreography continue to develop, it is discovered that the counterpoint movement between performers is a key element, particularly the moments when the two may be performing similar choreography but are approaching the material with very different tempos and amounts of energy. To highlight these differences between performers, it may be helpful to perform analysis on each performer separately (perhaps through separate computer vision systems, if the two are sufficiently far apart, or by adding some wearable sensors that can remain associated with the same performer throughout) and use expressive parameters from each dancer's individual performance to affect the visualization separately. Perhaps an even higher-level control parameter will need to be created that represents the synchronization of expressive parameters between the two performers.

All of the steps in this framework may be relevant to consider at different life cycle stages in a performance or installation, as well. What if an interactive installation is supposed to run for several months? Should it change its behavior at all based on what prior participants have done? Can a system learn over longer timescales, adapt to new inputs, grow beyond its original mappings?

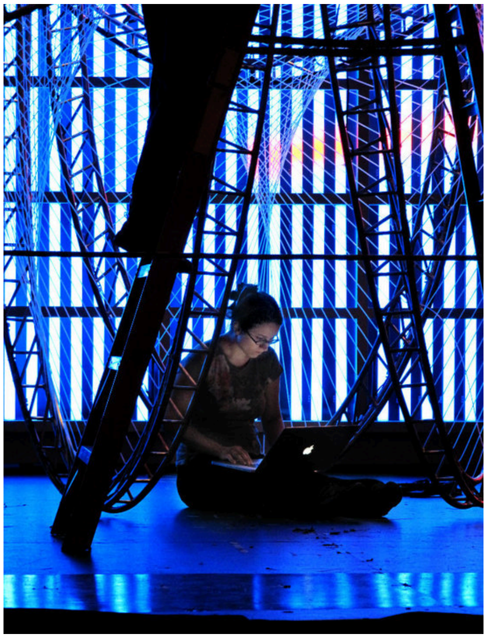


Figure 31. Tuning sensors on the Chandelier
The sensing and analysis for the *Death and the Powers* Chandelier was refined throughout the rehearsal process. Photo by Peter Torpey.

4.5. Conclusions: Goals and Principles for Performance Extension

This chapter and the prior chapter have presented a variety of goals for an interactive system to be smoothly and meaningfully incorporated into live performance and installation contexts. A system should:

- Help extend a sense of liveness through tight mappings from performance to performance extension.
- Incorporate meaningful representations of expressive content.
- Handle both discrete and continuous inputs and outputs.
- Be useful throughout the entire development arc of a piece: ideation, rehearsal, and performance or exhibition.
- Allow for quick sketches of ideas, and rapid iteration during the ideation process.
- Integrate smoothly into existing rehearsal processes.
- Analyze the expression not only of the immediate performance, but of features at longer timescales.
- Support performance analysis and mappings across multiple timescales.
- Handle both modes (states of the system, collections of settings) and triggers (discrete, momentary events: state changes, activation of processes, etc.).

This chapter has outlined a set of questions and principles to be kept in mind while creating technologically-extended live performances. It also has presented the Expressive Performance Extension Framework and a suggested workflow for designing and building interactive performances, particularly those that integrate machine learning. A system that supports this workflow, the Expressive Performance Extension System, will be described in the following chapter.

5. The Expressive Performance Extension System

This chapter presents the Expressive Performance Extension System (EPES), a flexible software system for sensing, analyzing, and mapping expressive performance parameters in live performances and installations. This system implements and tests key design principles of the theoretical Expressive Performance Extension Framework and system architecture described in the previous chapter, particularly the flexible use of machine learning techniques and mappings using user-definable abstract parametric qualities.

5.1. Structure of the Expressive Performance Extension System

The Expressive Performance Extension System extends the Disembodied Performance Mapping System, developed with and originally implemented by Peter Torpey for *Death and the Powers* (Torpey, 2009). This system allows flexible mapping of input data streams to output control parameters through a node-based visual language and is implemented in Java 6. I have extended this system to implement the four-layer framework of abstraction described in the prior chapter: capturing raw input data, specifying and calculating expressive features, using machine learning to abstract higher-level vocal and physical qualities, and facilitating the mapping of high-level expressive parameter spaces to output control parameters. I have also expanded this system to allow the user to work with time as an expressive parameter.

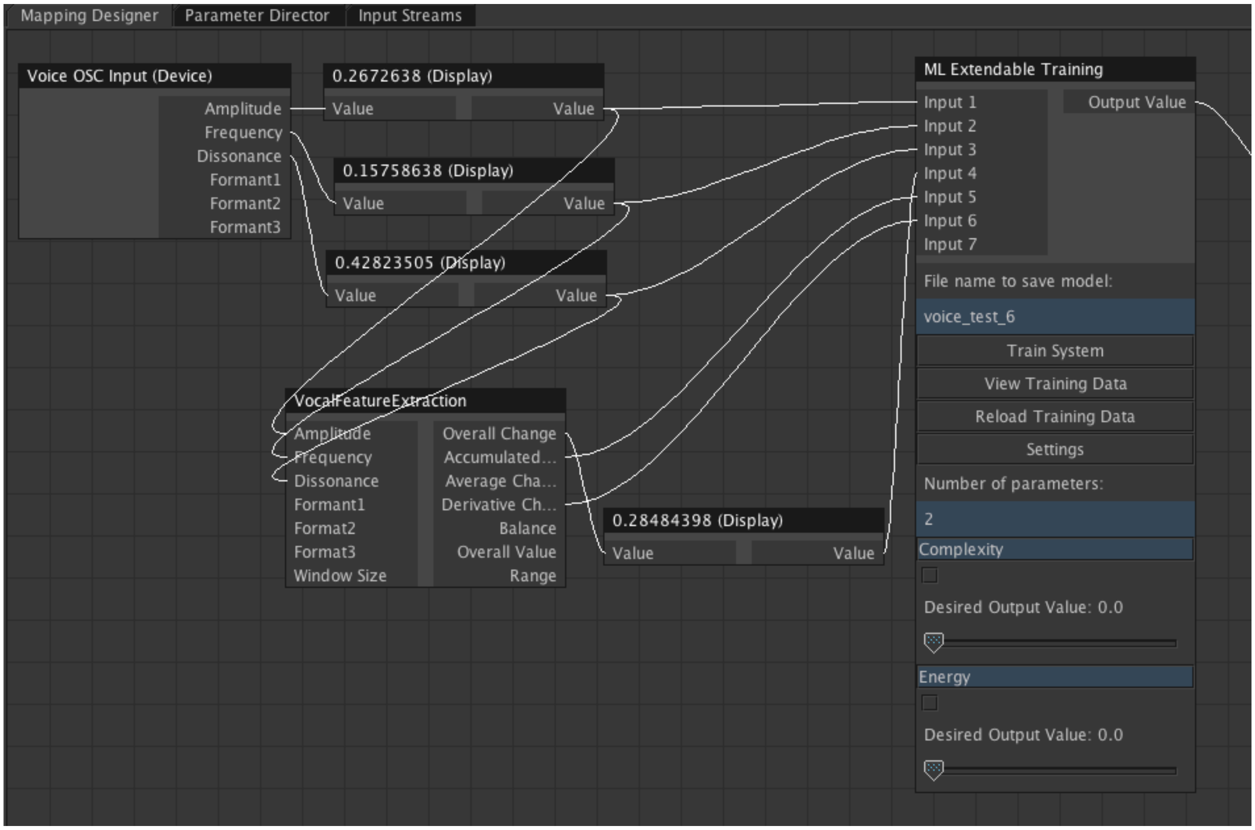
5.1.1. Basic System Flow

This performance analysis and mapping system was designed with the goal of avoiding recompilation of code when adjusting a mapping, allowing quick experimentation in a rehearsal process. The core of this system is the *mapping view*, which allows a GUI interaction with a node-based programming system designed in reaction to the limitations of popular node-based systems such as Max/MSP and Quartz Composer. Each *node* represents an operation on its inputs, which can be a simple or quite complex computation. Individual nodes can maintain their own state. Standard nodes have both *input ports* and *output ports*, representing the data flow between nodes. The output port of one node can be connected to an input port on another node by clicking and dragging the mouse to draw a link. A particular input port can only be associated with one incoming data stream; however, an output port can be connected to multiple inputs. Within the mapping, a selected set of *input devices* handles receiving input data from a variety of sensing systems such as serial microcontroller devices, cameras, microphones, MIDI controllers, and other applications via OSC. An *output device* sends OSC values to control external applications.

A particular set of nodes and links in a mapping is associated with a *cue* in the system. This allows a single configuration of the system to incorporate multiple different input to output mappings. Information about the configuration of the system for a particular production is defined in a *show* file. This XML configuration file includes: global information such as the update rate of the system; information about the input devices used by the production and their configuration settings; properties of the cues in the production and the specific mapping for each cue; and output information (Torpey, 2009, p. 106). Most elements of this XML file (such as the specific mappings) are constructed automatically by the system upon saving a show from the graphical interface.

However, certain elements such as the update rate and details of specific input devices are specified through editing the show file.

Mappings are directed acyclic graphs, with each node in a mapping implementing an abstract `Node` class. Nodes can have any number of input ports and output ports. Data is passed from node to node as floating point values, in a range that is often normalized from 0.0f to 1.0f. Certain types of nodes can handle a variable number of inputs, depending on how many things connect to them, while other types of nodes have a fixed number of inputs. For example, the *invert* node is fixed at one input (its output is 1.0f-input), while the *sum* node can take a variable number of inputs (its output is the sum of values at all inputs).



File name to save model: voice_test_6

Number of parameters: 2

Complexity

□

Desired Output Value: 0.0

Energy

□

Desired Output Value: 0.0

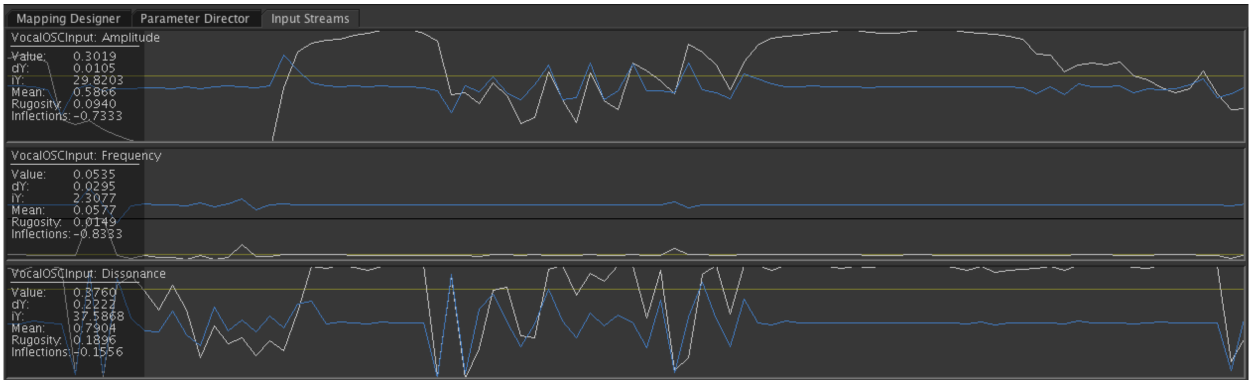
Figure 32. EPES Mapping Designer view

The original Disembodied Performance mapping system contained 22 types of nodes, including primarily arithmetic and statistical functions (e.g. sum, product, negate, min, max, mean), a few generation and control flow nodes (e.g. random, noise, threshold), and nodes relating to data (e.g. input, output, parameter). This original system was intentionally designed to be stateless within a mapping for ease of computation. The current implementation of the Expressive Performance Extension System has integrated a variety of nodes for feature calculation, processing inputs over timescales, machine learning of parameters, viewing data, and flexibly creating new types of operations for nodes.

The entire mapping is evaluated at a frame rate specified in the show file. At each update interval, the values are updated in a recursive depth-first process, starting at the output node and working backwards. This means that nodes that are not connected by some path to the output node will not be updated, avoiding the evaluation of nodes not necessary for producing output.

Input devices provide input data directly to the mapping system. Each device has a number of *axes* that each has a value at any point in time. When devices are connected to the system, their axes are displayed in the interface as output ports on a `DeviceNode` and are available to be connected to other nodes in the mapping. An input device implements the `InputDevice` interface, which provides methods for defining axes, updating the axis values, and retrieving current axis values. Devices are updated asynchronously to prevent against blocking evaluation of the remainder of the mapping while waiting for a new input value, as different types of input devices may have more or less computation or I/O blocking introduced in the device class. For example, a serial input device that gets data from an Arduino may simply store the current values of different Arduino input ports in a number of device axes. A vocal analysis input node may do some processing on a raw microphone signal to compute parameters such as pitch and amplitude, and use those processed values as the axis values for the input device.

Each input device also contains one `DataStream` for each axis, which calculates several parameters for that axis over a window specified in the show file and shared by all axes: normalized value, mean, maximum, minimum, instantaneous derivative, integration, and rugosity. The live data for each axis can be viewed in the Input Streams tab on the interface. In the original implementation of the Disembodied Performance mapping system, a `DataStream` node was the only node in a mapping that incorporated any sense of state rather than only an instantaneous value. However, use of this system in practice during *Death and the Powers* rehearsals made clear that representations of time and analysis of parameters over time needed to be much more flexible and variable, as will be discussed later in this section.



VocalOSCInput: Amplitude

```
Value:      0.3019
dY:         0.0105
iY:         29.8203
Mean:       0.5866
Rugosity:   0.0940
Inflections:-0.7333
```

VocalOSCInput: Frequency

```
Value:      0.0535
dY:         0.0295
iY:         2.3077
Mean:       0.0577
Rugosity:   0.0149
Inflections:-0.8333
```

VocalOSCInput: Dissonance

```
Value:      0.3760
dY:         0.2222
iY:         37.5868
Mean:       0.7904
Rugosity:   0.1896
Inflections:-0.1556
```

Figure 33. Input Streams view in EPES
In the Input Streams view, current values of all input device axes can be viewed, along with analysis metrics.

Additionally, this system incorporates the concept of *parameters*, values that can be adjusted by the user (the mapping designer). When a `Parameter` node is inserted into a mapping, the value of that node can be changed in the Parameter Tab on the interface, and the new value is immediately updated. This assists a user in creating a basic mapping and then fine-tuning it easily while the system is still running.

5.1.2. System Extensions for Machine Learning

This existing system provided a useful platform for input device handling and a node-based mapping interface, allowing me to focus on the redesign of this system to handle temporal input and incorporate higher level parameters. This redesigned architecture implements the four-layered model of data abstraction described in Chapter 4. The first level is the raw sensor data from inputs such as wearable sensors, video cameras, and microphones. The second level consists of computed features of the sensor data (particularly temporal features) that are related to expressive content. These features can be computed through machine learning techniques or hand-coded algorithms. The data at this level can then be associated with the third layer of abstraction: high-level parametric spaces of expression and temporal descriptions of trajectories through those parametric spaces. These associations between features and parametric spaces are created primarily through pattern recognition techniques. Finally, those high-level spaces can be manually mapped to the fourth level of data, parameters for the control of output media. The interactions and associations between each of these four layers are developed throughout the design and rehearsal process of a performance or installation, and can potentially be shaped by later performance input as well.

This architecture differs from existing work in the analysis of physical expression in several important aspects. First, it prioritizes the final representation of gestural expression as continuous trajectories through continuous expressive spaces defined by semantically-meaningful parameters, rather than as discrete categorization into semantic descriptions of expression (as in Camurri, De Poli, Leman, & Volpe, 2001), or as specific recognized gestures. Additionally, this architecture locates the creative practice of developing mappings at a higher level of abstraction than at either the raw sensor data or the expressive feature spaces. The performance-maker's mapping process between gestural inputs and output controls is designed to take place at the level of the continuous expressive space, while built-in machine learning algorithms and feature analysis tools can handle the association between sensor data and a particular expressive parameter space. Finally, the goal of generalization in this architecture is intrapersonal rather than interpersonal. The system should learn different relationships between layers for individual artists and individual pieces, not force one general relationship between sensor data, features, expressive spaces, and output control parameters. EPES allows users to create their own sets of input sensors, of computed features, of expressive spaces, and of output parameters.

A major emphasis for the machine learning components of the Expressive Performance Extension System has been to allow the user to interact with pattern recognition algorithms at a very high level; a user of the system need not know what algorithm is being used, nor the details about how that algorithm has been set up. A user can select or define the desired expressive parameter space and which input features are to be used in training, easily capture and modify sets of labeled sample data, and have a few simple handles to allow some tuning of the recognition system. A user who desires more complex interaction with the recognition system can access that through additional settings, but these settings need not distract the user who is unfamiliar with machine learning techniques.

I have added machine learning nodes that incorporate the Encog Machine Learning Library (Heaton, 2008), a library implementing a variety of popular machine learning algorithms including Hidden Markov Models, Support Vector Machines, and Neural Networks. Encog provides libraries for a variety of programming languages, including Java, and is designed to allow programmers to integrate and interact with many machine learning algorithms without having to implement them from scratch. As the focus of this dissertation is on the use of machine learning algorithms in expressive contexts, particularly for learning high-level expressive parameters, rather than on the details of any particular algorithm, this library was seen as a suitable base on which to build machine learning nodes for the Expressive Performance Extension System. As discussed later in this chapter, training nodes and evaluation nodes allow users to flexibly explore different algorithms.

5.2. Representations of Time

One of the major dimensions in which the Expressive Performance Extension System has been extended from the original version of the Disembodied Performance System is in the representations of temporal features available within a mapping, as well as notions of state and time. The original system was designed to be stateless for efficiency. The only temporal data was used for the `DataStreams` associated with each input device's axes, which calculated the average value, derivative, rugosity, maximum value, and minimum value of each axis over a fixed window of time specified in the show file. As soon as we started developing content for *Death and the Powers*, it was clear that stateless mapping nodes and a fixed window length for `DataStreams` were not nearly sufficient for live performance data. For example, when smoothing an input signal by keeping a running average of the signal, the number of values that are incorporated in the average has a large impact on the resulting balance of reactivity and smoothness in the output. One system-wide window size for averaging will not be sufficiently flexible for many mapping scenarios. This led me to design and implement a number of new nodes used in the *Powers* mappings that kept track of state and incorporated knowledge of data over time (such as a `QualityAnalysisNode` converting movement data from the *Powers* wearable sensors to values in a modified Laban space of *time*, *weight*, and *flow*) and nodes that were designed to work over user-defined lengths of time (such as an `AverageOverTimeNode`, which calculates the average value of its first input stream over a window length specified as a second input to the node). Typically, nodes that take window lengths as an input port are hooked up to a `Parameter` node whose value is adjusted while developing a mapping and then left at an ideal value. However, these input ports also allow for the use of a continuously variable window.

The Expressive Performance Extension System recognizes and addresses time as an important parameter of performance and of performance design. As multiple different temporal scales are especially important in the analysis of expressive performance qualities (as described in Chapter 4), this system incorporates knowledge of multiple temporal windows into mappings. First, the system can define a performer's instantaneous position in an expressive parameter space. Second, the system has parameters and structures to describe trajectories over different timescales through the instantaneous expressive spaces, allowing the user to define or scale expressive information in the context of what has come before. Finally, the system has the capability for a user to create separate "cues," each of which has a particular mapping from input data to output control parameters. These cues provide the system with knowledge of a larger-scale temporal structure. The system is designed to allow a user to fluidly shift between actions at any of these different timescales.

Various nodes and operations have been developed as part of the Expressive Performance Extension System to address the ability to examine and work with parameters at multiple timescales.

- `1DGraphNodes` provide the ability to view a value in the patch as it changes relative to time, where the desired value is given as an input. The timescale that the graph covers can be specified by the user.
- `2DGraphNodes` allow the user to view the current point and its historical trace in a two-parameter space, where the two axis values are given as inputs.
- `PatternComparisonNodes` allow the user to draw a shape of a target curve over a specified length of time (*n* frames). The input values to this node over the previous *n* frames are then stored and compared to the target curve in two ways. First, the node outputs the absolute difference of the target curve and the input curve: averaged over all points, how far from the target value is the input value at each point. Additionally, the node calculates the "relative difference": the average over all points of how the derivative of each pair of points in the target curve compares to the derivative of the corresponding points of the input curve. This is a comparison of the relative shapes and patterns of variation of the line, rather than of the absolute values.
- `RampNodes` continually increment or decrement their output value between 0.0 and 1.0 based on their input value. The maximum and minimum expected inputs can be specified.
- `AverageOverTime` nodes and `AmountOfChange` nodes calculate parameters of an input stream over a given window.

The Expressive Performance Extension System handles variation in behavior over time at a longer scale through the concept of *cues*. Each cue is associated with a particular input to output mapping flow. OSC messages sent to the system can place the system in a desired current cue. This is the primary way that the system represents structures that change over longer scales of time, such as different sections of a piece. In addition to changing the details of one mapping based on performance data, cues allow a user to create different interactions for the same performance data values as a piece progresses. For example, in one movement of a work the performance designer may want to use fast, smooth movements to control a particular soundscape; a minute later, the same kind of movements are desired to control a very different soundscape as the sound design of the piece progresses. Cues allow for this kind of long-term variation over the course of a piece.

Cues can be sent from external systems to coordinate the program with other show control systems, or to select cues through actions of a performer (such as pushing buttons or using foot pedals to switch to a desired cue). It is also possible to use helper programs to allow a mapping in EPES to trigger a desired new mapping cue based on performance information within a cue.

5.3. Steps for Machine Learning of Expressive Parameters

In this section, a subset of the steps described in Chapter 4 for selecting and training a system on expressive parameters are shown as they are implemented in the Expressive Performance Extension

System. The details of this process in EPES are discussed along with descriptions of how a user can perform these actions.

5.3.1. Select Expressive Qualities

EPES has a suggested abstract parameter set incorporated into the machine learning nodes. The default parameter set incorporates six expressive parameters that are broadly applicable to physical and vocal analysis. In these definitions, "input" can refer to information from the body and/or the voice. The precise physical or vocal definition of any of these parameters will vary from piece to piece and from performance-maker to performance-maker.

- *Energy* (calm to energetic): Strength and animation of the input.
- *Rate* (slow to quick): Frequency of events, speed of gestures.
- *Fluidity* (fluid to sharp): Continuity of the input. How smoothly is the input changing from moment to moment? How legato or staccato is the input?
- *Scale* (small to large): Magnitude of the input, relative to some range.
- *Intensity* (gentle to intense): Weight and tension of the input.
- *Complexity* (simple to complex): Amount of variation of the input, across many aspects and scales.

EPES can also handle any other set of parameters desired by the user. These parameters, like all values passed in EPES, should be considered to have floating point values between 0.0 and 1.0.

5.3.2. Select Sensors

EPES has the capability to process a wide range of input sensor information. The primary `InputDevice` interface is taken from the Disembodied Performance system and has been extended in this implementation to include meaningful input nodes for a wider range of input modalities, as well as to include generic input nodes to support rapid prototyping and exploration. Several varieties of input devices are discussed here, along with the process of constructing a new type of device to handle a novel sensing mechanism.

All input devices can take values in any range, which are mapped to floating point values in the range 0.0 to 1.0. The construction methods for a particular input device set the number of input parameters and the minimum and maximum expected ranges. Each input device also has an identifying "address," which is set and stored in the saved show file. In different types of input devices, this address may refer to a serial port identifier, an OSC port, or a webcam identification string. The types of input devices used for a particular show are currently defined by the user in the show file.

Arduino Input Devices

EPES currently implements a range of input devices that can be used for capturing data from wearable sensors. These nodes assume that the sensors in question are connected to an Arduino board or to a Funnel I/O board connected as a serial device via XBee radio modules. The generic `ArduinoInput` sends all analog data streams from a Funnel or Arduino board, each on a different input data axis. These axes are by default given generic labels corresponding to the associated Arduino analog input pin (Analog0, Analog1, etc.).

A series of specific Funnel-based input nodes have also been constructed manually for particular combinations of wearable sensors. `DanceGlove` input devices are designed for a glove or armband with two three-axis accelerometers. For the Disembodied Performance System, a `BreathBand` input device was designed expecting only one axis of input, the resistive stretch sensor for measuring breath. The sensors on the VAMP glove (accelerometers, bend sensors, and pressure sensors) are the expected inputs for another constructed input device. Any of these specialty devices could also now be implemented with a generic Arduino device.

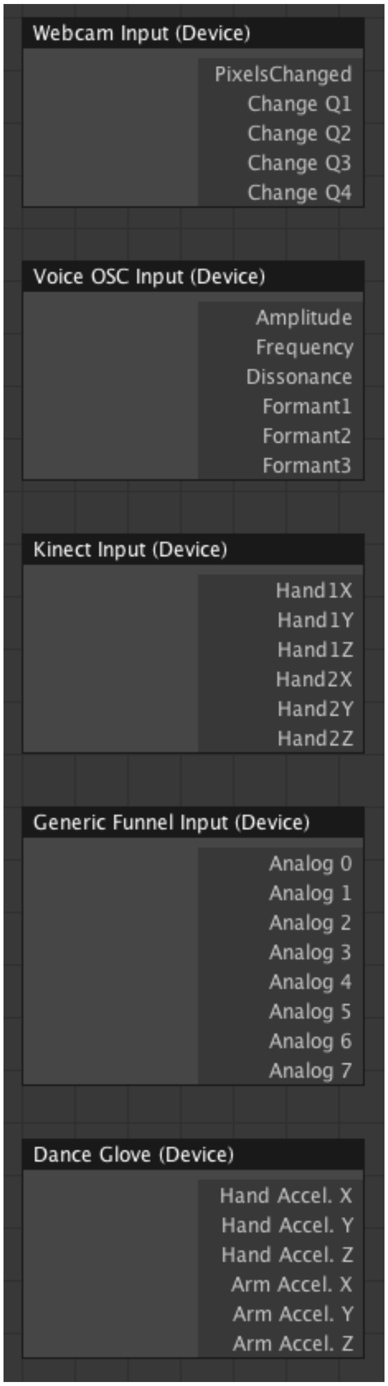


Figure 34. Various input devices for EPES

Kinect and Webcam Input Devices

EPES currently implements a few different input devices for handling input from a Kinect via the `SimpleOpenNI` library for Processing. Each of these input devices is connected to a separate Processing applet running in its own thread that receives video frames and depth images from the Kinect, as well as data on skeleton tracking and hand tracking, and processes that information into a few relevant parameters that differ for different types of input nodes. The most basic version, the `KinectInput`, outputs six parameters: the XYZ location of up to two hands tracked by the Kinect. If a hand is lost, this node will continue to output the last known location of that hand. The `KinectSkeletonInput` device outputs the XYZ position of several points on a detected skeleton: right hand, left hand, and head. Additional possible tracked points include shoulders, elbows, knees, and feet. The `KinectWebcamHandsInput` performs hand tracking and additionally treats the Kinect as a webcam, providing metrics of the amount of variation from frame to frame of the camera (a metric of activity). Additional parameters include the amount of variation within four different regions of the camera, which can be calibrated in the class file to represent four horizontal regions, four vertical regions, or four squares within the image.

In all of these devices, it is possible that the hand assigned as "hand 1" may fluctuate between left and right hands, depending on the order of acquisition of hand identifiers. Future extensions of these nodes could maintain some sense of absolute position in determining which hand should be assigned as "hand 1" versus "hand 2." Of course, this would not accommodate situations in which one moved a hand all the way across the body, for example. Additionally, if hands are moved too rapidly, the `SimpleOpenNI` libraries will repeatedly detect and lose hands. When hands are reacquired, they may have switched from "hand 1" to "hand 2" or vice versa, and produce spurious behavior. The built-in EPES feature calculation nodes for the Kinect attempt to minimize this issue.

EPES also includes a `WebcamInput` device type that provides the overall activity and four sub-region activity metrics given the analysis of an image from a webcam that can be built into the computer or connected via USB. This input type relies on an external PApplet using the Processing implementation of OpenCV for image processing. The identifying address of the desired webcam is specified in the show file by a number that can be determined by running any OpenCV software and printing the inputs it finds. The frame rate of the webcam input is the same as that of the overall show, as the input device requests the webcam applet to update on every device update step.

These Kinect and webcam input nodes illustrate an important layer of feature computation that needs to occur with any kind of complex input modality. The Kinect provides only a color image and an image representing the depth of every pixel. A webcam provides only the color image. In order for this information to be used in any meaningful way in an input node, specific features to be computed are selected. Some of these may be simple features such as the amount of pixels that have changed from the previous frame to the current frame (which requires the use of memory to have knowledge of the previous frame). Others may be more sophisticated features, such as hand tracking, that are accomplished by external libraries. Blob tracking has also been explored in the `WebcamInput` type, though the OpenCV implementation of blob tracking is not efficient enough to track more than a few points, and so is not generically useful. In any case, the image or depth image itself is not sufficient as the output for an input device type, so the input device makes visible to the system some computed features of the image over time.

Audio Input Devices

EPES also incorporates input devices that perform audio analysis within the input device. The `AudioAnalysis` device uses the Minim libraries for Processing. This input device makes three main parameters available to the mapping system: amplitude, frequency, and consonance. All of these parameters are computed via a Fast Fourier Transform. The FFT band with the greatest amplitude is the frequency value. Consonance is calculated as a simple measurement of how close the second and third strongest frequency bands are to multiples of the fundamental frequency.

OSC Input Devices

EPES can also receive input from any system that speaks the Open Sound Control protocol. Specific subclasses of OSC input nodes listen on a set of OSC addresses defined in the class. OSC nodes expect messages with a single integer, float, or double argument. The OSC port number for each input node is defined in the show file.

Additionally, a `GeneralOSCInputNode` allows a user to specify in a file any address to listen to and a range of values expected as arguments to that address to be mapped to the range 0.0 to 1.0. OSC addresses, identifying labels for each input stream, and input ranges can be added to, modified in, or removed from the node through editing a properties file associated with the input device in the main show file. This allows a user to rapidly iterate with desired OSC addresses for input without having to create or modify a special OSC input subclass for that device. In addition, as files can be reloaded without the system being restarted, this allows for rapidly changing OSC inputs.

MIDI Input Devices

Another class of input device implemented in EPES is a `MIDIInput`, designed to transform the input of MIDI controllers into a format that can be used in mappings. The implementation uses the proMIDI libraries for Processing and Java, with the addition of a supplementary PApplet owned by the `MIDIInput` class to handle the automatic callbacks from the proMIDI library. The current implementation of the `MIDIInput` device receives information from continuous MIDI controllers, using a device address specified in the show file. To stay reasonably generic, this device node has outputs for the first eight controllers on Channel 0. These values are transformed from the standard MIDI scale of 0 – 127 to 0.0 – 1.0. Future extensions of input devices that integrate MIDI could allow the user to specify more specific MIDI controller identification for each desired device (controller 1 on channel 0, controller 10 on channel 2, etc.).

This device also incorporates some handling of discrete MIDI notes as well as continuous controller values. One challenge in the mapping process is determining what information is relevant from each MIDI note. It is unlikely that an ideal MIDI input ought to have 127 separate ports, one corresponding to each note (though this is the case in some systems, e.g. Quartz Composer). In the current EPES implementation, the device outputs the current note number value and velocity value (both scaled from 0 – 127 to 0.0 – 1.0). A future issue to extend MIDI handling functionality is how to handle maintaining duration information about specific notes, rather than treating the onset of a note as a discrete trigger. If the duration of notes is relevant, there are design questions about how to manage polyphony.

5.3.3. Select Features and Perform Feature Computation

After input devices are selected, the input device nodes can be hooked up to nodes designed for feature computation. Importantly, this feature computation stage allows for many different kinds of simple to complex features to be analyzed about an input data stream or set of data streams. This can include basic features such as temporal statistics that do not care about what kind of data is used as an input (the average value of an input over a particular temporal window length, the derivative of a particular input stream) and more sophisticated features that take advantage of some knowledge of the form of the input. There are a variety of options for feature computation for specific sensor types built into EPES, as well as generic feature computation strategies. It is important to note that many of the specific input devices already provide some layer of feature computation in their design, such as the `WebcamInput`, which outputs the amount of pixels changed in the image and in quadrants of the image since the prior frame. The additional feature computation nodes currently implemented primarily handle calculating features of the input over time.

For several of the existing input device types that handle particular sensors, such as the `DanceGloveInput`, `KinectInput`, `WebcamInput`, or `AudioAnalysis`, individual hand-crafted feature computation nodes have been developed that calculate a particular set of features from the input parameters. Many of these features are calculated over a window length that can be specified by the user. The set of features that is calculated by these nodes was selected to cover some of the specific types of features determined to be relevant in vocal and physical expression, particularly metrics of energy and variation over different time scales. The specific relationship between input parameters and output feature values is calculated differently for each node. This relationship has been empirically determined by hand and scaled between 0.0 – 1.0. The features included in this analysis are: *overall change* (how much each input parameter has varied from frame to frame over the past window); *average change* (the average overall change of the inputs over the past window); *derivative change* (the amount of change over all input parameters in the past four frames, looking at a smaller window of time than the overall change value); *balance* (a metric of how much the input is varying similarly or dissimilarly across dimensions); *overall value* (a weighted average of all parameters); and *range* (how far apart the smallest and largest values of each parameter have been in the last window).

A final feature, *accumulated change*, requires additional explanation. In the development of various productions such as *Death and the Powers*, a feature of expressive input that I have found to be particularly relevant is a metric of how much variation there has been over longer timescales than an immediate window. If the input parameters have been changing a large amount, the accumulated change value will gradually increase (to a maximum of 1.0). If the input parameters have not been changing, the accumulated value will gradually decrease (to a minimum of 0.0). The value will vary on each frame by an amount proportional to how much the inputs have been changing. Thus, *accumulated change* is a metric of the historical variation of the input parameters. Typically, increment and decrement ranges for accumulated change values are not set evenly, such that the value requires “more effort” to bring high, and returns to its baseline of 0.0 without sustained high variation. This accumulation value is particularly useful as it reflects the “norm” of a piece over a length of time. One sharp gesture conveys a different type of meaning than one sharp gesture in the middle of many other sharp gestures. If a particular mode of movement continues for a period of time, we grow accustomed to it. This *accumulated change* value seeks to capture that sense of normalization.

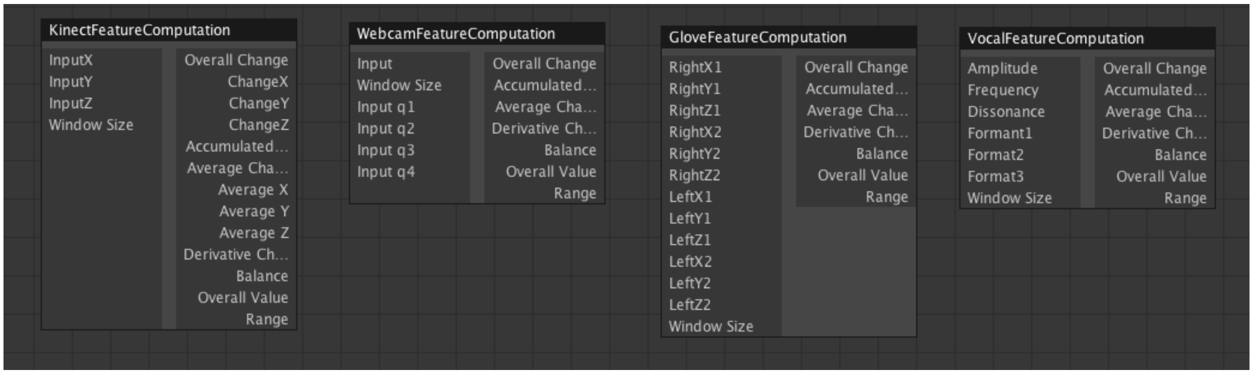


Figure 35. Sensor-specific feature computation nodes for EPES
Each of these nodes takes the output parameters of a particular input device and calculates a similar set of features: overall change, accumulated change, average change, derivative change, balance, overall value, and range.

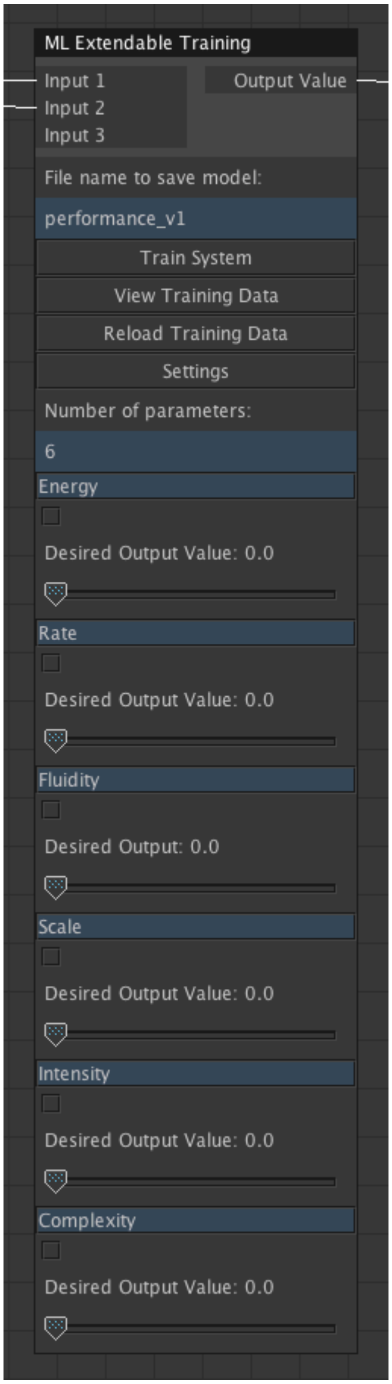
Other types of nodes in EPES have been designed for the computation of features from any type of parameter input, rather than calculated for a specific sensor setup. The `ChangeNode` calculates the amount of change in the input parameter over the specified window size in frames. This amount of change is calculated by summing over the window the difference of each stored value and the prior value. An `AverageOverTime` node calculates the average value of the input parameter over the specified window size. The `RampNode` performs the calculation of accumulated change discussed previously, with values for the amount of incrementing and decrementing specified by the user. The `TemporalScalingNode` automatically scales the current input value based on the highest and lowest values that have been seen in a specified window. It is important to note that all of these features are features of the input over time, rather than immediate properties of the input. All of these allow the user to adjust the window size as desired to examine different timeframes. Generally a static window size is used for a particular instance of the node, which can be set via a `Parameter`. Other nodes compare the current values of multiple input parameters to determine features of the input, such as the `MaxNode`, `MeanNode`, and `MinNode`.

In addition to the built-in feature computation nodes, other processes can be used for feature computation outside of EPES with the results brought into the system via an input device (either through OSC or through a custom input device). This allows the quick integration of existing feature computation or feature extraction techniques in whatever system is most convenient for the user. For example, in the Vocal Vibrations installation described in Chapter 6, the EPES input node received via OSC a variety of vocal features calculated in an external Max/MSP patch. These features were all computationally calculated from the current input signal (such as voice loudness, fundamental frequency, and spectral centroid), with temporal features calculated in EPES.

It is important to note that the machine learning algorithms used for EPES are intended to be able to handle both relevant and irrelevant features of the input. This is an important distinction to have the system be functional: part of the reason for incorporating machine learning techniques is that users may not always know what features of a given input signal are relevant for identifying the quality of movement or voice that they care about. Thus, while the system supports selecting a variety of features that may be particularly expressive, it does not dictate that only features known to be relevant can be used as inputs to the machine learning stage.

5.3.4. Collect Training Data

Once a sample set of expressive parameters has been chosen, the user can then collect training data examples labeled with different values of those expressive parameters. An `MLExtendableTrainingNode` allows a user to specify the desired number of expressive parameters and the names of those parameters. Any number of inputs can be connected to these nodes. These nodes contain an instance of the `TrainingDataCollector` class, which handles storing values of all inputs across each time frame and compiling collections of values into *samples*. Starting the mapping system begins collecting data and stopping the system ends a sample. Each parameter has a slider that represents the current desired value of that parameter for the given sample. These values can be set before starting to capture each sample, but should not be changed mid-sample.



File name to save model: performance_v1

Number of parameters: 6

Energy
□
Desired Output Value: 0.0

Rate
□
Desired Output Value: 0.0

Fluidity
□
Desired Output: 0.0

Scale
□
Desired Output Value: 0.0

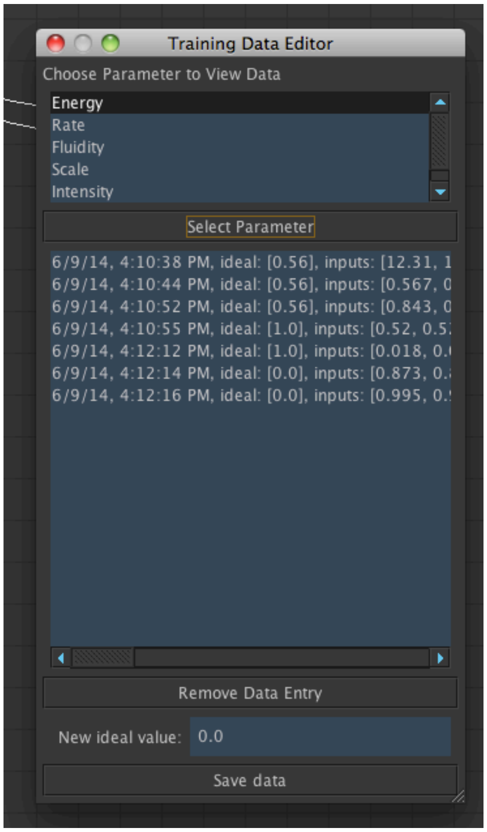
Intensity
□
Desired Output Value: 0.0

Complexity
□
Desired Output Value: 0.0

Figure 36. EPES machine learning training node

Via a checkbox, the user can select which parameter or set of parameters should be active for a given sample. If a parameter is active, the current sample should be added to the stored dataset for that parameter. This allows the ability to train each parameter individually or in small combinations as desired, rather than having to invent training data examples that vary across all parameters simultaneously. For example, say one is capturing vocal training data on a set of parameters including *complexity*, *rate*, and *intensity*. If all parameters were captured simultaneously, for each training example to be given to the system the user would have to determine how that example should be valued along all three axes. It is a much easier process to envision a set of examples that cover the desired range of different levels of complexity, a second set of examples that have different rates, and a third set of examples with different intensities. While it is useful to have some variation across each set of examples, as discussed in Chapter 4, this model is simpler for a user to conceptualize.

Once some data samples have been captured, a user can view and edit those samples as desired. By clicking the "view training data" button on an `MLExtendableTrainingNode`, a separate window is launched that provides the ability to view the training examples currently saved for each parameter, change the value label for any example, and remove any example. This ability to rapidly modify training data is especially useful for correcting mistakes in the training data capture process. For example, say that you are in the middle of capturing movement data samples that are intended to have low intensity. Then you switch to high intensity examples, but forget to change the value of the parameter slider until you have captured two examples. The training data window allows you to quickly adjust the label on those two examples, rather than discard all of the previously collected data.



```
6/9/14, 4:10:38 PM, ideal: [0.56], inputs: [12.31, 1
6/9/14, 4:10:44 PM, ideal: [0.56], inputs: [0.567, 0
6/9/14, 4:10:52 PM, ideal: [0.56], inputs: [0.843, 0
6/9/14, 4:10:55 PM, ideal: [1.0], inputs: [0.52, 0.5
6/9/14, 4:12:12 PM, ideal: [1.0], inputs: [0.018, 0.
6/9/14, 4:12:14 PM, ideal: [0.0], inputs: [0.873, 0.
6/9/14, 4:12:16 PM, ideal: [0.0], inputs: [0.995, 0.
```

New ideal value: 0.0

Figure 37. Training Data Editor window
In the Training Data Editor, users can view saved data samples for any parameter, remove particular samples, and re-label samples as desired.

One technical issue to be addressed with temporal data is that all training examples of particular qualities (whether vocal or physical) will be varied in length (number of frames). This makes such examples harder to compare to one another, as well as to format to fit machine learning algorithms that typically expect all examples to be of the same length. The Expressive Performance Extension system makes the choice to standardize all training data examples and testing data examples to the same normalized length, which can be specified by the user in the Settings window of the node. Early versions of the system obtained this length standardization by trimming longer samples and zero-padding shorter samples. The current version of EPES performs the standardization process through stretching or compressing the input data vectors to fit a normalized length. For vectors that are too short, the system linearly extrapolates extra data points between the original data points to extend the vector to the desired length. The system handles vectors that are too long by averaging multiple data points from the original vector into one data point in the new normalized vector. This process is performed separately on each data source incorporated in the input vector. For example, if the input data is captured from three sensor streams, the data vector sent to the machine learning systems will consist of interwoven data points from those sensor streams, in temporal order: [x1, y1, z1, x2, y2, z2, x3, y3, z3]. Simply normalizing this vector would not be meaningful, as the data streams are separate and their values should not be averaged together or extrapolated between. Thus, the algorithm first separates the data into component vectors ([x1, x2, x3], [y1, y2, y3], and [z1, z2, z3]), normalizes each component vector to the desired length (for example into [$x_2$1, $x_2$2], [$y_2$1, $y_2$2], and [$z_2$1, $z_2$2]), and reintegrates these normalized components into a new normalized vector ([$x_2$1, $y_2$1, $z_2$1, $x_2$2, $y_2$2, $z_2$2]). This normalized vector is then used as input to the desired machine learning process. Developing this strategy for temporal normalization of training data was a key step in preparing training data examples that could be used for regression purposes. This technique allows the use of machine learning algorithms that require a fixed number of input dimensions, such as Neural Networks, without padding the inputs with meaningless information. This technique also greatly simplifies labeling data.

It is important to note that because of this normalization step, it is desirable for training data examples to be captured that are of fairly similar lengths. For example, say the movement being captured is different tempos of waving a hand back and forth, and the input data is the position of the hand. If two examples are reasonably similar in length, the example with faster waving will have many more inflections when the hand changes direction than the slow waving example. However, say that the fast waving is performed twice as fast as the slow waving, but only captured for half the length of time. When the two examples are normalized in time, they may appear much more similar than intended.

For rapid capture of example training data, a foot pedal was suggested by Marc Downie as a useful addition to the system. A Yamaha piano pedal has therefore been integrated to start and stop data capture, with data capture beginning when the pedal is depressed and ending when the pedal is released. This allows for several samples of movement or vocal information to be easily obtained from one continuous example, without having to pause in between to stop and start the system manually.

The additional advantage of this pedal is the capture of cleaner data. In the capture of a training data example, it is important to examine whether the entire example is representative of the quality intended. In a system started and stopped with a keystroke, mouse click, or particular gesture, part of those auxiliary movements and potentially the sounds of those actions will also be captured in the training data. This issue then requires the data to be preprocessed (for example, removing the beginning and end of the sample) so as to remove traces of undesired movement. With vocal data or movement that is primarily upper-body movement, the movement required to activate a foot pedal should not affect the details of the data that is being captured.

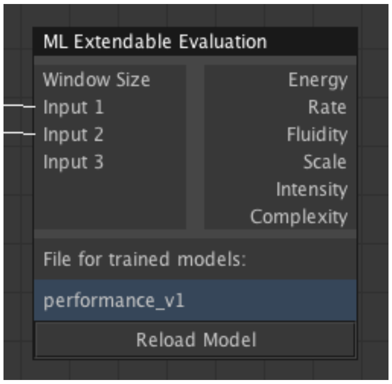
5.3.5. Train Model

When some example data has been collected, a model can be trained to recognize the selected parameters. The current implementation of the Expressive Performance Extension System requires labeled data, with the labels being a floating point value from 0.0 to 1.0. Since the system is designed for the ability to analyze continuous parameters, the algorithms currently implemented in the `MLTrainingNodes` and `MLEvaluationNodes` are all capable of performing regression. To treat this as a classification system on a particular axis (recognizing whether a movement is "slow" or fast") rather than a regression system (recognizing "how fast" a movement is), training data could be saved with only 0.0 or 1.0 values and the output could be thresholded in a post-processing step. The current implementation of EPES allows the selection of Neural Networks or Support Vector Machines.

EPES incorporates the Encog machine learning framework, which includes a set of Java libraries implementing a variety of machine learning algorithms as well as classes for processing and normalizing data. Individual EPES classes are written to allow interaction with the various Encog classes that implement particular algorithms. EPES defines an `MLHandler` interface, which includes a variety of basic functions including the ability to train a model with a given dataset, to save file representations of trained models, and to evaluate a particular example according to stored models. This generalization allows the process of capturing training examples to be completely separated from the selection of a machine learning algorithm to train. When the user hits the "Train System" button on the `MLExtendableTrainingNode`, a model is created and saved to the filename specified, using the Encog libraries for serializing model information.

In the current version of the system, each expressive parameter is handled with a separate model and trained independently. For example, when the user selects Neural Networks as the desired training algorithm, a separate network will be trained for each parameter, given the separate data set captured for that parameter. This allows for partial training of systems: say that the user has four parameters that he will want to work with eventually, but wants to test the behavior of only one parameter at the moment. Since training data examples are captured separately by parameter and a separate model is trained for each parameter, the user can develop a model that he is happy with for his first parameter. His further actions to gather data on other parameters and train the system on those will not affect the model of the first parameter.

Once a model has been trained, that model can then be tested and used via an `MLExtendableEvaluationNode`. These nodes, given the folder identifying the location of a desired trained model (or set of models, in the case of multiple parameters), create an instance of that model that can then be given new samples to generate output values. As with the `TrainingDataCollector` structures for the training nodes, a `TestingDataCollector` supports the breakdown of a stream of live data into samples to be run through the stored models. As this system is optimized for the analysis of continuous parameters, this `TestingDataCollector` does not have to do sample segmentation on the basis of specific features of the signal, as would be the case to improve performance in a gesture recognition system. Instead, the last N points for each input stream are stored and passed as a sample, where N is the window length defined by the user (which is expected to correspond to the normalized sample window length specified in the training process). Thus, a new sample is evaluated on every frame of the system.



File for trained models:
performance_v1

Figure 38. Machine learning evaluation node

5.3.6. Refine and Iterate

As discussed earlier, one of the key requirements of working with training data sets is the ability for a user to quickly and easily modify the data captured as part of the workflow. What happens if a user accidentally records a data example with the wrong label or output value set? What happens if a user adds examples that she then does not think are “good” examples, or that decrease the performance of the system? What if a sensor was malfunctioning, causing problematic input data? What if a user wants to copy some good training data captured in the process of making another piece, or to combine several sets of training data? It is necessary for a system to allow the user to change stored output values and labels, as well as to easily add and remove entries.

Initial representations of training data in EPES were simply text files containing input data sets associated with the desired output values corresponding to each input set. However, easy modification of the training data by users demanded that more information be stored with the training data set, and that this data be accessible through the EPES interface. Each data example is now timestamped so as to make it easier to identify which example was labeled in error. A button on each `MLTrainingNode` brings up a window through which users can explore and adjust the data in the training file associated with that node. The user can select a parameter to view all associated training data examples. This interface window allows for deletion of and relabeling individual examples. Once the training data has been modified to the user's satisfaction, hitting "save" will store this new training data set and load it into the associated machine learning training node, so that all changes become active.

More extensive modifications such as combining training data examples from multiple existing files can be performed directly on the files by copying the desired examples into the new file. Each `MLTrainingNode` has a button to force a reload of training data, which can be used if the training data has been manually modified on disk. Similarly, a `MLEvaluationNode` has a button to reload a trained model from a file, in case additional data has been added and the model retrained since the last time it was loaded.

As discussed in Chapter 4, one of the key design principles of systems for performance extension is the ability to create and adjust mappings on the fly while the system is running without needing a separate compilation stage. In order to experiment more flexibly and rapidly with mappings, JavaScript evaluation nodes have also been developed for EPES. These JavaScript nodes can take any number of inputs and evaluate a JavaScript program entered into the node to determine the output value. The JavaScript code is edited in a separate popup frame; when the new code is saved, that code is immediately evaluated on the next output step. These nodes are designed for rapid iteration of new processing techniques, so that several variations can be explored in real time without the need for coding new nodes and recompilation. If a particular functionality is found to be especially useful and sufficiently generic, a new mapping node can then be developed offline that implements that functionality. However, it is preferable not to have to code a new node every time one needs a new kind of analysis or mapping procedure, especially if that analysis is not particularly general but better suited for a specific mapping.

These JavaScript nodes also offer the ability to compress basic equations that might take up unnecessary visual space in the mapping. For example, a scripting node could easily contain within one node the code to take two inputs, multiply them together with a specified value, add the result to a third input, and output that value if it is over a given threshold or 0 if it is under the threshold. To do this in the standard mapping system would require a multiplication node, two parameter nodes, a sum node, and a threshold node. A possible extension of this concept would be to add the ability to create expandable sub-mappings within a mapping patch, as is offered in Quartz Composer and Max MSP. In these programs, a patch can contain sub-patches that themselves are complete input-to-output mappings.

5.3.7. Integrate with Other Components and Systems

Like the majority of the show control systems that have been developed in the Opera of the Future group over the past several years, the Expressive Performance Extension System communicates via the Open Sound Control protocol using custom Java libraries designed by Peter Torpey. The Open Sound Control protocol is widely implemented and allows EPES not only to communicate with a variety of our custom input systems and output systems but also to directly communicate with many popular existing systems for generation of sound and visualizations, such as Max/MSP, Processing, ChucK, Pure Data, and Quartz Composer. If the desired system to be controlled does not take OSC directly, helper programs such as OSCulator allow for easy transformation of OSC messages into other formats, such as MIDI triggers or control messages. This allows EPES to communicate with other popularly used systems such as Ableton Live, which can receive MIDI but not OSC directly.

The original Disembodied Performance System was designed to have one output node with a particular predetermined set of output parameters corresponding to the control parameters of the `RenderDesigner` visualization system. For the continuing use and extension of the system, the output node has been generalized to output to any set of OSC addresses specified in the show file. Additionally, the output node can send output OSC messages to multiple network addresses and multiple OSC ports. In the future, an output node might store which messages are relevant for which port and which network address, so as to eliminate unnecessary data passing. In the majority of cases, the amount of extraneous information passed will be small enough as not to be problematic.

EPES can also be connected via OSC to any external system for gathering and processing input data. EPES currently includes preprogrammed input nodes for handling a variety of sensing systems such as the Kinect and webcam activity processing, but many existing systems also exist for processing this kind of input data in different ways. Many other kinds of complex input data processing systems can be incorporated into the overall system without having to develop a new type of input node, since the `GeneralOSCInputNode` can take any addresses and scale any range of inputs to 0.0 – 1.0.

One might ask whether it would be better to have a single mapping and generation system that handled the entire data pathway, from input signal gathering through producing the final output media such as sound or visuals. However, this would limit the pieces that could be created via this system. A mapping system that can communicate in a popular protocol with existing input and output systems will prove substantially more flexible with respect to the works that can be created, as different existing output systems are carefully designed for working with particular types of media.

Since EPES can communicate with any input or output sensing system, the overall computational demands can be split across multiple computers or locations as necessary. For example, in *Death and the Powers*, the mapping system ran on a separate computer than the computer handling the visual rendering to allow the greatest speed possible for complex visual renderings. In the global interactive simulcast of *Death and the Powers* (discussed in Chapter 6), the mapping system sent output control data not only to other computers and systems within the show, but via a server to hundreds of cell phones that each performed visual rendering given the control parameters.

5.4. Summary of the Expressive Performance Extension System

This chapter has presented the design and features of the Expressive Performance Analysis System and shown how it implements many of the principles for technological system design defined in Chapter 4. It has shown an example workflow for machine learning of expressive qualities in EPES: selecting expressive qualities, picking input sensors, selecting features for the system to compute, collecting training data, training a model, and using the trained model as part of mappings.

As a specific example of how this system could be used in practice to speed the development of extended performances, let us look back at the performance qualities used for *Death and the Powers*. In the Disembodied Performance System, the performer's movement is translated into a modified Laban Effort Space of *time*, *weight*, and *flow*. For *Powers*, that process of movement quality analysis was carried out completely manually, by hand-coding a `QualityAnalysisNode` that takes the accelerometer data and calculates values of the desired parameters based on empirical exploration of possible mathematical relationships. *Time* was associated with the amount of change of the accelerometer values, *flow* by accumulating the amount of change over time, and *weight* by scaling the summed accelerometer values (the amount of "energy" in the movement). It is important to note that these calculations were refined over the development of the nodes, requiring significant testing and experimentation with different values. Additionally, this process results in a specifically created analysis node that is not flexible enough to handle changes of sensor input easily.

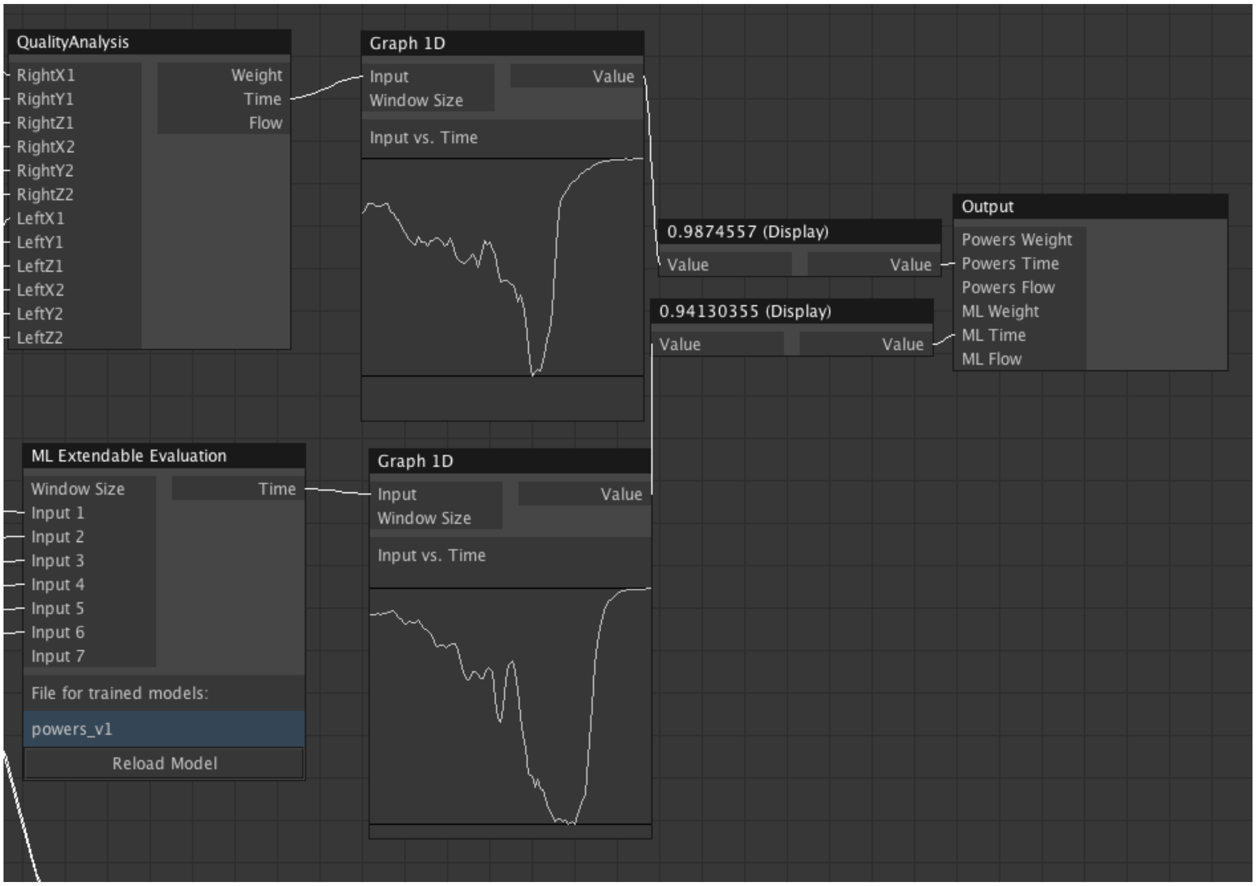


Figure 39. Comparison of Time parameters
Graphs show the continuous Time values generated by a hand-coded analysis node and a trained machine learning node

For comparison, a test was performed with EPES to train the system to recognize a subset of the same parameters used in Powers. The original wearable sensors and `DanceGlove` input devices used for Powers were hooked up to the `GloveFeatureComputation` node to determine an assortment of features of the combined input signals: *overall change*, *accumulated change*, *average change, overall value, derivative*, and *range*. Rapidly, the feature computation and machine learning nodes were set up, training data collected, and the resulting values compared with the hand-coded node. 9 training data examples were used, labeled with different desired values of *time*. 5 examples were labeled 1.0, representing still/very slow movement, and 4 examples were labeled 0.0, representing very rapid movement. These labels correspond to the very slow and very fast values produced by the `QualityAnalysisNode`. Eight of these examples were collected in the first data collection set, with another example of very slow movement captured after initial training and testing of the system when it was seen that the trained model was more accurate at capturing fast movement than slow movement. The length of these training data examples was normalized to half a second (15 frames at a frame rate of 30 frames per second).

A feedforward Neural Network was trained on the example data via the `MLExtendableTrainingNode`. This Neural Network used the default structure and number of nodes built into EPES, with one hidden layer consisting of 30 nodes. The size of this hidden layer had previously been determined through empirical exploration of the effects of different numbers of nodes in the hidden layer on output accuracy, generalizability, and speed of training. This network has an input dimensionality of 90, with the 6 inputs multiplied by the 15 frames of the normalized window size. The resulting trained system was able to interpolate from the examples to produce continuous output data that was quite similar to the values produced by the hand-coded system. Figure 39 shows graphs of the hand-coded value of *time* and the trained value of *time*, both calculated simultaneously on the same accelerometer input from two `DanceGlove` input devices.

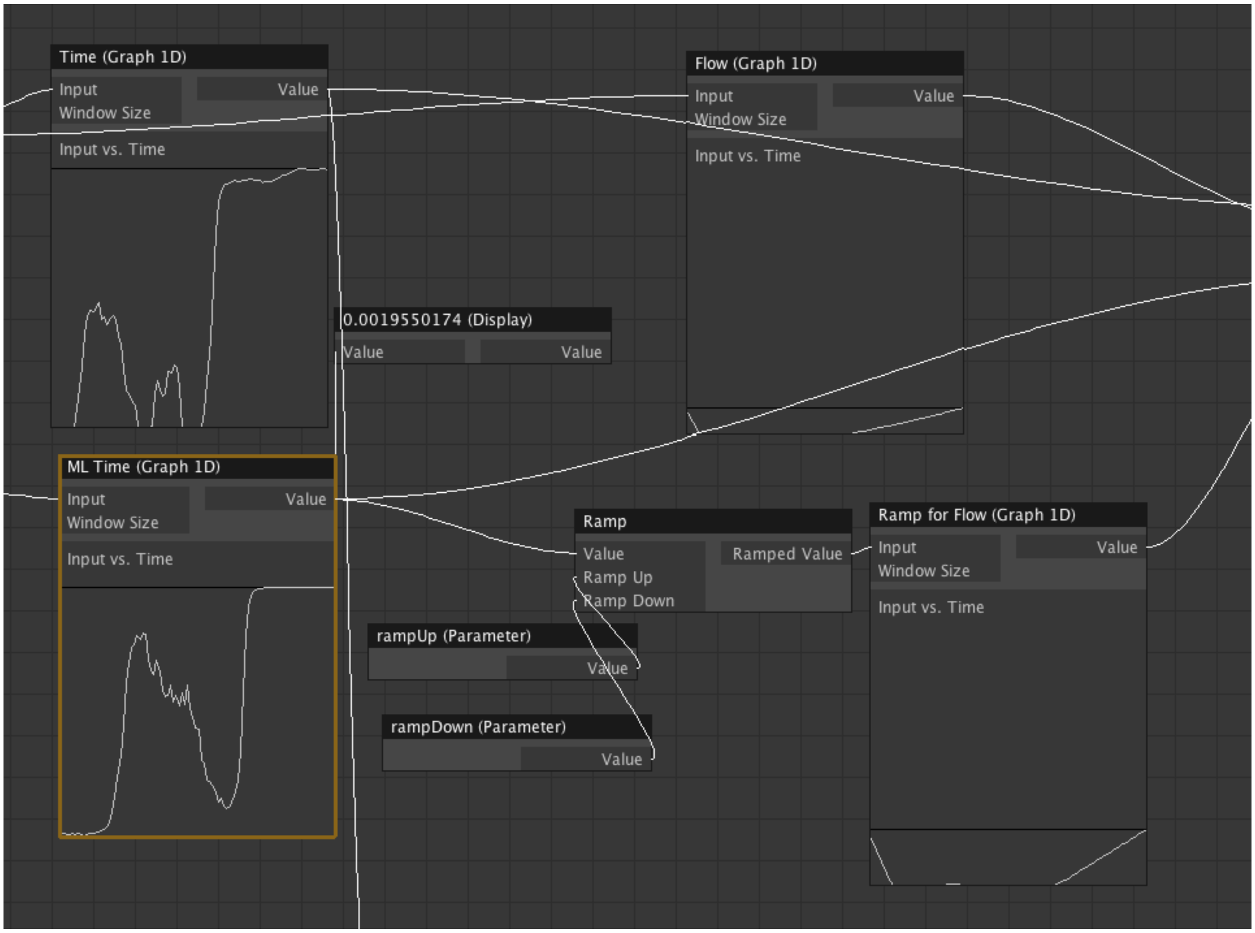


Figure 40. Comparison of Time and Flow parameters

As the measurement of *flow* was originally envisioned to reflect the change of the *time* parameter over a long scale of time, a quick replica was produced using the output of the trained `MLExtendableEvaluationNode` as input to a `RampNode`. The screenshot in Figure 40 shows the comparison of the *time* and *flow* values from the hand-coded `QualityAnalysisNode` (on the top half of the screen) with the *time* value calculated by the trained Neural Network and the *flow* value calculated from the trained *time* value using the `RampNode` for longer-scale temporal analysis of data.

Upon further examination of the original implementation of the *weight* parameter, the hand-coded computational relationship between the desired quality and the input data was found to be fairly weak. The value of the accelerometer axes varies more with their orientation in relationship to gravity than to the performer's actual movement, as would have been more desirable. Thus, a comparison between the hand-coded implementation and a trained version would not have been particularly meaningful. This likely explains why *weight* was not used in many mappings for *Powers*, with *time* and *flow* proving much more intuitively responsive.

Had we had the full EPES system while developing *Death and the Powers*, it would be interesting to see how our vocabulary of movement and vocal exploration could have been expanded. For example, while it was challenging to find a meaningful mathematical definition of *weight* for a hand-coded node, it would have been much simpler to give examples of strong and light movements. Initial training data could have be captured in the development process, and additional examples could have been added to the system in the course of rehearsal.

The following chapter will present a variety of larger-scale performances and installations that have incorporated EPES in their design and implementation. These include an installation based on expressive free movement, a public experience around the singing voice, and a set of performances extending body and voice. Discussion of these pieces and their development processes will show more specific examples of how this system integrates into live interactive contexts and allows expressive analysis of both movement and voice.

6. Primary Evaluation Projects

As part of my dissertation research, I incorporated the Expressive Performance Extension System into several performance and installation projects, exploring a variety of the behaviors of the system and its use in different vocal and physical contexts. Through discussion of these works, this chapter analyzes specific uses of the system in performance and installation contexts. It addresses the design goals of each experience, their development processes, the ways in which they incorporate the technologies and frameworks described in this dissertation, and the ways in which they illustrate previously discussed design principles for technical performance extension.

The first of these projects discussed is the Powers Sensor Chair, an interactive sonic installation where users can shape musical material from *Death and the Powers* through their expressive movements. The second project discussed is Vocal Vibrations, a public installation that encourages people to explore their singing voices and to have a novel experience of their voice in the form of vibration. The third project is *Crenulations and Excursions* and *Temporal Excursions*, a set of short solo performance and installation pieces exploring the voice and the body as expressive controllers for a soundscape. Other projects that have incorporated aspects of the Expressive Performance Extension System are also discussed, including: *Trajectories*, a multi-modal performance piece for eight actors and one narrator who gesturally manipulates sonic and visual elements; a variety of cross-disciplinary performances and experiences developed by participants in two Blikwisseling workshops in the Netherlands; and new visual and interactive content on mobile devices and an LED chandelier, developed for the February 2014 performances and global interactive simulcast of *Death and the Powers* in Dallas.

6.1 The Body: Powers Sensor Chair

6.1.1. Description

The Powers Sensor Chair was inspired by the original Sensor Chair designed by the Opera of the Future group for a project with the magicians Penn and Teller (Paradiso & Gershenfeld, 1997). That chair used capacitive sensing to detect the arm movement of a seated user, with sensors at the four corners of a frame detecting the position of the user's hands in relation to that frame. In this way, users could trigger sounds when they passed into the plane of the frame. One "percussive" mode divided the XY coordinate space into different sound zones, so that the location of the user's hand when crossing a particular threshold on the Z axis determined which particular sound to trigger. In another mode, the entrance of the user's hand into the Z plane of the frame triggered a sound, while the movement of the hand in the XY plane adjusted timbral coordinates of that sound. In a third mode, movement in the active space could influence the behavior of multiple notes at a time, dragging or guiding them through a frequency space.

For this re-envisioning of the Sensor Chair, we wanted to create an interactive installation where a participant could play with the sonic world of *Death and the Powers*, extending some of the movement capture technologies used in the live performance into experiences in which anyone could participate. This installation allowed visitors a special glimpse into *Death and the Powers* by giving them a new way to experience the auditory world of the opera, including vocal outbursts and murmurs, the sounds of the show's special Hyperinstruments, and rich spatialized textures.

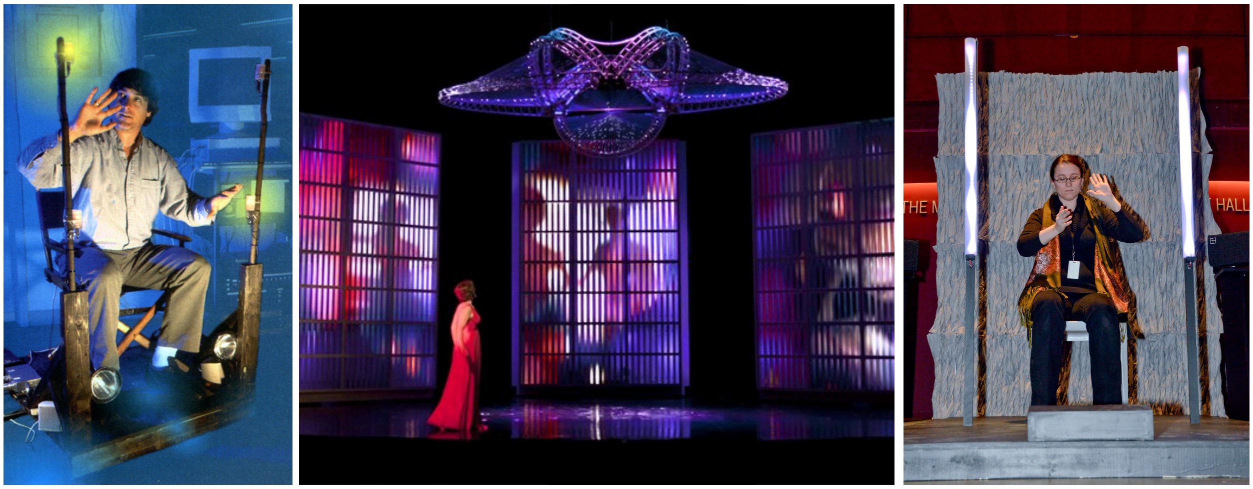


Figure 41. The original Sensor Chair and the Powers Sensor Chair
L: Joe Paradiso in the original Sensor Chair (photo from *Popular Science*). Center: the set of *Death and the Powers* (photo by Matt Chekowski). Right: the author in the Powers Sensor Chair (photo by Karen Almond).

In this installation, a solo participant sitting in a chair discovers that when she moves her hands and arms, the air in front of her becomes an instrument. With a small, delicate movement, a sharp and energetic thrust of her hand, or a smooth caress of the space around her, she can use her expressive movement to play with and sculpt a rich sound environment drawn from the opera. The sound surrounds her and the other visitors who become an audience for her performance.

Importantly, rather than using the sort of spatially-specific control models used in the original Sensor Chair, the new Powers Sensor Chair was designed with a focus on qualities of movement as the primary method of control. Similar to the aims of the Disembodied Performance System used in *Death and the Powers*, the goal of the Powers Sensor Chair is to augment a visitor's natural physical explorations, rather than to teach him a particular gestural vocabulary or a fixed and predetermined way of physically interacting with the experience. While this is a type of instrument, it is an instrument that allows each player to find his or her own way to play it.

In February 2014, the Powers Sensor Chair was played and experienced by a wide variety of audiences at the lobby of the Winspear Opera House, where it ran for around an hour and a half before each performance of *Death and the Powers* as well as for special Dallas Opera events. The chair was then transferred to the Perot Museum of Nature and Science, where it ran daily during the museum's standard open hours for two weeks. The Powers Sensor Chair then was installed in the Opera of the Future group space at the Media Lab, where it has been used by a variety of visitors during our sponsor week and other demos.

6.1.2. Technical Implementation and Mappings

The Powers Sensor Chair tracks the visitor's motion through a Kinect. Movement data is then processed to determine expressive qualities in the Expressive Performance Extension System. This information is mapped to control triggering of a variety of sound samples via MIDI, parameters shaping spatialization and dynamics of the soundscape, and parameters of the software program to shape the lighting patterns on the LED strips. The overall system diagram for the Powers Sensor Chair is shown in Figure 42.

Sensing System

Key requirements and features of the desired sensing system for this installation were:

- A user should not have to have any preparatory steps to be sensed. He should be able to sit in the chair and immediately begin.
- The sensor should be able to detect the motion of the hands, arms, and upper body.
- The sensing precision should work on both large arm movements and small hand and finger movements.
- The participant was known to have his movement and physical orientation confined within a known area.
- The sensing mechanism should be as invisible as possible to the participants.

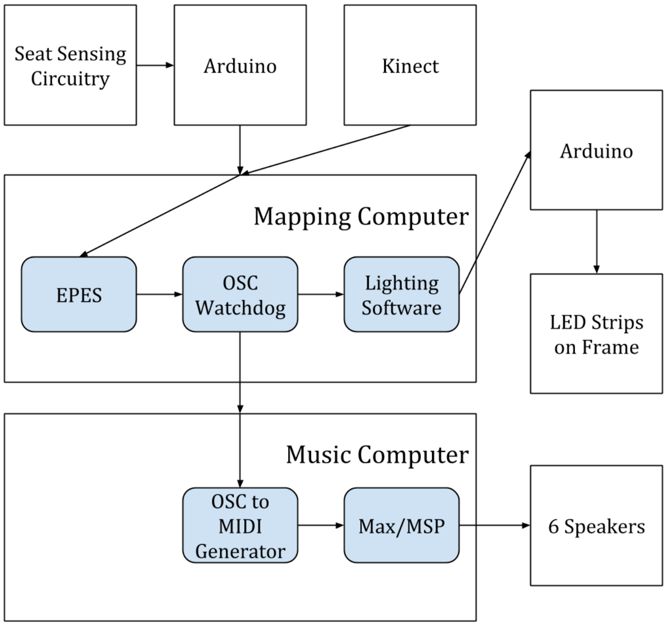


Figure 42. Powers Sensor Chair system diagram

In order to meet these sensing needs, a Microsoft Kinect was selected as the primary sensing mechanism for the Powers Sensor Chair. This Kinect is located on the floor approximately five feet in front of the chair, facing up toward the participant. An additional pressure sensor is located under one leg of the chair to detect when a participant is seated.

The `SimpleOpenNI` library for Processing was utilized in order to interact with the Kinect through Java. This library provides convenient wrappers and methods for getting access to the Kinect's webcam data, depth camera data, and higher-level processing such as user, skeleton, and hand-tracking data. A special input node, the `KinectInputNode`, was developed for EPES to process the Kinect data and output three categories of data. The first is hand tracking data for up to two hands, as obtained via the `SimpleOpenNI` libraries for Processing. The second is overall activity measurements generated via computer vision analysis of the Kinect's webcam, including an overall measurement of activity in a selected region of interest (specified in the input node and calibrated for a particular piece) and measurements of activity in four separate horizontal bands within that region of interest. For this installation, the observed region of interest in the webcam was calibrated to be bounded on the bottom by the seat of the chair (the user's lap), on the sides by the LED strips, and on the top to include a user's arms raised all the way up. The third feature comes from the Kinect's depth sensing camera, and is a measurement of what percentage of the pixels seen is closer to the Kinect than a depth threshold specified in the input node. These aspects of the data available via the Kinect were selected to provide a wide range of information and to be resilient to different kinds of movement.

The angle of the Kinect was set so as to maintain a successful rate of hand acquisition, as increasingly steeper angles (with the Kinect on the floor moved closer to the user) resulted in a lower speed and percentage of hand acquisition. Additionally, tests were performed comparing the hand-tracking results when the Kinect was located at an angle above the user pointing down to the user (as if it were mounted on a frame); these results were not as successful as when the Kinect was located on the floor pointing up to the user. Mounting the Kinect on an even plane with the user's arms had been removed as an option from the beginning, as that design would interfere with the ability of other audience members to observe a participant playing the chair as well as make the sensing mechanism too obvious to the user.

One challenge of working with the hand tracking information provided by the Kinect via `SimpleOpenNI` is its loss of information at high speeds of movement. The built-in hand tracking is quite accurate if the user is moving slowly or at a moderate speed, even with both hands being tracked. However, if the user waves her hands at very rapid speeds over a wide area (as might be reasonably expected in a standard interaction with the Sensor Chair), the hand-tracking algorithm often analyzes that movement as a series of new hand objects with different identifiers, rather than as a single hand object with one identifier and a series of new positions. Given this limitation of the sensor data, it was decided that the hand-tracking information should not be the sole data stream used for movement analysis. Additionally, when hands are rapidly lost and re-acquired, it is not straightforward to keep track of which hand is the "left hand" or the "right hand." Fortunately, the desired interaction paradigm did not need to differentiate between movements made with one hand and movements made with the other hand. The `KinectInputNode` was designed to always provide the current XYZ position of the two most recently identified hands (or most recent hand, if only one hand is being tracked). As spurious tracked hand IDs are generated during fast movement and then destroyed by the `SimpleOpenNI` libraries, the input node continually updates its currently active IDs to maintain the most up-to-date information.

Another challenge with the Kinect hand-tracking data was the occasional tendency of the system to identify an unrelated piece of background/object in the environment as a "hand." Since this object would not move out of the field of the camera, that "hand," once identified, would persist. In order for the Kinect input node to avoid getting stuck tracking one of these spurious 'hands," the input node stops tracking a particular hand ID if that hand's location has not changed in any dimension within a very small threshold for the past two seconds. Even if a user holds his hand very still, natural tiny movements generally are registered by the input node as enough movement to maintain connection with that hand.

The `KinectWebcamInput` device was extended to output an additional parameter, relating to the percentage of points in the image with a Z-depth lower than a given threshold (that is, closer to the sensor than a given threshold). This input device performs a mathematical rotation on the depth information from the sensor to compensate for the fact that the Kinect is placed on the floor in front of the user rather than on a horizontal axis with the user's hands. The raw data indicates that points at a given horizontal distance from the Kinect have different depth values depending on their height above the floor, which does not correspond to the user's perception. The transformation allows us to approximately compensate for this effect. In this manner, even if the user's precise hand positions are not currently being detected, it is possible to track the key distinction of whether the user has hands in the desired control range.

One additional sensor is used in this installation: a pressure sensor located under one of the legs of the Chair. This sensor's values are smoothed and thresholded to detect whether someone is sitting on the Chair or not; this information is provided as an input node to EPES. In the mapping system, this sensor can be used as a switch to determine whether the data from the other sensing strategies should be considered or rejected. This sensor has a threshold value that was easily adjusted as the chair was tested on a variety of participants, in order to obtain a value that would work for even the small children using the Chair at the Perot Museum. For very small children perching on the front of the seat, a museum guide could rest a hand on the back of the chair to maintain contact with the pressure sensor. Through this step, participants were encouraged to remain seated in the Chair if they wanted to continue controlling sound and lights. If they stood up, the interactive sound would fade out and return to a standard loop.

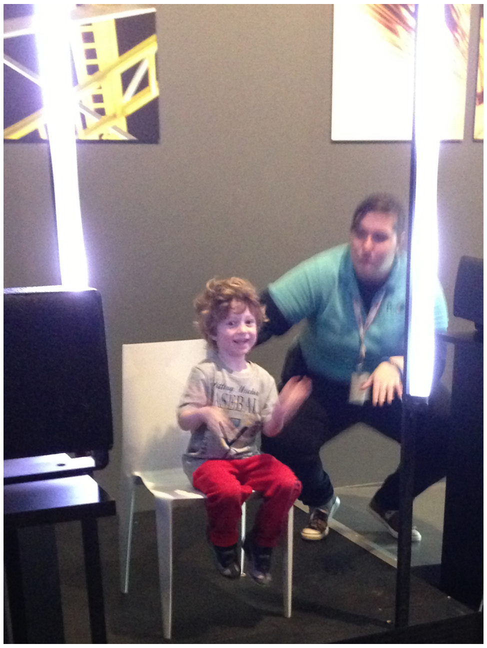


Figure 43. A young visitor in the Powers Sensor Chair

Feature Computation and Expressive Parameter Analysis

These hand tracking and webcam activity data streams are then processed via feature computation nodes to determine additional features across different timescales, such as their rates of change, derivatives, and smoothed values. In the EPES feature computation nodes used for this piece, the features chosen for computation were selected by hand and the calculation functions were hand-coded. The three different types of sensor data provided by the `KinectWebcamInput` device (activity in the selected region of the webcam, percentage of pixels over the specified Z threshold, and the XZY locations of each hand) are each processed to obtain the same temporal features over a half-second window, 15 frames at a frame rate of 30 frames per second. These features are defined in the `WebcamFeatureComputation` and `KinectTwoHandFeatureComputation` nodes, as discussed in Chapter 5: *overall change* (how much each input parameter has varied from frame to frame summed over the past window); *average change* (the average amount all the values have changed over the past window); *derivative change* (the amount of change over all input parameters in the past four frames, looking at a smaller window of time than the overall change value); *overall value* (a weighted average over the window of all parameters); and *accumulated change* (an accumulated metric of input variation that is incremented or decremented on each frame by an amount proportional to how much the inputs have been changing). A subset of these features are then used as direct inputs to the machine learning algorithms.

The features used as inputs for training expressive parameters included features of the points over the Z threshold (*overall change*, *average change*, *derivative change*), and of the hand-tracking information (*overall change*, *derivative change*). Some of the training samples were gathered when the Kinect was providing accurate hand-tracking information (when the hands were moving slowly and fluidly), while others were obtained while the Kinect was not correctly tracking (if the hands were moving very rapidly and being lost by the tracking system). This behavior of the system was deemed similar to the sensor values that would be measured on real participant data.

I selected three high-level expressive parameters to be learned by the system: *rate*, *energy*, and *fluidity*. Training data examples were captured separately for each of the parameters. The normalized length of the training data examples saved by the system was half a second. These training data examples primarily focused on different kinds of movements of the arms and were all performed by myself seated in the chair. I used a foot pedal to start and stop the system for capturing each example, since the user's feet are outside of the region of interest sensed by the webcam. In all cases, I made sure to have a static backdrop behind me so that any movement picked up by the system would be from my performance. This was consistent with the expected installation setting.

The final number of training data examples used were: 11 for *energy,* with 5 labeled 0.0 (calm) and 6 labeled 1.0 (energetic); 13 for *fluidity*, with 6 labeled 0.0 (fluid, smooth) and 7 labeled 1.0 (jerky, discontinuous); and 10 for *rate*, with 4 labeled 0.01 (very slow) and 6 labeled 1.0 (very fast). In the original training set for *rate*, the examples gathered included some labeled 0.0 (being completely still), and some labeled 1.0 (moving very rapidly). However, as I began to use this learned parameter in mappings, I discovered that I was not as interested in the range from no movement to very fast movement as I was interested in the range from very slow movement to very fast movement. I captured a new dataset to reflect this range.

These training data examples were captured with the Kinect positioned at an appropriate distance and angle from the chair, but before the platform for the installation was built and installed. The resulting trained values were found to be still accurate once the setup had been installed in the Winspear. Since the distance between the Kinect and the chair was predetermined and fixed, as was the angle of the Kinect, the sensor input was predicted to be comparable between locations.

Standard feedforward Neural Networks with sigmoid activation functions were used for performing regression on the labeled training data examples and live test input examples. These networks used the default Neural Network structure and settings defined in the Expressive Performance Extension System, with one hidden layer with 30 nodes. This number of nodes in the intermediate layer had been tested on a variety of vocal and physical inputs, and found to increase the accuracy of the trained results while not being so large as to make the system require too many iterations to train or not be generalizable. The Encog Neural Network's default number of nodes in the hidden layer (8) had been found to be too small to produce accurate results given a high-dimensional input. The dimensionality of the input in this context is the number of input data streams (5 inputs) multiplied by the normalized window length (15 frames). This results in a model with 75 nodes in the input layer, one hidden layer with 30 nodes, and a single node in the output layer. Three networks were trained, one for each expressive parameter. The output value for each network represents the predicted value of that parameter, and labels for training data represent the ideal output value for that parameter. These networks were then incorporated into `MLExtendableEvaluationNodes` for testing and mapping.

My process for testing each version of the trained network for a particular parameter focused on several aspects. For each test, I ran the evaluation system in real time for generally around 20-30 seconds (evaluating a new sample on each frame), providing examples that I believed to have quality values at various points along the parametric axis being evaluated. Could the trained system correctly output a high value for something I thought should have a high value? A low value for something with a low value? What about examples that I thought should be somewhere in between, how did it handle those examples? What about examples that I knew how I thought they should be labeled, but that were different than the examples I'd used to train the system? The accuracy of the system was judged in real time by observing the changing values of the expressive parameters via a `1DGraphNode`. I typically began by testing each parameter individually, then watching several graphs simultaneously to observe the composite results. As part of the testing process, once I had completed a few trial mappings, I brought in additional participants who were unfamiliar with the system to see how the system responded to different types of movement. I observed both the values that were output by the system and the sonic behavior of the system caused through mappings to these expressive parameters.

At each point in the testing, if I was not satisfied with the results of the training, I would add a training data example or two that seemed to represent the aspect of the quality that the system was not evaluating correctly, retrain the system, and repeat the testing. This process was generally repeated a few times for each parameter. Occasionally, an added example would cause the system's performance to degrade. In this case, the newly added example would be removed and a different example recorded in its place.

In the structure of the mappings, I used both the learned expressive parameters and other lower-level features. This combination of levels of analysis helped to compensate for the slight latency introduced by the machine learning. For example, a sudden change of movement from slow to fast would be expected to produce an audible result. Given the half-second training and analysis window, the system will not recognize this change in the *rate* parameter immediately. However, the *derivative change* feature can be used to pick up a particularly large change and trigger a layer of the system to respond immediately, with other layers coming in as the machine learning catches up, if the participant continues to move quickly.

Mappings: OSCtoMIDIGenerator

For the Sensor Chair, as well as *Crenulations and Excursions* and *Temporal Excursions* (discussed later in this chapter), the desired output of the mapping system was control parameters for a soundscape. These soundscapes were constructed from a variety of individual prerecorded sound samples. In the case of the Powers Sensor Chair, the samples were drawn from material used in and collected for *Death and the Powers*. In the case of *Crenulations and Excursions* and *Temporal Excursions*, this material was primarily excerpted from open source samples (both recordings and generated samples) collected from Freesound.org. These samples were mapped to MIDI keyboards, which are then controlled by an converter program I designed for generating MIDI notes parametrically. This program, the `OSCtoMIDIGenerator`, controls up to 16 channels of MIDI.

In the `OSCtoMIDIGenerator`, input control values have a different OSC address for each MIDI channel. The *note* message for a given channel specifies where along the keyboard for that channel the next note played should be located. Input values of 0.0f to 1.0f are mapped from the minimum to the maximum MIDI note desired for each keyboard; these values are specified individually for each channel in the `OSCtoMIDIGenerator`. Keyboards can be assembled with many different logical progressions along the keyboard. In a standard instrument, pitch may be the feature along which the keyboard is sorted. In other keyboards of samples, other arrangement mechanisms may be meaningful. For example, a set of vocal samples could be arranged from pure vocal tones to complex timbres, or from simple and slow melodies to complex and rapid melodies. The location of a current note along a keyboard is therefore an expressive piece of information.

Each MIDI channel in the `OSCtoMIDIGenerator` can be specified as either a continuously playing keyboard or a triggered keyboard. In a continuously playing channel, an input *rate* parameter determine how rapidly notes should be selected from that MIDI keyboard. The OSC input messages for rate are mapped from 0.0-1.0 to a number of frames between notes. This mapping range of frames is currently implemented to be the same for all MIDI channels. At each analysis frame in the program, the system checks the time at which the most recent note in a given keyboard was played, and the current value for the desired number of frames between notes. If the current time is later than the specified delay, a new note will be played that frame using the current note value. In a triggered keyboard, the system keeps track of the previous input values for rate. An OSC input rate value change to 1.0 will cause the current note value to be emitted immediately on the current frame. A minimum time between triggers can also be set, so as to prevent overly rapid triggering of notes on a signal that has not been de-bounced and thus fluctuates back and forth around 1.0. These triggered keyboards can be used for generating notes precisely with a particular behavior (when a movement's energy crossed a particular threshold, for example).

The `OSCtoMIDIGenerator` also has parameters for controlling the volume of each note played. As with the note value, this value for volume for every channel is updated on each received OSC message, but only looked up when a note is triggered or the delay length between continuous notes has passed. This system has generally been used with samples set up to play the entire sample given a Note On MIDI message, but the `OSCtoMIDIGenerator` also incorporates a *duration* parameter that reflects how long in seconds a particular note should be held before a corresponding MIDI Note Off message is sent.

In this way, the `OSCtoMIDIGenerator` creates MIDI messages based on shaping and triggering information it receives via OSC. In the Powers Sensor Chair implementation, these MIDI messages are then sent to Max/MSP where they are used to control a virtual MIDI instrument. The different channels of this instrument each consist of a different type of Powers samples arranged with a particular mapping from low to high. One keyboard is entirely samples of James Maddalena singing the word "more" on different melodic patterns, building from the simplest single notes to the fastest and most complex melodies. In *Death and the Powers*, these samples are combined into longer multi-layered triggers used in Scene 4, the love scene between Powers' wife Evvy and Powers as the Chandelier. Another keyboard consists of Chandelier sounds from Scene 4, building from low drones to multilayered textures to sharp string "bongs." Another keyboard is constructed of bits of samples from the Prologue and the Epilogue, the high twinkling music accompanying the robots.

Mappings: LED Control Program

The original Sensor Chair was designed with light bulbs at the four corners of the sensing space, whose intensity varied with the proximity of the performer or participant's hand. This served as feedback about the location of the user's hand. Since the Powers Sensor Chair was not designed to use specific location sensing, but instead expressive qualities of movement, I determined that the kind of feedback provided by lights on either side should similarly be qualitative rather than positional. The lights do not imply to the user that he should put his hand in specific places, but instead help give feedback on his levels of energy and rate of movement.

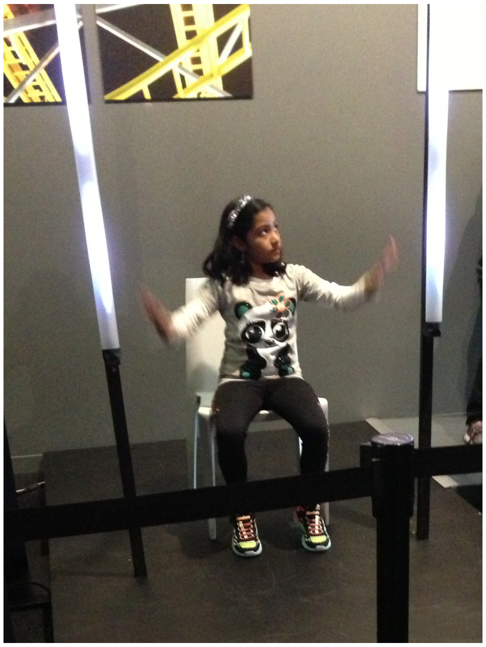


Figure 44. A visitor observes the lighting patterns on the Powers Sensor Chair

The Powers Sensor Chair incorporates two LED columns, one on either side of the active sensing area. Each of these columns consists of a frosted acrylic tube with a strip of 60 individually addressable RGB LEDs, running according to the NeoPixel specifications ("The Magic of NeoPixels | Adafruit NeoPixel Überguide | Adafruit Learning System," n.d.). The basic behavior of the LEDs is a point or points of light moving in a sinusoidal pattern up and down the tubes. I developed two programs to create and control these sinusoidal patterns of light in the tubes.

An Arduino Mega running custom software controls the behavior of these points of light and calculates and sets the intensity of each pixel at each moment, given parameters to shape the desired sinusoidal patterns. The program on the Mega adjusts several control variables of the lights, including their peak brightness (*intensity*), the speed of movement of the point along the sinusoidal path (*rate*), the number of moving points along the whole LED strip (*density*), and the "width" of each point (*size*). Width is a measurement of how many LEDs a "point" occupies: the central LED is the current location of the traveling point, and the included LEDs to either side of center decrease linearly in intensity. All of these parameters have been designed to be continuously variable except for *density*, which is translated to a discrete number of points from 1 to 10. An additional application written in Java translates Open Sound Control messages into serial control commands for the Arduino Mega program. In the mappings used for this installation, the lights speed up, grow brighter, and grow more complex (consisting of more points and smaller points) as the user becomes more energetic in her motions and maintains that level of energy over time. Brightness is also used to respond to particularly high-intensity movements.

Backdrop and Environment

Behind the Sensor Chair, a backdrop made of twisted tissue paper was set up in the Winspear Lobby to provide a visual focus for audiences watching a participant play with the installation, to help provide a consistent backdrop for the camera sensing (so that others passing behind the installation would not confuse the signal), and to help focus sound around the participant in the chair. The design of this backdrop was inspired by that the paper backdrop used for the *Crenulations and Excursions* installation, combined with the appearance of the LED walls used for *Death and the Powers*. When the Sensor Chair was moved to the Perot, the platform was located against a wall in the space, eliminating several of the challenges for which the tissue paper backdrop was originally developed. In addition, in order to properly install a wheelchair accessible ramp in the Perot, it was necessary to remove the tissue paper backdrop.

In order to make the Chair accessible to those in wheelchairs, as desired for both the Perot and the Winspear, additional mappings were created that were identical to the primary mappings but did not incorporate the use of the under-chair sensor. The Chair itself is not affixed to the platform but instead sits in holes cut in the shape of the legs. For the Sensor Chair setup to be adjusted for a handicapped participant, the only steps are to temporarily remove the physical chair, move the participant's wheelchair into the active sensing space, and switch mapping modes.

6.1.3. Analysis: High-Level Qualities and Participant Experience

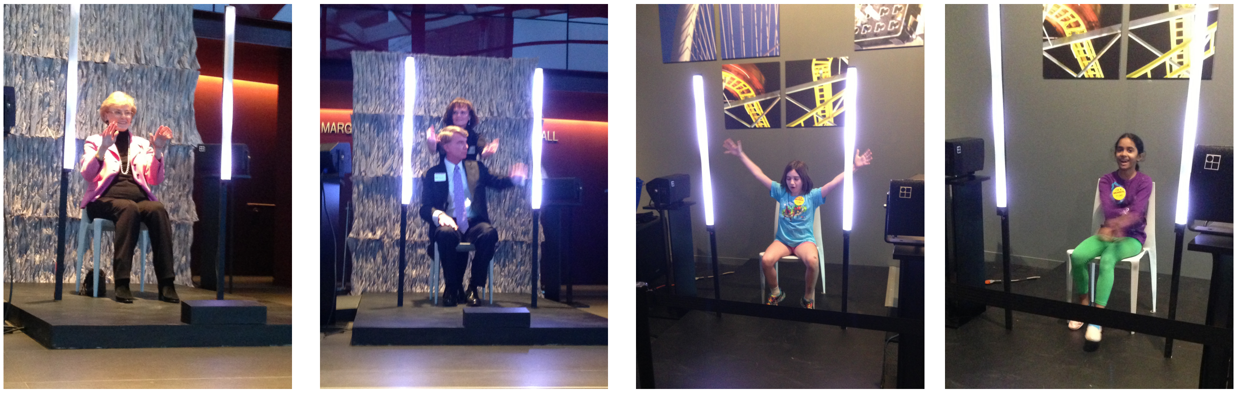


Figure 45. Powers Sensor Chair visitors at the Winspear and Perot
A variety of different people played the Powers Sensor Chair in its two installation locations in Dallas, the lobby of the Winspear Opera House and the Perot Museum of Nature and Science.

The user populations at the two locations where the Sensor Chair was presented were quite different; the Winspear audiences were primarily adults (many of whom were interested in the fusion of technology and art), while the Perot audiences were primarily children and their families. In order to design a piece that would work similarly well in both locations, it was necessary to create an installation with a "low floor, high ceiling." That is, it was necessary for a novice (perhaps a child just waving their arms rapidly) to immediately have a sense of control of the experience, a direct connection between his movements and the sound he was experiencing. However, it was also necessary to provide more sophisticated levels of control available for those who wanted to take a little bit of time to explore the installation and try a variety of kinds of movement.

One aspect of the Sensor Chair that was particularly interesting to observe was the wide range of movement vocabularies that different participants used to engage with the chair. Since there were no particular instructions specified by the exhibit or by the staff (besides general instructions such as "try moving your hands"), visitors were free to experiment with whatever movement came to mind. I believe this flexibility of movement vocabularies for interaction was supported by the fact that the mappings for the Sensor Chair focused on qualities of movement rather than on gesture recognition or on position-based information. The mappings did not require that a visitor used the same gestural vocabulary in the chair as I used while designing the interaction, or that a visitor learn a particular mapping of space to sound; instead, participants could use their own natural movements in dialogue with the Chair's sonic design. Participants were freed to explore and be comfortable with the installation because it was clear that there was not one "right way," one obvious vocabulary of interaction. The chair did something interesting no matter what they did. They could experiment with a variety of different behaviors to see how the chair responded to them, but they were not constrained by trying to learn the interface.

An unexpected aspect of the limited set of instructions given to Sensor Chair visitors showed up in the range of activities participants experimented with beyond moving their arms and hands. Some visitors tried playing the chair by moving their feet, leaning side to side, or moving their head into the active zone. Quite interestingly, some participants explored the use of the Chair as a duet instrument, despite the form factor of a single chair indicating a solo experience. Couples sat together on the Chair and attempted to coordinate their movements. In the Perot, parents sat with young children on their laps (I observed some with children as young as one or two years old) and guided their hands and arms. This unanticipated use case was also made possible by the design of the system not limiting all the movement analysis to a specific body. As long as at least one person was sitting on the Chair, the sensing is activated.

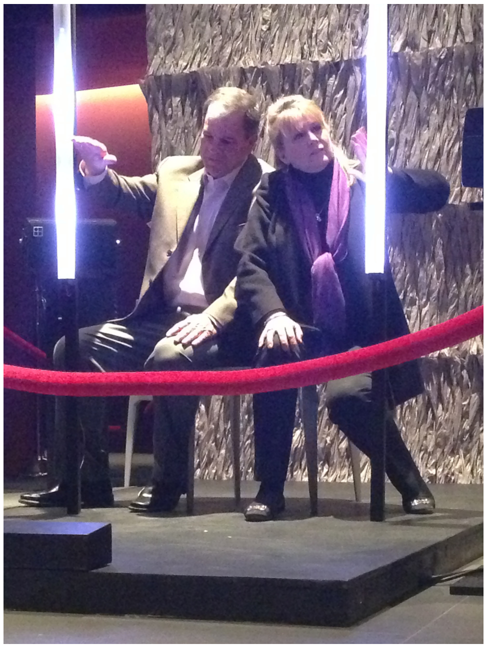


Figure 46. A couple plays a duet on the Powers Sensor Chair

While the sonic landscape provided immediate feedback for participants, the lighting on either side of the Chair also served as valuable feedback. This visual element appeared to be one of the primary ways that people could immediately tell they were having an effect on the space. Additionally, I observed an interesting interplay between the movement of the lighting and participants' movement explorations. The sinusoidal lighting patterns were designed to never go completely still or dark regardless of whether the participant was engaging with the instrument or not. I suspect that the continuous spatial movement in the lighting gave a sense of motion to participants and suggested that they should be moving as well. Indeed, the qualities of participants' movements were also occasionally influenced by the qualities of the visible lighting. I observed some participants pacing their motions to the speed of the lights along the LED tubes, and other participants consciously or unconsciously mimicking the sinusoidal patterns by moving their arms up and down. Some users were not sure whether the LED tubes were the movement sensors; generally, participants did not detect the Kinect positioned on the floor in front of them.

The Powers Sensor Chair is also interesting to consider through Benford's framework of expected, sensed, and desired actions. For example, a limitation of the sensing methods of the Powers Sensor Chair is the precision of the Kinect hand tracking algorithms when the user is moving his or her arms very quickly with large movements. As such movements are certainly both expected and desired, it is necessary to design the interface to compensate. For instance, some of my original experimental mappings used the built-in Kinect hand tracking as the sole method of determining whether a user's hand was in the correct range to trigger notes. Given the range of expected and desired movements that go outside the bounds of the sensing capabilities, later designs incorporated raw depth-tracking information to provide backup, less precise sensing capabilities.

An important point brought up by Benford is the opportunity to use expected but not sensed movements to allow for users to rest. In the context of the Sensor Chair, we are not tracking hand movements that a user performs behind the frame with lights, only those at or in front of the frame. This is not a limitation of the sensor setup, but an intentional choice to allow users a way to have their movement not be sensed. In a musical context, it is very important for users to have moments of silence and control over when they are playing or not. While holding one's hands completely still in the tracking field causes the sound design to become quiet, this does not allow the user a break. The ability to sit quietly in the chair without the majority of movement being sensed proves a useful way for participants to pace themselves in their explorations.

One thing that was especially interesting to note in my observations of Powers Sensor Chair visitors was the degree to which people were amazed that their movement could have a sonic effect. To some extent, this was a result that surprised me. We live in a world where technology is omnipresent. People are constantly interacting with their cell phones and computer screens. The Kinect and Wii let video games use the body as a controller. Speech recognition techniques have gotten increasingly accurate. Even small children, like the visitors at the Perot Museum, are exceedingly familiar with the power of technologies and expect to be able to do things easily via a computer or handheld device. And yet, when participants sit in the Sensor Chair and the first vocal sample plays, the first reaction of participants is usually one of astonishment, followed by even greater astonishment upon moving for the first time and finding the system react to that motion. Seeing that the sonic behavior and the lighting of the chair are not autonomous but actually has a connection to one's own actions seems to still be a surprising experience for the majority of participants. This interface seems to create a sense of magic in its interaction. I suspect this experience is partially due to the interface using free movement, rather than requiring interaction with a mouse or touchscreen. Additionally, its feedback comes in the form of auditory and abstract lighting, rather than using any visualization of the participant's body as in a Kinect-based video game. Finally, there are few interfaces that people interact with on a regular basis that do not have a set vocabulary of interaction or a particular goal. I hypothesize that a system that provided access to an immersive, interactive experience, but did not dictate the form of interaction with it, proved particularly compelling.

To conclude, the Powers Sensor Chair demonstrates a few key features:

- The chair focused on control mappings using expressive qualities, rather than specific gestures or physical positions in space.
- A qualitative parametric model of physical expression supported interesting results with many different users' vocabularies of movement.
- Participants were comfortable exploring the installation, perhaps because it was clear that there was not one "right way" to use it or one obvious vocabulary of interaction.
- Use of both trained parameters and computed features in mappings helped to address any latency introduced by Neural Network evaluation.
- The installation systems ran smoothly while on display for several weeks and used by hundreds of visitors.

6.2. The Voice: Vocal Vibrations

6.2.1. The Vocal Vibrations Initiative

In the Opera of the Future group, we are currently seeking to expand our work in technologies for sophisticated measurement and extension of the singing voice in performance to create new kinds of vocal experiences in which everybody can participate. Many people are not comfortable "singing" or do not feel that they can use their voice to become part of a rich musical experience. To address this, we are developing techniques to engage the public in the regular practice of thoughtful singing and vocalizing, both as an individual experience and as part of a community. In addition, we are exploring the ways that the singing voice can affect the body and mind, and how the act of focused vocalization can build contemplative practice, concentration, and listening skills.

Since the summer of 2012, we have been exploring these topics through the Vocal Vibrations initiative. As part of this initiative, we launched the first Vocal Vibrations public installation in March 2014, commissioned by art-science lab Le Laboratoire in Paris. The Opera of the Future team involved in this project consisted of Tod Machover, myself, Rebecca Kleinberger, and Charles Holbrow. We sought to create a space where people could come and explore their voices, both through a public space for careful listening and through a solo interactive vocal experience that used multiple sensory modalities to help the user explore their own voice and the vibrations created by their voice.

My role in this project was as a primary interaction designer of this installation, defining a meaningful set of expressive vocal parameters and shaping the resulting behavior of the system. The solo interactive portion of the installation incorporates the Expressive Performance Extension System, which analyzes and recognizes expressive parameters of a user's voice and uses that information to shape the user's experience, primarily through vibration in a handheld device that we called the Orb.

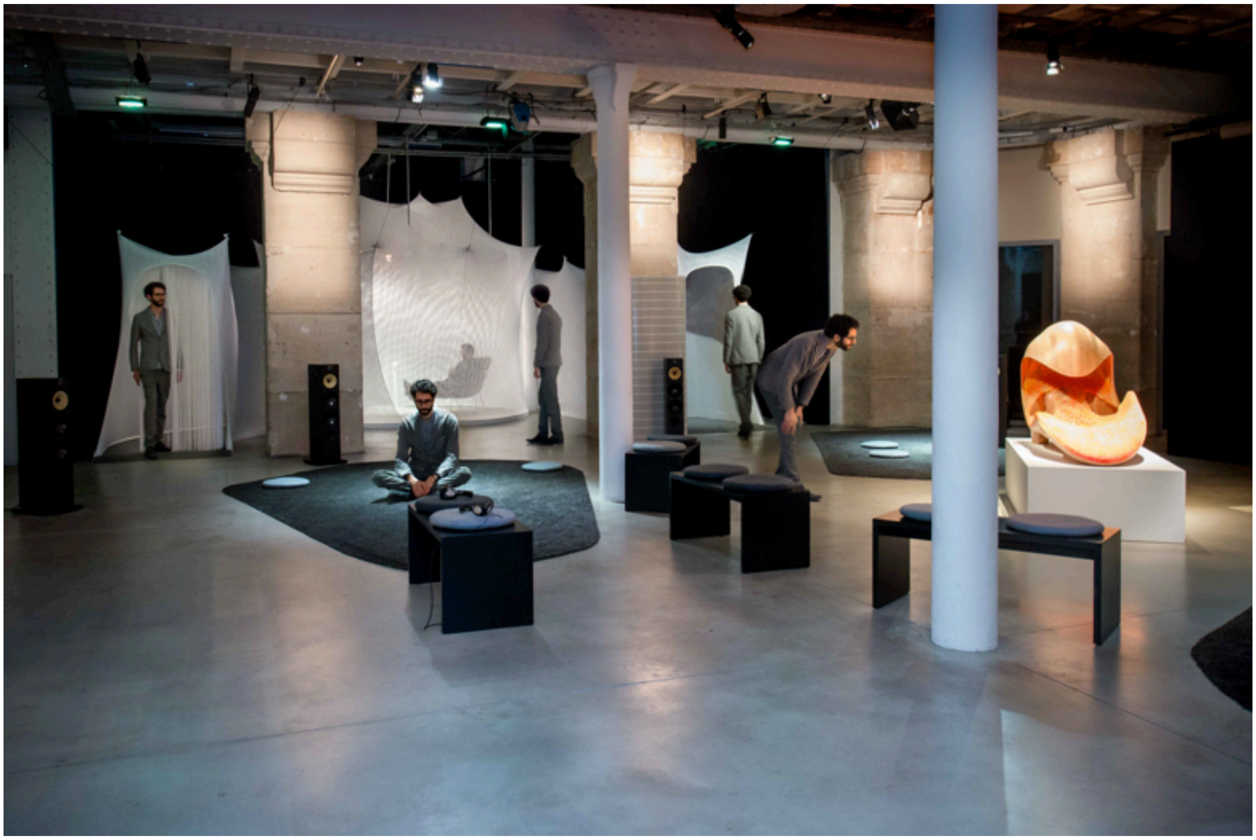


Figure 47. Vocal Vibrations installation at Le Laboratoire
The Vocal Vibrations space can be experienced in many ways. Image by Bold Design.

6.2.2. Components of the Vocal Vibrations Installation

The Chapel

Figure 48. Visitors to the Vocal Vibrations Chapel
The Chapel provides a space for careful listening to music together with other visitors. Image by Le Laboratoire.

When installation visitors arrive at Le Laboratoire, they first enter a public space, which we call the "Chapel," designed to encourage careful and meditative listening. In the Chapel, 10 Bowers and Wilkins high-fidelity speakers are located around the space playing a spatialized composition by Tod Machover constructed from recordings of an early music choral ensemble, solo soprano vocal explorations, and Tuvan throat singers (Machover, 2014a). The composition in the Chapel centers around one particular pitch, a D, which participants are encouraged to follow. At any point, singing a D will fit into the composition. Headphones located on benches in the space play voices singing the D in different octaves to help participants hear the pitch and locate it in the larger composition.

Also on display in the Chapel space is the *Gemini* chaise designed by Neri Oxman. This piece explores different acoustic and resonant properties of materials, with the eventual goal of constructing a personal space for singing that could modify the acoustic properties of one's voice and the sound in the surrounding space. *Gemini* is constructed from resonant wood and sound-dampening 3D printed shapes in 40 different material combinations, with varying stiffnesses, opacities, and colors.

The Cocoon

After a visitor has spent time in the Chapel, an assistant leads her to a solo experience in an isolated environment within the installation, the "Cocoon." The private Cocoon environment guides an individual to explore his or her voice and its vibrations, augmented by tactile and acoustic stimuli. In the Cocoon, the visitor is given headphones, a headset microphone, and a small vibrating Orb to hold. The assistant instructs her to sing the D, be guided by everything she hears in the headphones and follow what she hears with her voice, and see how the vibrations in the Orb change with her voice. She is then left alone in the Cocoon to have a solo vocal experience, while a six-minute pre-composed soundtrack by Tod Machover plays in the headphones (Machover, 2014b). In this environment, we seek to encourage visitors to experiment and play with their voices, as well as to gain new understanding of their voices and the vibrations produced in their body.

The Orb

As we worked to develop the Vocal Vibrations installation, it became clear that we needed something to enhance awareness of the vibrations in the body caused by singing. We originally began with the idea that a participant in the installation would sit in a chair that was enhanced with vibratory properties that could be mapped to respond to the qualities of the voice. We tried several experiments with the effects of transducers touching various points on the body. We also explored the tactile effects of series of vibratory motors hooked up to respond to vocal parameters. However, few of the experiments on the body were fully compelling; rather than enhancing the sensation of one's own vocal vibration, it was challenging to create an effect that did not simply feel like a massage chair.

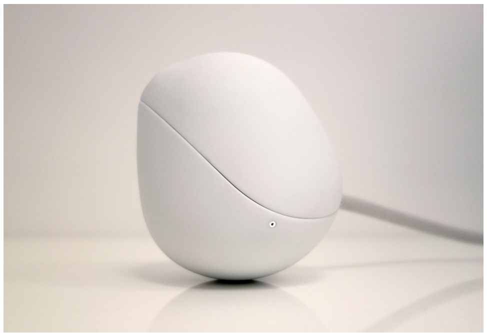


Figure 49. The Orb
Five transducers glued to the inside of the ceramic Orb allow vibration patterns to be generated across the object's surface. Photo by Bold Design.

Through these experiments, we found that transducers playing the raw vocal signal were most compelling when touched with the hands and fingertips. The hands are one of the most sensitive parts of the body, with many closely spaced nerve endings for detecting vibration (Gunther, 2001). We found that the hands could detect many variations in vibration caused by amplitude, frequency, and timbre. Given these results, we decided to develop a device that could be held in the hands that would vibrate with a participant's voice and give them an awareness of the variation of vibration contained in their voice.

The prototype version of this device was developed in Fall 2013, and shown at the Media Lab sponsor meeting. We used a hollow glass sphere as the base for the device and attached five transducers to the interior, one on the top and four around the sides. Each transducer could receive a different signal. After the transducers were affixed to the inside, the remainder of the sphere was stuffed with wool and polyfill batting, so as to hold the transducers firmly in place against the curved interior surface and avoid the vibration of the transducers causing them to detach from the surface.

The data analysis that we used for this version had separate programs for initial feature computation and for extended feature computation and expressive analysis. One program (developed by Rebecca Kleinberger) captured basic vocal analysis parameters such as loudness, frequency, harmonicity, and noisiness from a live microphone stream. These parameters were then sent to the Expressive Performance Extension System, which performed additional layers of computational analysis at multiple timescales, looking at smoothed harmonicity and loudness over time, overall change of the signal, derivative change, and average amount of change. Through these values, we focused on the overall stability of the input signal. These analysis results were then mapped in EPES to the input parameters of a Max/MSP patch for controlling the Orb: should the sound move from transducer to transducer, how quickly, in how scattered of a pattern, how prominent additional sounds should be, how fast should the Orb's additional pulses of sound be? The Max/MSP patch used the raw vocal signal and this control data from EPES to determine what sound patterns to send to the Orb's transducers.

We explored many variations of mappings from input sound parameters to behaviors of the vibration in the Orb. Initial experiments showed that even simply sending the pure vocal signal to the Orb was quite interesting, as very small changes in the voice caused completely connected transformations in the experienced vibration. And yet, due to the modality transformation, the experience was not simply that of touching a loudspeaker. The surface and material of the Orb affected the ways that the vibration patterns interacted and were amplified or dampened, such that different frequencies of the voice or different amplitude patterns began to take on their own behavior.

We also experimented with different patterns of moving sound around the Orb, either from the direct audio signal or from a generated wave or burst pattern. Due to the resonance of the Orb's material, simply playing sound from one transducer versus a different transducer was not particularly noticeable; for example, it was difficult to distinguish tactilely whether a sound was being sent to the left-most transducer, the right-most transducer, or the top transducer. Smoothly shifting a sound from one transducer to another was also hard to distinguish by touch. In order to have a sense of spatial variation in the vibration, we found it necessary to use quick pulses of sound that could be rotated or bounced around the orb from one transducer to another.

The mapping that proved most compelling in this initial version used a blend of a pure vocal signal sent to the top transducer and a pulse that could be brought in to rotate around the other transducers. We sought to emphasize the simplicity or complexity of the user's voice in an unexpected direction, by rewarding a pure, extended tone with a growing complexity of the vibration experience. Speaking or making other complex sounds caused the orb to fall back to outputting the pure vocal signal. If a pure tone was held for long enough, the mapping brought up the intensity and speed of the moving pulse, building from a tiny shudder to a complex shake. Meanwhile, having a layer of the raw vocal signal incorporated into the Orb's top transducer kept the results feeling immediately connected to changes of input.

The Orb used for the final installation was designed in collaboration with the French company Bold Design. This Orb is made from ceramic, with two pieces held together by small screws. Due to its asymmetry, it turns out to have some interesting vibrational properties. When the transducers are installed, input signals of different frequencies propagate with different decay rates across different parts of the Orb. Low signals resonate more in the rounded end, higher signals resonate more in the pointed end. A frequency sweep or a signal that is changing in frequency rapidly results in the impression of the signal moving from end to end of the Orb.

Musical Material and Simple Vocal Interactions

Since a major goal of this installation was to give novices the ability to be part of a musical experience centered around their voice, we explored what kind of musical tasks we could give someone that would be simple, not intimidating, and still rich enough to fit into a musical experience. We determined that one of the easiest "gateways" into a musical experience would be simply asking participants to sing only one note, to experiment and explore variations on a single pitch. They could play with rhythm, with vowels, with timbres, with positioning the sound in different places in their head and body. They would not have to learn a part beforehand, they could be taught their basic part quickly and easily. The musical experience around them could then be composed such that this single note would always fit into the composition.

The sonic material gathered for the Vocal Vibrations installation primarily consists of vocal material recorded from a choral group specializing in early music (Blue Heron) and an operatically trained soprano (Sara Heaton). Other material was recorded from the Tuvan throat-singing ensemble Alash. Over a set of recording sessions, we captured a wide range of raw material for the installation, varying from musical material pre-composed by Tod Machover to improvisations in a specific timbre. With Blue Heron, we recorded a number of chords and melodies sung with a very pure tone on a variety of vowels, as well as with improvised "morphing" timbres. With Sara Heaton, we recorded a wider variety of material: individual notes in the key of the piece, with a variety of timbres and articulations; pre-composed melodic fragments with multiple and shifting timbres; longer melodies; improvised passages of morphing timbres; improvisations with continually gliding and shifting pitches; even explorations of whispered sounds and unvoiced sounds. The majority of this material was composed to center on a D. Through these recording sessions, we sought to capture material that could be edited into short fragments. Those fragments then served as the building blocks of both the individual interactive experience and the longer composition for the Chapel.

6.2.3. Development of the Cocoon Solo Experience

I will primarily focus on this installation's individual, interactive experience, the Cocoon, as this is the aspect of the Vocal Vibrations installation that incorporates the Expressive Performance Extension System. Additionally, the Cocoon is particularly interesting in the extent to which our conception of the interactive experience changed frequently throughout the development process.

EPES supported rapid prototyping of a variety of different interactive models as we sought to determine what interaction design would support our overall goals for the experience.

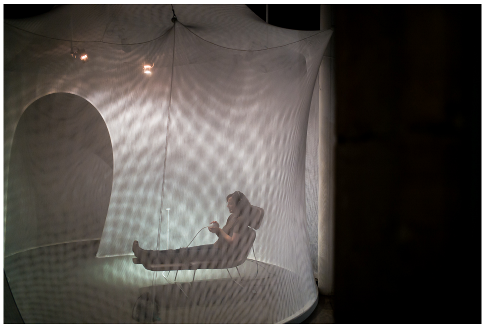


Figure 50. The Vocal Vibrations Cocoon
Participants entered the Cocoon space to have a solo vocal experience with the Orb. Photo via Le Laboratoire.

Our goals for the Cocoon experience were multiple. First, we wanted to give people a new way to explore singing and vocalizing, to draw those who are not usually singers into a musical experience through their voices. Second, we hoped to guide people to use their voices as a tool for careful listening to music. The third goal was to draw participants' attention to the way in which the act of vocalizing creates vibrations in their bodies, and to enhance that vibration and their awareness of voice as a tactile experience. As we worked to realize the individual Cocoon experience, we continued shaping these goals and tried to use them as guidelines. We knew that the interactive technologies and techniques we were using needed to support these goals and not become a distraction from them.

Perhaps more so than in most other projects, our ideas about how this individual Cocoon experience would work and what the interactivity would look like shifted substantially between our early conceptions and our final realization. The flexibility of EPES was found to be quite useful as we explored a variety of different ideas. While we were rapidly prototyping different kinds of mappings and incorporating different input information and output models, we did not have to change the technological core of the experience. The data pathway remained the same throughout a variety of different interaction models, allowing us to flexibly experiment with many different ideas. The nature and purpose of the mappings changed substantially throughout the development process, but the core of the system did not.

Our basic signal flow remained the same throughout the design process. A Max/MSP patch (designed by Rebecca Kleinberger) first calculates computational features of the voice, given a microphone input. These computational features are sent via OSC to an input node in EPES, which handles additional feature computation (such as the variation of features over time), adds higher-level trained parameters, and allows the creation of mappings to output control parameters. These mappings can incorporate both low-level computational features and high-level abstract parameters. The output control parameters are sent via OSC to another Max/MSP patch that controls output sound for the headphones and the behaviors of the Orb. The input Max patch for vocal feature computation could mostly have been replaced with an extension of EPES's existing vocal processing input nodes (which currently analyze fundamental frequency, harmonicity, and amplitude from a microphone signal), but it was decided to keep the analysis of computational features in a separate system for ease of development with multiple collaborators.

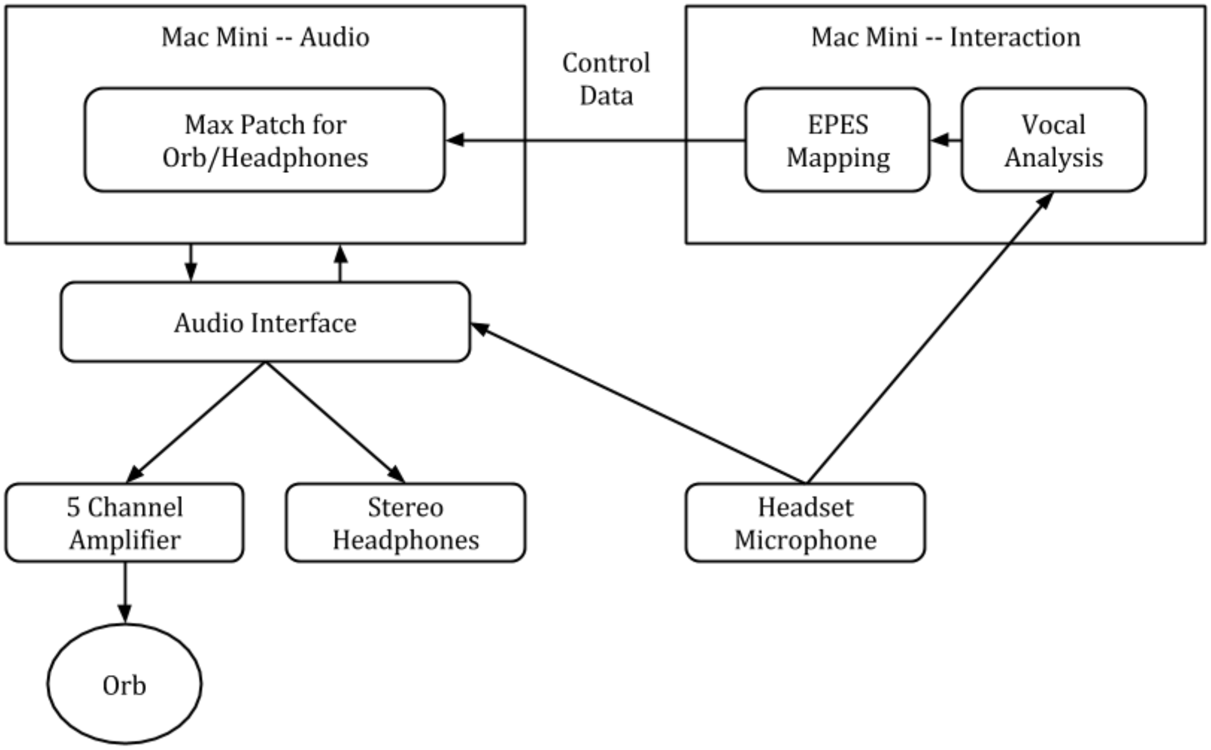


Figure 51. System diagram for the Cocoon experience
The system flow of the Cocoon remained constant even as we explored many variations on the shape of the experience and the nature of a participant's vocal interaction.

We originally intended to combine a pre-determined shape of the experience, which could serve as a guideline for the user, with moment-to-moment interactive sound generation. A pre-composed baseline track could serve as the core and throughline of the installation, with different kinds of sound at different moments helping to sculpt the participant's experience. On top of that stable core, we could layer fragments of material (including melodic snippets, additional notes, and different timbres) that would be added in response to the participant's vocal qualities. By using different mappings at different points in the experience (easily changed at times synchronized with the playback of the core track), we could make different vocabularies of interaction accessible at different moments. However, as the development of this initial Vocal Vibrations installation progressed, we chose to simplify the interaction design substantially to draw closer to the goals of the experience.

An important question in the Vocal Vibrations solo portion of the installation was the balance between how much the system and musical content should be in reaction to the participant's vocalizations, and how much the musical content should guide the participant's vocalizations. In an earlier design for the experience, we had considered using brief text phrases displayed in front of a participant to suggest how she might vocalize in relationship to what she was hearing ("like this," "something unexpected," etc.). As the design of the Cocoon space evolved, it was clear that displaying text would not be a good option, so we discussed giving a participant instructions at the beginning telling them to vocally follow what they heard in the experience. However, the danger with having this instruction in an interactive system was the prospect of the participant and the system ending up in a feedback loop, where the user would follow the sound qualities suggested by the system, which would then further enhance those sound qualities, which the user would then repeat.

Once Tod had finished composing the soundtrack for the individual experience in the Cocoon, it became clear that it was a very interesting experience simply to sing along with and be guided by that piece of music as a polished whole. Originally, we had envisioned that everything a participant heard in the headphones would be shaped and controlled by their vocal behavior, providing a full instrument under their control. We explored whether we wanted to interactively affect the playback of the Cocoon composition through standard techniques such as breaking it into layers to be built up by the participant's involvement, adding samples interactively triggered on top of it, or changing the processing of the sound in real time. However, the existing Cocoon music plus a participant's vocal explorations stood strongly on its own. We realized that the music had its own shape and arc that guided the participant through a variety of vocal explorations; to strip the music down to various layers would lose some of the power of the complete piece of music. Similarly, with the careful composition of the musical arc, incorporating various additional material or audio processing triggered interactively did not feel necessary or helpful either.

Therefore, we chose to use an ideal mix of the Cocoon piece for headphones created by Charles Holbrow, which carefully spatialized the material to add another dimension for a participant to vocally explore. We decided to try having the participant's live voice fed back into the headphones, and have the sound of this vocal reinforcement be an aspect affected by the user's vocalization. We experimented with different techniques for vocal modification, including binaural spatialization of the user's voice, vocoding, pitch modification, and other effect parameters. We used the Expressive Performance Extension system in our rapid iteration process. As we were experimenting with different vocal modification effects in Max/MSP, we kept iterating on new mappings in EPES to explore the entire loop of the system, from computational feature analysis through high-level parametric representation to output control parameters. As we continued our experiments, it was clear that any distortion of the voice needed to be carefully performed. The musical material used for the composition is primarily pure, unprocessed voice, and a distorted voice would not fit into the composition. Any vocal shaping needed to be quite subtle.

As we progressively simplified the interaction design of the Cocoon in the weeks leading up to the installation's arrival in Paris, we realized that a fully interactive instrumental model, where the user controlled aspects of what they heard through how they vocalized, was not actually what we wanted for this experience. Our goals included bringing people into a state of focus, where they could carefully listen to music, explore vocally, and experience their voice as vibration. We came to the conclusion that giving participants too much active and conscious control over what they heard (whether through changing aspects of the musical composition, affecting the processing of their own voices, or triggering generated vibration patterns in the Orb) would actually detract from the core of the experience. Instead of having the user "control" the musical composition, we chose to have the soundtrack for the experience "control" the behavior of the participant. Instead of hoping that a participant would happen to try different kinds of vocalizations and find themselves rewarded by some interactive result, we believed that the participant's explorations could be influenced more strongly by having everything they heard be carefully composed as a path for them to follow vocally. In fact, we even went further and took the stance that hearing one's own voice in the headphones was not only unnecessary, but even distracting. The point of this experience was not to hear oneself singing beautifully, but to feel free to play with one's voice, to experience one's voice as vibration. We felt that hearing one's own voice fed back into the headphones could lead to self-consciousness about how one's voice was blending with the musical composition, or about how "good" or "correct" one sounded.

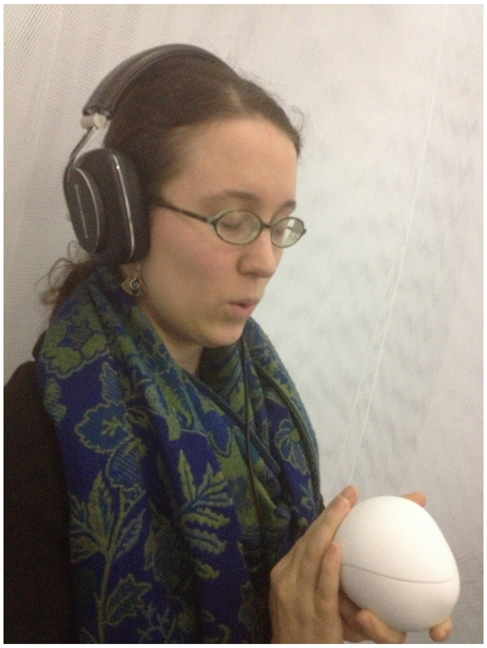


Figure 52. Vocalizing while holding the Orb
In the Cocoon experience, a visitor listens to a carefully sculpted composition and is instructed to sing a D guided by the music. The Orb vibrates with the vocalization. Photo by Rebecca Kleinberger.

We went through a similar simplification process in shaping the users' interaction with the Orb in the interactive installation. At first we envisioned the vibrational behavior of the Orb going in many different directions, with different pre-composed vibration patterns triggered by aspects of a user's voice, and certain vocal behaviors obviously "rewarded." For example, in the early mapping previously discussed, holding a pure note eventually caused the orb to add little knocking patterns that bounced around the sphere creating a kind of "purring" sensation. Many different kinds of tactile effects were tested on the Orb, and we explored various EPES mappings from vocal behavior to the shaping of those effects. However, the thing that was most compelling turned out to be material that felt like it came directly from the user's voice, such as using a filtered audio signal from the microphone and moving that microphone signal in patterns around the Orb. Modes that made the Orb appear to have too much of its own behavior (such as vibrational patterns completely unrelated to the actual vocal signal even though they were triggered or affected by characteristics of the vocal signal) broke the effect that this was a user's voice turned into vibration.

For the final version of the solo Cocoon experience, characteristics of the user's voice subtly affected the movement behavior of the sound patterns in the Orb, while the composition heard in the headphones was static. Based on parameters of the user's voice, we adjusted how much the audio signal was coming from all transducers or how much it was moving or bouncing around the Orb, and how quickly the signal was moving from transducer to transducer. We incorporated computational detection of vocal onsets to help link the movement of the sound to the user's articulation of new sounds. These subtle mappings allowed us to draw the participant's focus to the vibration of his or her voice and to the act of carefully listening and vocally exploring, while still maintaining a sense of liveness and variety in the tactile experience.

We carefully thought about what information was necessary to give participants beforehand to put them in the right mindset for this experience. In the installation, a trained mediator brings participants individually into the Cocoon and prepares them for the individual experience, by fitting them with microphone and headphones, giving them the Orb, and starting the pre-composed track. We gave the mediators a number of instructional points to convey to participants:

- This is a solo experience, You will be alone and free to experiment and play with your voice.
- It will last six minutes.
- Like the Chapel, this experience centers on one note, the D, that you can hear and sing along with.
- Try to be guided by and follow the music that you hear.
- This is a very different kind of experience of your voice, so don't worry if you don't hear your voice in the way you expect.
- The more you experiment and try different things, the more the Orb will come alive with your voice. See how different things feel.

6.2.4. Different Temporal Scales of High-Level Parameters

For Vocal Vibrations, one of the most important steps was determining which kinds of parameters we wanted to measure from the voice. We determined that there were three categories of relevant information: low-level features (such as frequency and amplitude), mid-level features that could still be computationally calculated from the signal (such as the number of onsets within a given temporal window and the spectral centroid), and high-level parameters that could be abstracted from the voice through machine learning (such as *stability*, *intensity*, and *complexity*).

In our analysis of the individual Cocoon experience, we identified one particular high-level expressive quality scale as being particularly useful. Originally, while prototyping a more instrumental model, we trained parameters such as *intensity* and *complexity*. However, once we had determined that the composition heard in the experience would be static, we started to see how that composition affected the vocal behavior of participants. A participant can respond vocally to the pre-shaped piece of music in many ways, with two kinds of models at the extremes. One extreme is a meditative model: trying to hold the D as purely and steadily as possible throughout regardless of the variation in the musical composition. The other is an exploratory model: playing with many different variations of rhythm, timbre, position of the sound, etc. as inspired by the variation in the musical composition. Given this, we decided to explore a qualitative parametric continuum of *exploration* from meditative to exploratory, as this parameter was highly relevant in describing an individual's expressive arc during this particular experience.

As we continued simplifying the interaction design, it was clear that we did not need this exploratory to meditative axis to affect the behavior of the music from moment to moment. Instead we viewed this parameter as something analyzed over the course of a participant's complete experience and used to provide some qualitative feedback about that experience. This experience has no "right" or "wrong" way of engaging with it, so we would not want to give feedback that related to accuracy (how good someone was at keeping on the D, for example, or how closely they followed the kind of sounds at each point of the musical composition). However, giving exploratory-meditative feedback about the overall experience might help guide people to try new things on additional visits to the installation. A larger goal of the Vocal Vibrations project has been the creation of models for long-time engagement with the singing voice. While the initial Paris installation is an experience that many people will only visit once, we wanted to continue thinking about models of repeat engagement. What if, after trying the experience, a visitor was given a card with some feedback about his experience? Say, where he fell along the meditative to exploratory scale, divided into a few different stages? He could then be given instructions about things to try for his next time at the experience: "You were very exploratory and played with a lot of different aspects of your voice. A different kind of experience you could try next time would be to really focus on holding the D, trying to keep it very steady…"

The final computation of *exploration* consisted of analysis at two timescales. First, a machine learning node was trained on vocal examples that were approximately one to three seconds long, consisting of several examples on each side of the spectrum and a few in the middle. The evaluation node calculates the desired value on the prior two seconds of data. This produces a continuous meditative to exploratory value reflecting the current amount of vocal exploration over the past two seconds. Second, a post-processing algorithm is used on this continuous parameter to produce a value representing the visitor's exploration over the course of their experience.

Vocal features that are used as input to the Expressive Performance Extension System include a variety of vocal features calculated computationally in an external Max/MSP program. These features are calculated on the current FFT window and include "Loudness," "Noisiness," "Frequency," "Skewness," "Odd to Even Ratio," "Centroid," "Sharpness," and whether or not an "Onset" is currently occurring. Additional features that describe the behavior of some of these values over time are calculated within EPES, including the number of onsets in the past second, the amount of variation of loudness over the past quarter second, the amount of variation in frequency over the past quarter second, and the amount of variation in the spectral centroid over the past quarter second. Six of these metrics are used directly as inputs to the machine learning process for *exploration*: amount of variation in loudness, amount of variation in frequency, amount of variation in skewness, amount of variation in sharpness, amount of variation in the spectral centroid, and the number of onsets in the past second. These features of variation were used rather than the direct values for parameters such as frequency and loudness because I considered the amount of change or lack of change along many axes more likely to indicate expression than the specifics of how those values changed.

The final training data set for *exploration* consists of 14 examples. 5 are labeled 0.0, representing extremely meditative samples (long notes, steady pitch, steady timbre). 5 are labeled 1.0, representing very exploratory vocalizations (rhythmic variation, lots of timbre change). 4 are labeled 0.36, serving as examples of some gentle vocal variation (perhaps changing only timbre). The addition of these intermediate values helped to correctly scale the range of the trained system, showing that variation in timbre still meant that the user was exploring different vocalizations, even if the pitch was reasonably continuous.

This training data set was gathered in Paris to replace the original training data, since the amount of background noise due to the Chapel piece playing simultaneously in the space was substantially different than the amount of background noise for the training examples recorded at the Media Lab. This required the microphone levels and the sensitivity of the computational analysis algorithms to be changed to attempt to only pick up signal when a user was singing, which provided different ranges of values to the system. Due to the speed of the process for capturing new training data examples, it was possible to easily replace training data and get more accuracy in the real space.

A multi-layer feedforward Neural Network was used to perform regression on the input data to produce a value for *exploration*. This network has 360 input nodes (6 inputs multiplied by 60 frames of data, 2 seconds at 30 frames per second), 30 hidden nodes, and one output node. The trained network is then used to evaluate input data continually in real time, calculating a current value from the past two seconds of data. While the length of the samples used in this process introduces an element of latency that would be extremely noticeable if any output value was being controlled live by this parameter, this latency is completely acceptable in this evaluation process because it is being used for a calculation whose results are not presented to the user during the experience. Even if some kind of feedback about the user's amount of exploration was desired in the middle of the experience (perhaps through a visualization), it is reasonable that this metric would not be something expected to change completely in a fraction of a second.

A second expressive vocal parameter of *stability* was also added to refine the concept of the expressive space of exploration. This parameter attempted to draw out the difference between the kind of variation that is caused by a user being very free with different kinds of vocalizations around a pitch, and the kind of variation that represents the user giving up on holding a note and just speaking or jumping around through many pitches. The former type of variation is more desirable in this experience, the latter is less desirable. While one extreme of each axis may be similar (stable, meditative notes), the other extreme is fairly different (exploratory variation vs. random variation). A separate set of training data was gathered for this parameter using a different machine learning training node, as the desired length of examples was shorter than those used for the *exploration* data. This training data consists of 16 examples labeled 0.0 (very unstable, random, talking) and 5 labeled 1.0 (very stable, constant pitch, smoothly changing or steady timbre). This parameter took more iterations for gathering sufficient training data than other parameters, attempting to clarify the distinction between very random vocal input (such as talking) from vocal input that varied but was still expressively meaningful. When testing on a pulsed rhythm on one pitch, for example, the system should not predict "very unstable."

In addition to the continuous *exploration* and *stability* values calculated live on the user's vocal behavior, a post-processing algorithm was designed to gather a value for this *exploration* parameter that represents the participant's overall percentage of exploratory versus meditative behavior through the course of the entire experience. This post-processing step checks at each time step to see if the participant is currently singing (if the average amplitude for the last second has been above an empirically determined threshold). If so, the current *exploration* and *stability* values are stored, as well as the amount of change in the spectral centroid over the past two seconds. The average of the stored exploration and stability values can then be calculated and combined through a hand-crafted algorithm with the average of the amount of change in the spectral centroid (to reflect subtle timbre changes). This combined value forms a metric of the user's overall exploratory behavior for a longer time period, up to the entire length of the six-minute experience.

An important thing to note is that this kind of exploration model requires machine learning at longer timescales and several timescales. Suppose a participant is attempting to hold the D reasonably steadily, with consistent timbre and volume. Too short of an analysis window might overly weight moments when the user pauses to take a breath (when no pitched data is captured within the given analysis window). A longer analysis window or set of windows is necessary to properly capture concepts of "exploration." One could imagine definitions of "exploration" that examine only a participant's timbral variation across the entire experience, for example. For this project, we chose to primarily focus on giving windows of a few seconds long to the machine learning systems, and then analyzing the range of values collected throughout the experience to determine one "overall" measurement of the experience. While we did not end up incorporating this post-experience feedback layer into the final design of this version of the Vocal Vibrations experience due to logistical and staffing limitations at our venue, our experiments with this parameter and the resulting feedback were very informative and could easily be integrated into future experiences.

6.2.5. Public Reaction to the Vocal Vibrations Installation

The Vocal Vibrations installation had a private opening for press and invited guests on March 27, 2014, with a public opening on March 28. The installation is running in Paris through the end of September 2014, and will come to the new Le Laboratoire in Cambridge, MA in October 2014.

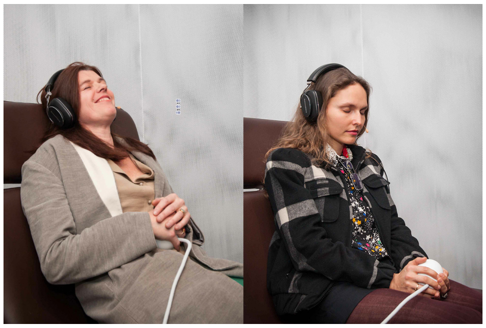


Figure 53. Cocoon visitors exploring the Orb
Visitors had a variety of reactions to the experience of feeling their voice in the Orb. Photos provided by Le Laboratoire.

Participants were very excited to have a completely new form of experience with their own voices. Even visitors who began interacting with the installation very quietly and tentatively still found the interaction with their voice in the Orb and with the musical composition to be compelling. Interestingly, we indeed saw many participants experimenting with their voices in the exploratory and meditative directions that we had predicted and seen in our original tests. Some participants chose to carefully focus on the audio experience and to attempt to hold the D steadily. Others, interested in the transformation of their voice via the Orb, wanted to try many different sounds with their voice to see how the Orb would behave. Accordingly, some visitors found the overall Cocoon experience calming, while others found it exciting.

Additionally, visitors found the Chapel experience to be quite engrossing and meditative. Many visitors stayed for long periods, sitting on the cushions and benches and listening to the piece. Even families with small children enjoyed the experience; we observed one family with an infant peacefully staying in the Chapel for almost an hour. Le Laboratoire has also collaborated with other groups in Paris to hold experiences in the Chapel space, such as yoga classes and even gatherings for parents with babies.

One challenge in the initial opening of the installation, when we were faced with very large crowds, was how to let as many people as possible experience the Cocoon. Since the standard Cocoon interaction is structured to be a six-minute individual experience, this normal experience would not work at times of high traffic. We thus ran two demonstration variations of the experience during the first two days. We allowed groups of participants into the Cocoon together for an explanation, then let each member of the group take a limited turn with the headphones, microphone, and Orb. In order to allow more participants the opportunity to experience the Orb, we also performed some demonstrations where one of us would sing to show the effect of different vocal sounds in the Orb and pass around the Orb for visitors to hold. Interestingly, people found the sensation of holding someone else's voice in their hands equally compelling as holding their own; they experienced it as a very personal connection. For the remainder of the installation, the exhibit moderators guided individual participants to have the entire six-minute experience.

In the first months that the installation was running, we have received additional feedback from the moderators at Le Laboratoire about visitors' experiences. Many visitors highly enjoy the experience, particularly the solo experience portion. However, we did get the feedback that visitors commented the vibrations of the Orb felt too subtle at high vocal frequencies. Due to the material properties of the ceramic shell, higher frequencies dropped off much more rapidly and did not carry to the fingertips as strongly. In response, we have adjusted the behavior of the Orb to add a slight pitch-shifted layer of sound to the existing filtered signal, so that there is always a component of the sound that was an octave lower than the participant's voice. This signal stays sufficiently connected to the voice while allowing for a strong vibration in higher vocal ranges.

6.2.6. The Role of the Expressive Performance Extension System

An important aspect of the use of the Expressive Performance Extension System in this installation was its flexibility for rapid prototyping and iteration throughout the course of the design and development process. We began the concrete development of the Le Laboratoire installation imagining one particular model of interaction, but rapidly moved through several different interactive models attempting to get the feel of the overall experience right. Having one system stay at the core of the interaction design allowed for consistency and quick development, even as we tweaked the computational vocal analysis process and continued to reinvent the output modalities and output control parameters.

As we developed the exploratory to meditative scale, a new kind of expressive parameter that was meaningful in the context of this specific piece, we also were able to quickly train on examples and experiment with mappings. Had we tried to do these qualitative mappings by hand, it would have been much more time-consuming to develop algorithms to connect the variation of several variables to a position on this expressive axis. We knew that several features of the input might be important for helping to convey these concepts (such as the amount of change of frequency, change of amplitude, change of spectral centroid, harmonicity, and rate of onsets), but we did not need to empirically figure out how each of these features was related to our perceptual sense of exploration to meditation in order to explore these concepts.

EPES also allowed for development of our interactive ideas in one environment (the Media Lab), and then quick re-training of the system when we arrived in Paris and set up the installation in its real context. Since the only input data we were using for the mapping system was parameters of audio data from the wearable microphone, the differences in the sonic background environment were a concern. There was little sound isolation between the Cocoon and the Chapel, so it was necessary to retrain our expressive parameters in the real space with the background noise of the Chapel. Since we had already figured out what expressive parameters were useful, and some examples of the variety of training data that we needed to capture for each parameter, it was quick to capture new data.

As with the Powers Sensor Chair, the Vocal Vibrations installation had to be designed to be interesting with a wide variety of input behavior. Participants were given very broad instructions and each participant would come in with their own vocabulary of sonic exploration. While the arc of the musical composition suggested different kinds of sounds and vocalizations that visitors could explore, each participant had a unique interaction with the installation, from quietly humming into the microphone to slow chanting to swooping glissandos to a broad range of vocalizations. Some participants experimented with many different interactions over the course of their solo experience, others preferred to keep one mode of interaction. The range of each participant's vocalization and variation of vocalization also differed. Given the great variability of input, and the fact that we could not predict anything beforehand about how a participant would interact with the installation, we had to design a system that did not have a "right" or a "wrong" way to interact with it. Different vocabularies of interaction might evoke different responses in the Orb, certain kinds of behaviors might be found to be particularly powerful, but everything had to be interesting.

In future versions of such a system, it would be useful to do some quick system tuning for each participant. Perhaps when a participant begins vocalizing, the system could listen to the first fifteen seconds or so and try to get a sense of whether this person is tentative or bold, singing softly, trying many things or being quite stable. Then the behavior of the Orb could be scaled or moved into different modes to create the best possible experience. For example, if someone comes in and only hums very gently, it might make sense to make the Orb quite reactive, so tiny changes of the participant's voice would evoke greater variation in the Orb.

In conclusion, the Vocal Vibrations installation demonstrates several interesting elements:

- The system supported analysis of expressive parameters at many different timescales: current, last phrase, and entire performance.
- The most interesting expressive axis measured defined a exploratory to meditative vocal experience.
- A subtle interaction through a tactile interface was a better choice than a consciously-controllable instrumental model for the installation's goals of encouraging careful listening and focusing on one's own vocal vibrations. Should the system control the user or the user control the system?
- The question of how much and what kind of feedback to provide the user was particularly important.
- The Expressive Performance Extension System was quick to re-train on location as necessary.
- The design process demonstrated the flexibility of the Expressive Performance Extension System throughout the entire development arc of an installation: ideation and quick sketches, rapid prototyping, and exhibition.

6.3. The Body and Voice: *Crenulations and Excursions* and *Temporal Excursions*

With the Powers Sensor Chair and Vocal Vibrations, we have discussed work featuring analysis and mapping of either the voice or the body. As the Expressive Performance Extension System provides similar tools for either performance modality, it was important to test how the system could support creating pieces that incorporated both body and voice in combination. *Temporal Excursions* is a solo vocal and physical performance piece where layers of sound proliferate and surround a performer, shaped by qualities of both her voice and her movement. The sonic world and movement vocabulary used for *Temporal Excursions* is inspired by that of an earlier extended movement piece, *Crenulations and Excursions*, which will be discussed first.

6.3.1. *Crenulations and Excursions*

Crenulations and Excursions is a combination dance performance and installation space. This piece allows a solo performer or a visitor to explore a rich sonic space through her expressive movement. With a tiny, energetic movement, with a fluid and sweeping gesture, a performer can create and shape layers of sound around herself. *Crenulations and Excursions* draws on a conducting metaphor rather than an instrumental metaphor, where the performer's movement is generally used to shape and guide the qualities of the resulting soundscape rather than to trigger individual sounds. The performance explores the body as a subtle and powerful instrument, providing continuous control of continuous expression.

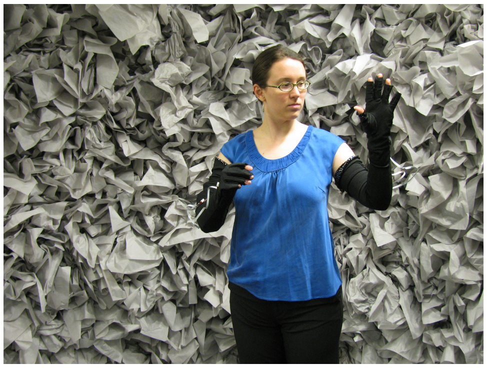


Figure 54. ***Crenulations and Excursions***
In the *Crenulations and Excursions* space, a performer can create a soundscape through her movement. Photo by Peter Torpey.

Crenulations and Excursions is driven by the Expressive Performance Extension System, which captures data about movement in the space and transforms that data to a continuous space of abstract expressive parameters. Points in and trajectories through this parametric space are then mapped to control parameters for the ordering and layering of micro-samples of sound arranged in a set of sonic spaces.

The sonic material for this piece is controlled via `OSCtoMIDIGenerator` software described in Section 6.1. This control program determines when different MIDI commands should be sent to a virtual MIDI instrument, based on musical shaping information from an interactive mapping. The virtual instrument playback is performed in Max/MSP. Each MIDI channel features a different vocabulary of sound, with small samples that play in their entirety when a note is played. These samples, from .5 to 8 seconds in length, are created from material from Freesound.org, as well as from recordings made at the Lab of group vocal improvisation in a resonant space. The sonic vocabulary and note arrangement of the individual MIDI channels include: vocal samples, arranged from pure notes to complex chords and processed voices; string-like samples, arranged from a low gritty drone to atmospheric synthesized chords to higher-pitched, purer single note samples; and "mechanical" samples arranged from legato to staccato, starting with longer multilayered drones and moving to short metallic bangs and buzzes. Another MIDI channel incorporates elements from several of these vocabularies, building through low vocal drones, to layered sung melodic fragments, to synthesized chords, to mechanical soundscapes, to staccato bursts of crunching glass, electric buzzing, and typing on a keyboard. In the performance of this piece, sonic palettes from multiple keyboards are layered to create even richer sound worlds. Multiple notes can be played at once. The sound dies away completely only when the performer is still for a few moments, and is brought back immediately when she again raises an arm.

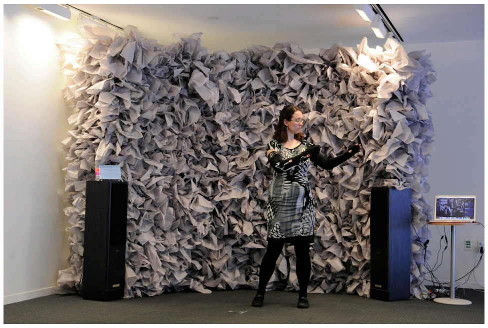


Figure 55. The *Crenulations* installation space
Speakers on either side of the tissue paper backdrop create the sonic space. Photo by Andy Ryan.

This performance also incorporates a scenic design component to transform the space and create an evocative and inviting environment for performance and experience. This environment appears as a sculptural and textural outgrowth of the Media Lab building, visually extending and surprisingly shifting the space. It mirrors the rich range of textures and qualities present in the sound and movement, and physically alters the acoustic results of the sonic playback to create an enveloping sonic environment. This backdrop is constructed of tissue paper attached to sheets of poster board that are then mounted on a curved display frame. Speakers are positioned on either side of the scenic backdrop, pointed into the space implied by the curve of the scenery. From afar, the sound design can be heard sufficiently clearly; when a performer or installation visitor steps into the space, the sound is enveloping.

Installation visitors are tracked through noninvasive sensing: a Kinect used as a hand tracking system and as a webcam processed to determine activity levels in the space. The solo performer additionally wears a set of long gloves enhanced with accelerometers that capture higher levels of detail about her arm and hand movements. All of this data is sent to the Expressive Performance Extension System, which handles feature computation and abstract movement quality analysis. In this version, the primary movement qualities are Laban-inspired, consisting of the axes of *time*, *weight*, and *flow*. The impact of different sensor features on the performer's current location on these expressive axes was hand-coded into the `QualityAnalysis` node class originally designed for *Death and the Powers*. EPES is also used for the mappings from features and abstract qualities to the control parameters for the `OSCtoMIDIGenerator`**,** including which keyboards are selected, the velocity of given notes, and the length of time between notes on continuously playing keyboards. In addition, to the control parameters for the `OSCtoMIDIGenerator`, other control parameters are sent directly to the Max/MSP patch that contains the virtual MIDI instrument. These parameters affect general aspects of the sound as a whole, such as the overall dynamic level. In the performance version, there are several different mappings in EPES, each controlling different sets of keyboards, that are switched via a timer in Max/MSP to synchronize with a soft layer of sound that plays throughout the piece. This performance version incorporates input sensor data from both the wearable sensor system and the Kinect. In the installation version, the system is placed in one mode where multiple keyboards are layered and controlled, and the only movement capture information comes from the Kinect and the Kinect's webcam, using the specialized `KinectInput` devices and hand-coded feature computation nodes for webcam analysis and for hand tracking analysis. This piece did not incorporate machine learning techniques for expressive quality analysis, as all of the feature computation and expressive quality calculation were hand-coded.

I designed and implemented the wide range of creative elements in this piece, including the choreography, wearable sensors, sonic systems, interaction design, and scenic design. I also served as the solo performer in the performance version of the piece. I completed a version of this performance and installation for the Media Lab's first internal version of "The Other Festival" in April 2013.

6.3.2. *Temporal Excursions*

As a continuation of *Crenulations and Excursions*, I developed a solo performance piece titled *Temporal Excursions* for the Media Lab's Festival of Art and Design. The first performance of this piece took place at the Media Lab in December 2013, as part of a concert of new performance works entitled *WOOD-WATER-WHISPER-WILD*. This piece sought to expand the movement vocabulary of *Crenulations and Excursions* and incorporate recognition of vocal qualities, creating a performance work for both body and voice. Movement and voice trigger and shape a sound cloud of vocal, choral, and mechanical samples to accompany the live vocal performance of a text that is part spoken, part sung. This piece explores the idea of "nostalgia for the present," the sensation of experiencing in a present moment some of the nostalgia one will eventually feel for that moment. I created this piece for myself as a solo performer.

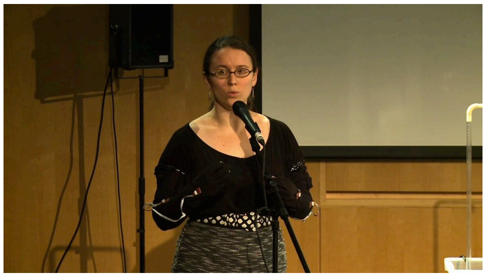


Figure 56. Performing *Temporal Excursions*
Image via Paula Aguilera

Sensing requirements for this piece included that both movement and vocal information be captured, that the sensors be quick to transport into the necessary performance space and not require additional calibration once set up (as this piece was one piece on a program of performances), that very subtle movements could be captured, and that a clean vocal signal could be obtained that was not distorted by the sonic accompaniment. To meet these goals, movement data is captured via accelerometers on a pair of long gloves, and vocal data is captured via a stand-mounted microphone.

This piece uses the Expressive Performance Extension System to gather input, calculate features, perform machine learning to determine expressive parameters, and map movement and vocal information to control parameters of the output soundscape. Signal features calculated within the `AudioAnalysis` input device on the FFT of the input signal include *amplitude*, *frequency*, and *dissonance* (a hand-crafted metric of how different the three formats of maximum energy after the fundamental frequency are from multiples of the fundamental frequency). Additional vocal features are calculated in the feature computation stage over a half-second window, 15 frames at a frame rate of 30 frames per second. These features are defined in the `VocalFeatureComputation` node, as described in Chapter 5: *overall change* (how much each input parameter has varied from frame to frame over the past window); *average change* (the average amount of change between frames over the past window); *derivative change* (the amount of change over all input parameters in the past four frames, looking at a smaller window of time than the overall change value); *overall value* (a weighted average over the window of all parameters); and *accumulated change* (an accumulated metric of input variation that is incremented or decremented on each frame by an amount proportional to how much the inputs have been changing). The vocal features used as input to the machine learning nodes are *amplitude*, *frequency*, *dissonance*, *overall change*, *average change*, *derivative change*, and *overall value*.

Particular vocal qualities of interest learned for this piece were *energy* and *complexity*, while key movement qualities were inspired by Laban's concepts of *time* and *flow*. The function between accelerometer values on the performer's gloves and the expressive axes of *time* and *flow* was hand-coded into the `QualityAnalysis` node class originally designed for *Death and the Powers*. The vocal qualities were analyzed via the machine learning tools in EPES. The concept of the *complexity* of the voice was found to be the most significant quality axis used in the design of this piece. The system was trained for this *complexity* parameter with 6 samples labeled 0.0 that demonstrated a simple vocal quality (pure, extended vowel tones in a variety of pitches and amplitudes) and 4 samples labeled 1.0 that demonstrated a complex vocal quality (sequences of harsh consonants, spoken text, and quick rhythms). This axis of complexity was then used in several performance mappings to select different pools of samples with different sonic qualities depending on the complexity of the input vocal behavior. In one performance mode, an additional *complexity* model was trained to reflect a different definition of complexity focused on the singing voice (pure extended tones to fast patterns). This second model used 9 training data examples, 5 samples labeled 0.0 demonstrating a simple vocal quality and 4 labeled 1.0 demonstrating a complex vocal quality. Both *complexity* models were used in one of the performance mappings to control different aspects of the sonic extension. The final training data set for *energy* consisted of a total of fourteen samples, 6 labeled 0.0 (calm examples) and 8 labeled 1.0 (energetic examples). All of these training data examples were normalized to half a second long. I performed all of the collected vocal training examples, since I would be the performer for the final piece.

In the testing process for this piece, small numbers of training examples were gathered for each parameter, a model for each parameter was trained on the training data set, and the accuracy of each parameter was then tested on live input. Input samples were evaluated using the most recent half-second of data as a sliding window. These predicted values could be compared by eye in real time to the expected values. If the parameter values seen as output did not reflect the desired values, additional samples were added to the training set to attempt to clarify the boundaries of the parameter. For example, the initial training data set for *energy* did not consist of any samples where the singer was silent. On providing test input to the trained system, silence caused the system to output high values for *energy*, which was not seen as desirable or expected behavior. Additional samples of silence were thus added with a 0.0 value for *energy*.

This system uses feedforward Neural Networks for performing regression on expressive vocal parameters, with a separate network trained for each parameter to be analyzed. These networks are constructed with the default Neural Network structure and settings defined in the Expressive Performance Extension System, with one hidden layer with 30 nodes. The dimensionality of the input for each of these networks is the number of input data streams (7 inputs) multiplied by the normalized window length (15 frames). This results in a model with 105 nodes in the input layer, one hidden layer with 30 nodes, and a single node in the output layer. The output value for each network represents the predicted value of that parameter, and labels for training data represent the ideal output value for that parameter. These networks were then incorporated into `MLExtendableEvaluationNodes` for testing and incorporating into mappings.

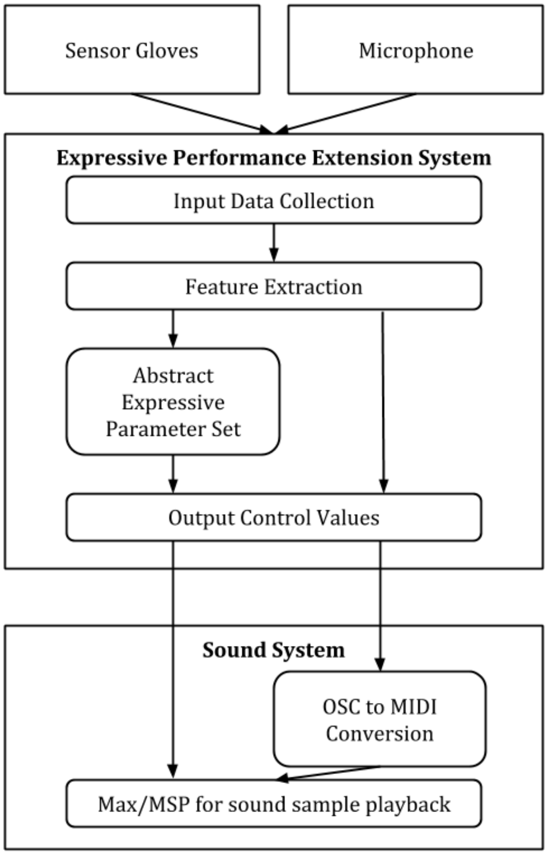


Figure 57. ***Temporal Excursions*** **system diagram**

The piece as a whole develops and builds through several different interaction modes, represented by different mappings in the Expressive Performance Extension System. The interaction begins with pure movement controlling the accompanying sound, followed by continuous vocal control, and then by two different mapping modes that combine expressive information from both movement and voice in shaping the sonic accompaniment. These modes are switched by the performer through the use of a single foot pedal. As the desired sequence of mappings is known beforehand, the performer can trigger the next mapping through pressing on the foot pedal, which sends a signal to the Max/MSP patch. The Max patch calculates the number of the next mapping cue, and sends a message to the Expressive Performance Extension System to trigger the desired cue.

6.3.3. Analysis and Evaluation: An Expert Performance System

This performance was useful to verify that the Expressive Performance Extension System could easily learn the desired range of expressive qualities from a reasonably small sample set. Additionally, since Neural Networks were used for the machine learning component, providing examples at both extremes of the specified axes allowed for reasonably good interpolation along the axes without needing a large set of training data. I began the process of developing the piece with a first version of my desired text and drafts of the keyboards for the soundscapes (borrowed from *Crenulations and Excursions*), then started envisioning how the interaction could develop throughout the piece. As I explored how I wanted the text to be performed, I was able to easily train new sets of experimental parameters or retrain parameters based on what I was developing about the performance style. Was the text to be spoken, sung on one note, sung with a range of melodic or timbral variations, some combination of all of these? How could the soundscape bring attention to the distinction between spoken and sung text? How much movement did I want in the piece: would I be standing at a microphone with only arm movement, or would I have a broader range of dance movement around a space? What elements of my vocal performance were interesting to highlight? How much of the performance details of the piece (movements, particular melodic patterns) would be improvised or be specifically set?

One design aspect that proved especially challenging in this piece was integrating movement information and vocal information into a single control mapping. In my early mapping attempts, I explored having certain pools of samples controlled by the body and others by the voice, only to find that the effect was that of playing two instruments simultaneously, leading to a challenging performance task. More effective were mappings that incorporated both movement and voice information seamlessly to control different aspects of the same instrument: for example, using vocal complexity to select kinds of accompanying sound samples (from pure tone vocal samples to harsh mechanical sounds) while using the rate of movement to control the density with which those samples were layered. The best mappings encouraged a performance in which movement and voice were not thought of as separate control elements, but as unified aspects of an expressive performance.

The creative mapping problem of this piece differs in an interesting way from the movement and voice mappings in *Death and the Powers*. In *Powers,* the performer who is measured is unaware of the specific results of his expressive actions. He can perform expressively with his voice and body together, without being aware of the digital extensions of his performance. Our job was to create mappings that worked with that natural performance style. In *Temporal Excursions*, I was developing the performance style along with the development of the mappings to the sonic extension of that performance. I was very aware of the sonic results that occurred from shifting vocal or physical qualities of my performance, and used that awareness while shaping my performance choices in the development of the piece.

An interesting element in designing mappings around these more abstract high-level quality parameters was the temporal latency inherent in parameter analysis. Since this piece was trained on training data samples around half a second long, the system would immediately start to identify that a given quality (such as complexity) was changing, but would take a moment to be able to confirm that the quality had, indeed, shifted to a particular place on the quality axis. For example, if the performer is speaking with high focus on consonants (a complex sound) and then shifts to a sustained pitch, the system immediately begins to drop the estimated complexity, but has to wait a (noticeable) fraction of a second before correctly identifying that the current sound now has very low complexity and reacting appropriately. On the one hand, this is desirable behavior, since we do not want the system reacting to overly short intervals of sound (if a performer is holding sustained notes, for example, a breath or a consonant should not immediately be identified by the system as "very complex"). However, it is necessary in mappings and performance to be aware of these different layers of temporal latency.

When performed in December for the Media Lab community and other audience members at the *WOOD-WHISPER-WATER-WILD* concert, *Temporal Excursions* received some very positive feedback. One audience member commented on the way he "forgot about the technology two lines in" and was able to experience the emotion and story of the piece. Given my goals of developing new performance experiences through technology rather than performances that are about using new technology, this was a particularly satisfying piece of feedback.

[creation of a sound world through gesture, which then dissolves: mapping #3]

[spoken, no movement, no sound, a pause]
Tell me, what do you feel for the past? Or better, what do you feel for the present? A sense of nostalgia, perhaps, a sense of distance? The feeling that something that was here is now irretrievably far away?

[spoken in tempo, moving, bringing back in sound with movement]

It's a lonely state, the present. Distant, disconnected from itself. Or is that just me?

Here's the question: would you miss tomorrow, today? Or maybe it's the other way around. Would you miss today, tomorrow?

Imagine:
[sung, mostly one pitch: mapping #9. Just voice control of interactive sound]
you're standing on a bridge at midnight,
looking out over the water,
watching lights shine on the surface.
And you know you will look back
and miss this moment
miss yourself in this moment,
this place and time.
And you stand there, looking over the water
already missing it now.

[Vocal improvisation interlude, no text, movement added...playing with the system, full range, biggest mapping: mapping #12]

[spoken rhythmically]
And the stories of paths that never happened...the things that never were your past and will never be your future. The ones where things went another way, or many other ways. You miss those too.

[sung on one note: mapping #11, sporadic sounds triggered by movement but quality shaped by voice]
And you stand on the bridge,
missing a moment
that has not yet passed.

[moment of just movement control of sound: mapping #3, as in beginning]

Figure 58. Script for *Temporal Excursions*
Shifts between spoken and sung text and changes of mappings are indicated.

Temporal Excursions demonstrates a few interesting features of the Expressive Performance Extension System and the Expressive Performance Extension Framework:

- The system and frameworks proved useful for an instrumental model, combining movement and vocal analysis for performance extension in a way that was learnable and repeatable.
- The Expressive Performance Extension System could easily learn the desired range of expressive qualities from a reasonably small sample set.
- Designing output media that was simultaneously controlled by qualities of movement and voice required careful thought.

6.4. Other Projects Utilizing These Systems and Constructs

6.4.1. *Trajectories*

For the first annual HackingArts conference on art, technology, and entrepreneurship at MIT, Peter Torpey and I were invited to create a short performance piece with interactive visuals and sound using several of the systems that we have designed over the past few years, including the Expressive Performance Extension System. This piece was performed as part of the conference in September 2013.

This seven-minute theater piece, which we called *Trajectories*, explored the concept of the multiple paths that a story or a relationship can take, using the device of an imperfect storyteller. The storyteller (a role I performed) creates the world of the story, in which a man and a woman meet and connect with one another. The storyteller does not simply create this world through words, but also through using her movements to control a soundscape and a series of interactive projected visuals. As the piece progresses, however, it becomes clear that she is having trouble getting the world just right for the story to progress the way it is "supposed to," requiring her to reset and retry, changing her behavior and performance parameters until events happen in the desired manner. As the story takes on more of a life of its own, she starts to long for the many possible alternate paths that are, in their own ways, equally true and possible. The man and woman are then joined by several different pairs of actors, each playing out the same interaction in different directions.

Figure 59. The cast of *Trajectories*
In the final scene, many different variations on the story unfold as the narrator loses control. Photo from Hacking Arts.

Trajectories was an excellent example of how the use of abstract high-level parameters helped to simplify communication between a variety of different systems. The piece integrated many systems: the Expressive Performance Extension System; live visuals created through the `RenderDesigner` system originally developed for *Death and the Powers*; an interactive music generation system, `ParaMIDI` originally developed for Peter Torpey's performance of *Figments*; and the `OSCtoMIDIGenerator` system developed for *Crenulations and Excursions*. We designed the overall shape of the piece in Peter Torpey's Media Scores software, a scoring and live control system for multimedia performance (Torpey, 2013). In the Media Score, we defined different sections and modes of the piece, as well as shaped the overall expressive arc of the show, defined through abstract parameters such as intensity, density, and rate. In the performance, Media Scores sent this continuous abstract parameter data to the Expressive Performance Extension System and sent discrete triggers to change modes in EPES, `ParaMIDI`, and `RenderDesigner`. EPES took in live data from two accelerometer-outfitted gloves and analyzed that to determine positions in a modified space of Laban-inspired parameters (weight, time, and flow). Combined with the live parameter data from the score, this data was then mapped to control parameters for `ParaMIDI`, `RenderDesigner`, and the sonic world. All of these systems were connected via Open Sound Control and used abstract parameters for high-level control and mapping.

This piece was an interesting exercise in controlling multiple media at once through movement. As we developed mappings for different sections of the piece, we found that there were times when we wanted the live performance to affect the *Crenulations* soundscape, the musical score, and the visuals simultaneously. In order for me to be able to think about controlling all of these elements as a solo performer, it was necessary to design the same qualities of movements to have meaningful effects on all of the different output media. I did not want to play several separate instruments with one movement, I wanted to use the language of my body to control many media simultaneously with a simultaneous effect. We found that the use of abstract parameters and movement qualities was very helpful in this effort, allowing us to think about the input movement as a cohesive unit.

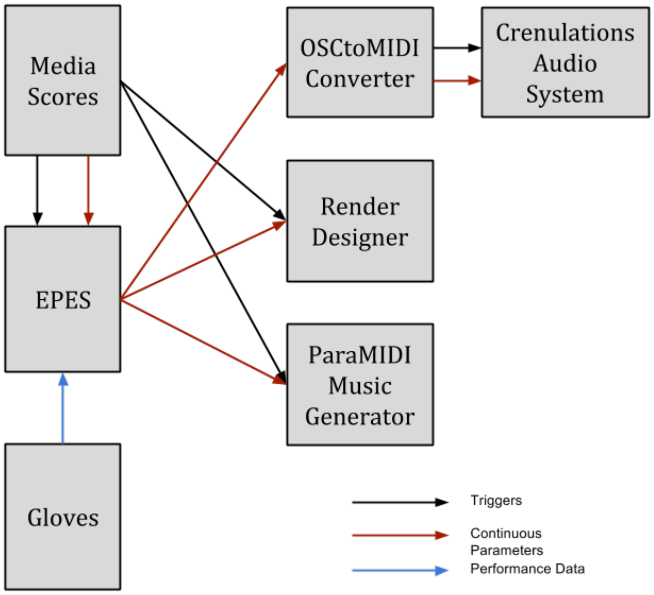


Figure 60. ***Trajectories*** **system flow diagram**

6.4.2. Blikwisseling Workshops

I have also used the concepts of abstract parametric modeling of movement and voice and these performance extension mapping systems in a series of interdisciplinary workshops in the Netherlands. In May 2013 and May 2014, Peter Torpey and I led week-long master classes on performance and technology. These workshops were constructed around the concept of "Blikwisseling" ("Change of Perspective"), bringing together participants from a variety of backgrounds, from music theater to product design to chemistry to computer science to architecture to cinematography to creative technology to music therapy, working together to create performances and experiences.

The May 2013 workshop was themed around the 100th anniversary of Stravinsky's *Rite of Spring*, imagining developments in musical and performance technology in the next 100 years. In the May 2014 workshop, the theme was "Waves," which inspired us to explore how a simple concept and structure could be valid across many modalities in both literal and metaphorical ways, from light to sound to waves of emotion to water to physical movement to artistic movements. In both workshops, our explorations with the participants included the concept of abstract parameters to represent expression in a multimodal performance or experience. We led the participants through a variety of exercises exploring abstract parameters, including the creation of a parametric score using pieces of yarn to represent different parameters. Through these exercises, participants discovered key points about parametric representations and performances of a piece, such as the difference between discrete triggers and continuous parameters, and the difference between low-level parameters (such as volume) and high-level parameters (such as the tension of an argument).

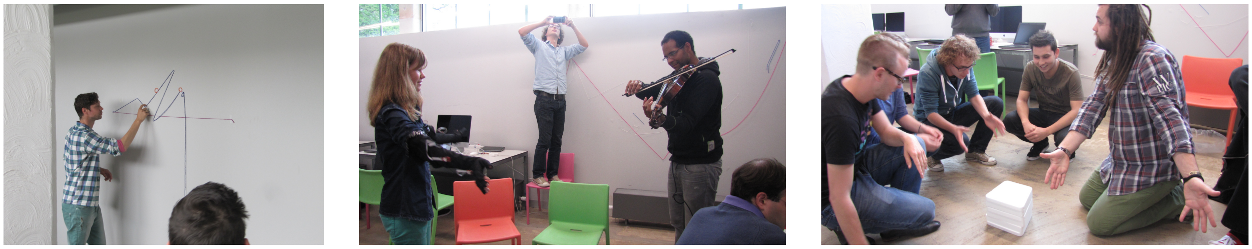


Figure 61. Blikwisseling workshop participants
Left: A participant explains a parametric score made with yarn. Middle: Early explorations of an augmented viola duet, where arm movements by one performer modified the sound of the viola. Right: Rehearsals for a piece creating a ritual for interacting with a movement-sensing box.

Several of the final performance and installation pieces developed as part of both workshops incorporated versions of the Expressive Performance Extension System. In one musical performance, the audience was able to collectively conduct a composition through their movement, with different sections of the audience in control of different instrumental lines. In a single-instrument duet for viola and gestural interface, the qualities of one performer's movement wearing sensor-enhanced gloves manipulated the sound of the violist's instrument. One group created a ritualistic performance around a special box, which triggered and shaped sounds and recorded text by the way it was shaken, thrown, and caught. A solo performer created an improvisatory "hyper-theater" experience, where his character of a patient in an asylum was accompanied by sound and music triggered by his arm and head gestures. Finally, in an interactive installation, carrying and tossing a ball across different areas of the space controlled visual projections and a musical composition, and audio parameters of the musical composition in turn helped shape the visuals in real time. In these pieces, the Expressive Performance Extension System proved flexible for a broad range of different concepts and mapping strategies.

6.4.3. *Death and the Powers* in Dallas

In February 2014, we brought *Death and the Powers* to the Winspear Opera House in Dallas, Texas, presented by The Dallas Opera. An important detail in this set of performances was that the role of Simon Powers, originated by James Maddalena and performed by Maddalena in Monaco, Boston, and Chicago, was portrayed in Dallas by baritone Robert Orth. As the character of Simon extends into the entire theatrical set, the performance of the actor is used to control a variety of media from the patterns of light and color on the LED walls to transformations of the sound in the space to the movement of robotic elements. The mappings of Powers' expression to all of the interactive media were developed based on Maddalena's performance, so we found ourselves with the challenge of how much to adjust the mappings based on the differences in Orth's performance, and how much to guide Orth's performance to work with the existing technological systems. We introduced Orth to the sensors and the Disembodied Performance System at a visit to the Media Lab, allowing him to experiment with how his vocal and physical actions affected the wall visualizations in a few cues.

It was fascinating to see the differences in the output media between Maddalena's performance and Orth's performance. In particular, the message had not immediately been conveyed to Orth by the directorial team that he was responsible for physically gesturing and moving in the way he would onstage even once he had been sent down into the sound isolation booth and been fitted with his movement and breath sensors. In the initial full technical rehearsals, the visuals on the walls seemed too dull. There was some quality that we remembered them having that was not evident in their current state. Confused, I examined the live sensor data streams via the mapping system, only to find that there was almost no movement being detected on the sensors on the arms. I thought perhaps the sensors were not being put on Orth during the rehearsal, but I was told that his dresser was putting them on correctly for each run. We realized that he was wearing the sensors, he simply wasn't moving his body. After a session with Orth where the associate director and I walked him through the show and explained the sections where he particularly needed to be physically (as well as vocally) expressive, the difference in the media for the following rehearsal was striking. As the visuals also respond to the voice, they were already clearly connected to the live performance; however, they had been missing the element of the mappings controlled by the qualities of movement. We were reminded of the influence of the gestural input on the resulting visual output: it is, indeed, a rather different extended performance when the performer is providing appropriate movement content.

Figure 62. Robert Orth as Simon Powers
Robert Orth in Scene 1 of *Death and the Powers* with Patricia Risley as Evvy and Joelle Harvey as Miranda. Photo by Karen Almond.

6.4.4. The *Powers* Interactive Global Simulcast

The Expressive Performance Extension System was also used to create new performance mappings developed for the interactive global simulcast of *Death and the Powers* that accompanied the opera's performances in Dallas. In concert with the Dallas production, we sought to address the challenge of bringing the opera to a broader audience. While *Powers* is designed to be easy to tour, it has certain venue and budgetary requirements, and, as with any modern opera, it has limited opportunities for performance. To broaden the show's reach, the final Dallas performance was broadcast live to nine cities around the world as a multi-camera video and surround sound mix. An interactive iPhone and Android application was also designed to accompany the simulcast experience. The design of this broadcast and mobile second-screen experience were guided by two primary conceptual challenges: how to privilege the remote audiences, and how to make a remote experience that needed to be connected to a live performance of *Powers*.

The first challenge was to explore how a simulcast experience can be more than the "cheap seats" version of a production. How can a simulcast offer an experience that serves as a counterpart or an additional model of experiencing the show, rather than a lesser replacement for the "real thing"? We decided that remote audience members should be privileged, given a glimpse into aspects of the show that the live audiences may not see. In fact, to connect these remote audiences to the storyline, we envisioned them as part of the pervasive System: others who had gone into the System and were thus granted many viewpoints. What might the show look like from the point of view of a robot, or from inside a wall, or from the Chandelier? How might it feel and look to be inside the System? The video content integrated into the broadcast incorporated these alternate points of view using cameras on stage, as well as special processing and distortion effects performed live on certain video streams.

As another part of granting the remote audience a privileged viewpoint inside the System, remote viewers received second-screen content on their mobile devices through a specially designed app. This content was primarily a language of light and color that echoed or complemented the behavior of the *Powers* walls. Frequently, the visuals on the second screen devices were designed to appear as small sections of the content on the walls, with the idea that many devices in a space together would serve to spread the imagery and behavior seen on the screen out across the audience, making each audience member a part of the System's visual presence.

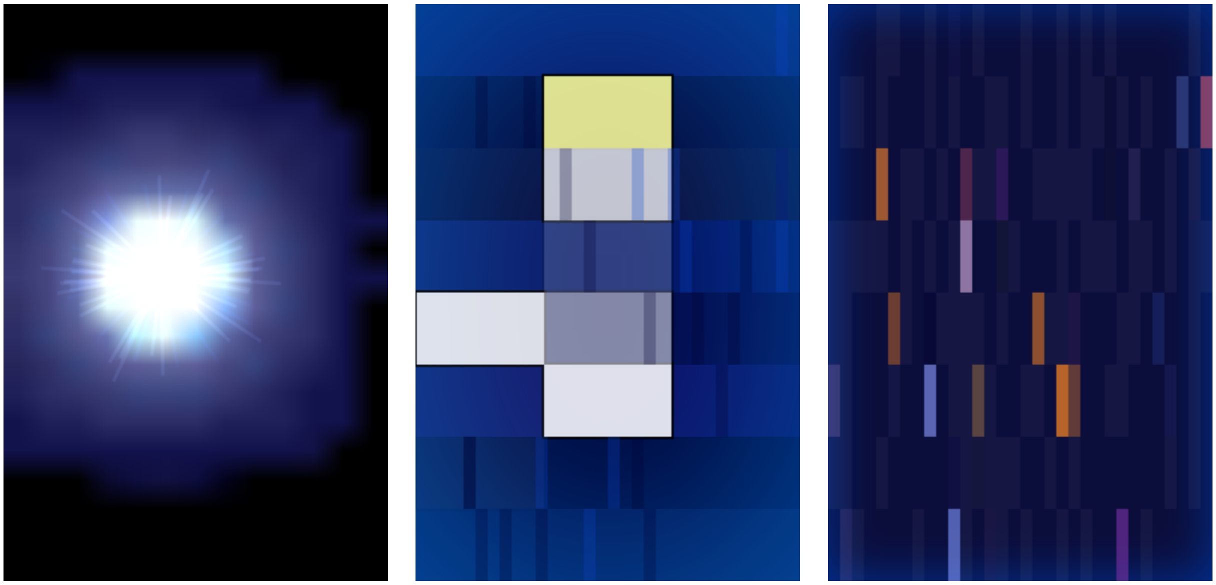


Figure 63. Screenshots from the Powers Live mobile application
At different points in the opera, the Powers Live app showed visualizations that moved and changed based on the live performance of the actor playing Simon Powers. Screenshots provided by Peter Torpey.

The second major conceptual challenge we addressed was creating a simulcast that actually had to be performed in sync with a live production. While the liveness of technology is always a question when augmenting performance (as discussed in Chapter 4), the stakes become even higher when working in a medium, such as a simulcast, where the physical presence of the performers is completely removed from that of the audience. If one envisions a simulcast that is only shown on a video screen, why does it matter that this video is currently broadcast live from a real show performed simultaneously? Would the experience actually be any different if that video had been taken at a real show the day before, or two months earlier? Indeed, there have been broadcasts of live performance separated in time from the original production, such as the February 2014 screenings of the Broadway production of *Romeo and Juliet* that closed in December 2013 (Isherwood, 2014). By incorporating interactive technologies into this simulcast, we sought to create an experience where it mattered that all of the remote audience members were having that experience simultaneously and in sync with the live production. At special moments in the show, audience members were guided to interact with their phones to send information about their participation back to the show. This information was aggregated to affect the behavior of elements in the Winspear and give a sense of the presence of the remote audiences.

A major new element incorporated for the Dallas production of *Death and the Powers* was the Moody Foundation Chandelier in the Winspear Opera House in Dallas. This chandelier features over 300 LED rods that can be individually color-controlled and positioned in groups to form different shapes. We connected this chandelier with the *Powers* show systems, using the chandelier to extend the show's language of light and color and movement out into the audience. This Moody Foundation Chandelier also served as the medium through which remote audiences could contribute to the show: at specific times, the remote audiences were guided through the visuals of their second screen experience to shake their devices or to touch their devices. This interactive information was fed back to the show networks and aggregated. We looked at how much audience members were in sync (with the downbeats of the music, at one point; with following the contour of a pitch displayed on their screen, in another) and modified the visuals on the chandelier to reflect the amount of synchronicity.

We chose to shape the large amount of new visual content on the Moody Foundation Chandelier and in the second screen experience by data from Simon Powers' live performance. As with the original elements of the show (the behavior of light and color on the LED walls, the *Powers* Chandelier, the movement of robotic elements, the sonic transformations), the behavior of the Moody Foundation Chandelier and the second screen content were affected and sculpted in real time, moment to moment, completely linked to the expressive movement and breath and voice of the performer. Peter Torpey designed the visual content on the chandelier and mobile devices, and we collaborated on creating new mappings from the input performance data to all of the new visual content using the Expressive Performance Extension System. These mappings took as input all of the original performance analysis features used in *Powers*, as well as special audience information captured only in the simulcast performance (such as the synchronous touch behavior of audience members, or what percentage of audience members were interacting with their device at a given moment).

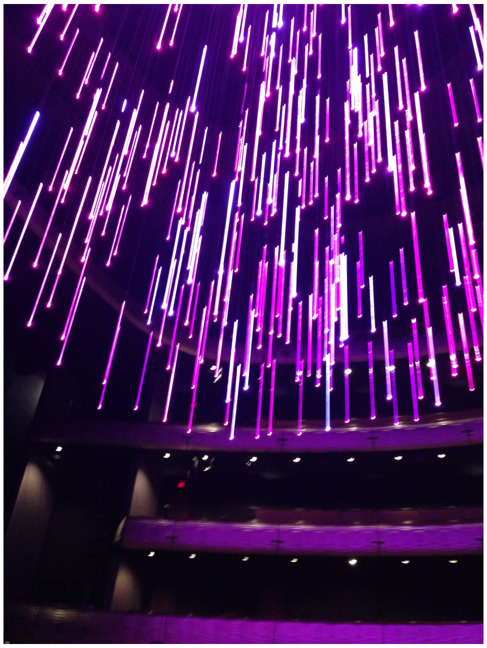


Figure 64. The Moody Foundation Chandelier The LED rods in the chandelier allowed light and movement to spill off the periaktoi set pieces and into the audience.

This usage of EPES did not incorporate the machine learning nodes, but leveraged several of the feature computation properties and other new features of the system, such as the visualization tools. This use case for the system was a major test of the system's ability to support rapid development and iteration of mappings. The majority of control mappings for the new chandelier and mobile content had to be created and tuned in real time during rehearsals, as there was a very limited amount of time to tune the new mapping content given live performance data, and no rehearsal time was allotted for this process. We developed a palette of initial mappings before the rehearsal process began, but had to adjust and modify them on the fly, as well as quickly add a variety of additional mappings as the visual content on the Moody Foundation Chandelier was developed and more content was added. Additionally, the interactive audience mappings had to be quickly tuned during the live performance. The final simulcast performance was our first and only opportunity to see the audience mobile interaction data measuring aggregate behavior from hundreds of users. While we had developed mappings and tested those mappings during performances and rehearsals with a handful of mobile devices, we did not have the opportunity to know what "real data" would look like when scaled. Given this, we put parameters into the mappings that could be immediately adjusted by hand during the performance to scale the data we were receiving into a range appropriate for the designed mappings. The system facilitated these rapid modifications even at performance time.

6.5. Summary of Example Projects and Implementation of Principles

This chapter has discussed the design and implementation of several expressive works for the voice and body that have incorporated versions of the Expressive Performance Extension System, including:

- The Powers Sensor Chair: a movement-based interactive musical installation for novices, exploring the sonic world of *Death and the Powers*
- Vocal Vibrations: a public installation for inspiring vocal exploration, incorporating interactive tactile feedback
- *Crenulations and Excursions/Temporal Excursions*: performances where qualities of the body and voice control a soundscape
- *Trajectories*: a short multimedia extended theatrical performance
- A series of interactive performances and installations developed by Blikwisseling workshop participants in the Netherlands.
- The *Death and the Powers* interactive global simulcast, including a interactive second-screen experience and visualizations on the Moody Foundation Chandelier

Each of these examples explores different features of the Expressive Performance Extension Framework and the Expressive Performance Extension System. The Powers Sensor Chair demonstrated the utility of a qualitative parametric model of physical expression in accommodating the movement vocabularies and explorations of a wide range of installation visitors. Vocal Vibrations showed the flexibility of the Expressive Performance Extension System throughout the rapid prototyping process for the installation, and tested the system's ability to incorporate multiple timescales of expressive parameters. *Temporal Excursions* demonstrated the utility of the system and frameworks for an instrumental model, combining movement and vocal analysis for performance extension in a way that was learnable and repeatable. *Trajectories* showed the utility of a shared model of expression in creating a performance that could be extended to control many existing output systems. The Blikwisseling workshop performances showed that the overall structure of creating mappings in EPES was flexible enough to be used quickly by various performance and installation creators. Finally, the *Powers* interactive global simulcast demonstrated the extension of a live performance in one space into media in distant locations through expressive representations of that performance.

This chapter has shown specific cases of the design process and framework for creating interactive performances and installations discussed in this dissertation. Through these concrete examples, this chapter has shown how the Expressive Performance Extension System has been able to integrate into the creative development process and final realization of a variety of different live interactive works, from expert performances to installations for novices. These examples have used different sensing systems, different expressive parametric spaces, and different output media. However, they have all been created with the same core interactive system.

7. Conclusion and Discussion

In this dissertation, I have introduced a new model for capturing and extending the power of the expressive body and voice. The Expressive Performance Extension Framework and its implementation in the Expressive Performance Extension System lay the groundwork of a novel technique for flexibly representing vocal and physical expression and extending that expression into digital media. These methodologies have important implications for expanding the power of a live performance or interactive installation across multiple spaces and multiple performance modalities. Additionally, as interactive technologies begin to address the body and voice as expressive elements, these frameworks and ways of thinking have applications beyond the context of live performance.

7.1. Primary Contributions

The Expressive Performance Extension Framework, the Expressive Performance Extension System, and the related discussion in this dissertation offer the following contributions:

- A conceptual framework and methodology for the use of machine learning technologies in extending expressive physical and vocal performance.
- An "instruction manual" for incorporating tools for performance extension into the creative process, including a collection of necessary questions for practitioners to address when working with machine learning for expressive performance extension in the context of a specific performance or installation.
- A new representation of physical and vocal expression through abstract, high-level expressive parametric spaces.
- A suggested set of parametric "quality" axes equally relevant for describing both vocal and physical expression.
- A framework that supports and prioritizes the continuous analysis of expression through regression algorithms, rather than through expression classification tasks.
- The implementation of a flexible software system for incorporating machine learning technologies into extended performances and installations.
- A systematic description of the expressive role of different elements of performance, particularly of various temporal scales of analysis.
- A catalogue of necessary principles for flexible analysis and mapping systems to be used in live performance and rehearsal contexts.
- A comparison of the strengths of computer systems and the strengths of human technicians for extending performance into digital media.
- Three primary performance and installation works that use the Expressive Performance Extension System to use qualities of movement and the voice for control of multimedia: the Powers Sensor Chair, Vocal Vibrations, and *Crenulations and Excursions/Temporal Excursions.*

I now return to the set of key questions for technological extension of physical performance that I proposed at the beginning of this dissertation, and examine the ways in which my research has addressed these questions.

How can raw sensor data be abstracted into more meaningful descriptions of physical and vocal expression? What features of physical performance can convey particular expressive and emotional content?

The Expressive Performance Extension Framework presented in this dissertation provides a unified methodology of analysis for different aspects of physical performance. Although movement and vocalization are both innately shaped by the physical properties of the body, these elements have not previously been considered together in frameworks for an expressive performance context. This framework includes computational features, sets of expressive parameters that can describe both body and vocal qualities, and methods for transforming raw sensor data into expressive parameters via machine learning techniques. An important aspect of this framework is its analysis of the different temporal windows that are appropriate to track different kinds of expressive events.

How can we create evocative high-level descriptions of movement and voice so that they can be used intuitively and creatively in the process of choreographing, composing, and performance-making?

I have outlined a set of recommended parameters for representing expressive qualities of movement and voice. This set of expressive parametric axes is sufficiently generic to describe both movement and vocal qualities, but constructed to represent a broad range of aspects of physical expression. This parameter set consists of *energy* (calm to energetic), *rate* (slow to quick), *fluidity* (legato to staccato), *scale* (small to large), *intensity* (gentle to intense), and *complexity* (simple to complex). These parameters are inspired by research in dance and vocal analysis, as well as by my own experience with extended performance systems and performance representations. I have also discussed methods of determining expressive parametric axes that are useful for describing a specific work and presented frameworks that are flexible enough to allow a creator to define and use whatever expressive parameters are most meaningful in a particular performance context.

How can we create tools that encourage metaphorical, meaningful, and rich associations between movement and media, rather than naïve one-to-one sensor to output mappings?

With the development of the Expressive Performance Extension System, I have demonstrated a concrete technological model that uses high-level descriptions of movement and vocal expression to enhance the behavior of a human performer or installation visitor. This system incorporates machine learning techniques for flexibly defining and working with abstract expressive parameters of movement and voice. It thus steps beyond standard models of gesture recognition or preprogrammed sets of specific continuous parameters that limit designers of performances or installations. This system has been incorporated into a variety of installation and performance contexts and tested in real-time scenarios. In this dissertation, I have also presented a variety of principles that are important for designing systems for meaningful performance extension, particularly systems that incorporate machine learning techniques for analysis of expressive qualities.

What principles should systems for performance extension follow in order to be easily incorporated into existing creative processes? What are good practices for extending live physical and vocal performance through machine learning techniques?

While this dissertation has centered on the use of machine learning systems for learning abstract parametric qualities of movement and voice, it has also explored the broader context of designing technologies that extend a live performance. Through the Expressive Performance Extension Framework, I have formalized a set of the core issues and questions to be addressed by a practitioner who seeks to create a technologically extended performance or installation, as well as defined requirements for systems to assist in performance extension. I have presented a workflow for augmenting a performance through machine learning of expressive parameters, exploring both the creative issues and the technical issues that are relevant at each stage. At every step in this workflow, core questions are presented for consideration by a performance creator. This work does not seek to present a fixed set of answers to these questions, as the majority of these answers will vary with every performance and installation depending on the artistic goals and constraints of the work. Instead, I seek to give practitioners a structure for thinking about performance extension technologies and a methodology for determining how best to integrate those technologies into the goals of their experience. Additionally, this framework provides the field of technological performance extension with methodologies and practices for incorporating high-level parameters and machine learning of movement qualities into existing performance practices.

This dissertation has also presented some of my contributions to the repertoire of augmented expression, including several performance and installation works for the extended body and the extended voice. The majority of my works presented here (including the Disembodied Performance System for *Death and the Powers*, the Gestural Media Framework and *Four Asynchronicities*, the *Powers* Global Simulcast, the Powers Sensor Chair, *Crenulations and Excursions/Temporal Excursions*, and Vocal Vibrations) have incorporated high-level parametric expressive spaces. The latter three works have also incorporated machine learning techniques for analyzing expressive movement and vocal qualities at various stages of the development process. For a range of different performances and installations, this dissertation has explored the development process of each piece, the artistic and design goals of each experience, and the use of the principles, frameworks, and systems presented in this dissertation.

- VAMP exemplified the power of well-mapped movement to create a compelling gestural instrument.
- The Gestural Media Framework and *Four Asynchronicities* showed that continuous concepts of movement quality were more interesting in performance than simple gesture recognition used to trigger events.
- The Disembodied Performance System for *Death and the Powers* extended a performer's vocal and physical expression into many simultaneous modalities while taking advantage of his virtuosic skillset.
- The *Sleep No More* Extension balanced the storytelling skills of human operators and an interactive content generation system.

- The Powers Sensor Chair demonstrated the utility of a qualitative parametric model of physical expression in accommodating the movement vocabularies and explorations of a wide range of installation visitors.
- Vocal Vibrations showed the flexibility of the Expressive Performance Extension System throughout a rapid prototyping process, and incorporated multiple timescales of expressive parameters.
- *Temporal Excursions* explored the utility of the dissertation system and frameworks for an instrumental model, combining both movement and vocal analysis for performance extension in a way that was learnable and repeatable.
- The *Powers* Global Simulcast explored extending a live performance across a variety of spaces and devices through an expressive representation of that performance.

7.2. Next Steps

For the further development of the Expressive Performance Extension System, there are various features to be implemented or extended that would add to the general applicability and ease of use of the system. Additionally, the concepts and systems developed as part of this dissertation lay the groundwork for addressing other challenges in the field of performance extension, as well as issues in domains beyond performances and installations. Formal representations of physical expression can be used for transformation into other modalities, for the persistence of performance expression, for modularity of creative tasks, for performance analysis, and even for the process of training performers or novices.

7.2.1. Next Stages for the Expressive Performance Extension System

As I continue to use and develop the Expressive Performance Extension System through live performance and installation contexts, I will continue to refine the necessary techniques for performance extension and the needs of the system to be increasingly integral to the rehearsal process, yet as invisible as possible.

First, there are improvements that could be made to the system that would make it even quicker to modify input and output devices. For example, it would be useful to be able to add additional input devices and output addresses on the fly via the GUI as the system is running, rather than manually in the show file or through additional properties files. The `GeneralOSCInput` device allows for some aspects of this behavior, but requires reloading the mapping file. More generic types of input devices, similar to the OSC inputs and Arduino inputs, would reduce the amount of coding necessary to use the system with a variety of sensing systems. Additionally, while the system currently utilizes a single output device that can send messages to multiple IP addresses and ports, it might be beneficial to create multiple output device nodes that can be associated with different addresses and ports so as to limit the amount of data flowing through the network.

The data flow structure of the Expressive Performance Extension System has great flexibility inherent in its feature computation step. More types of feature computation nodes can be created for the system, including techniques for automatic feature extraction, as well as nodes for particular feature computation techniques that have tunable settings to easily adapt to many different kinds of sensor inputs. Most of the types of computed features described in this thesis look at time windows of less than ten seconds, and do not compare the live performance with any predetermined score. Additional varieties of feature computation nodes could be created to provide feedback on variation from a recorded norm or a canonical form.

An interesting larger-scale addition to EPES would be to provide additional handles in the mapping system for creating more complex rule sets and conditional logic: do this thing X only if these three things ABC have already happened. This sort of performance logic is closer in concept to the JEML story logic engine designed for the *Sleep No More* Extension (discussed in Section 4.2.4). How does one construct a set of rules about what options are possible at any particular state of the performance, when those options depend on everything from prior actions to a performer's current behavior to the current section of the performance?

Integrating more backup features to the system would be useful in ensuring its stability across a variety of performance situations and potential technological failures. As the input and output systems become increasingly complex, backup systems are increasingly necessary. In the current EPES implementation, `Parameter` nodes can allow an external technician to have a layer of real-time control over a live performance in case of technical failure. Say the sensing system completely fails mid-performance: how can a technician monitoring the system step in and simulate some data? By attaching a `Parameter` in place of the output of the expressive parametric analysis, the technician can adjust some high-level values on the fly. This will not, of course, have the detail and expressivity of the live experience, but would likely be better (in many circumstances) then completely turning off the system and removing all variability from the digital media. However, what if the system also incorporated the ability to immediately switch from live data to some kind of saved performance data for playback? Performance data from a rehearsal could be recorded, and then run in loose temporal synchrony with the live performance, ready in case of emergency.

A few other potentially useful features to add to future versions of the system include:

- Additional techniques for representing and visualizing trajectories of a parameter over time.
- JavaScript nodes that incorporate a sense of state and time, for even more flexible experimentation.
- The ability to confine more information within one visual node through sub-mappings or encapsulation of groups of nodes (as in Max/MSP, Quartz Composer).
- Techniques for representation of relationships between multiple performers or performers in relationship to a performance space.

7.2.2. Further Extensions of Machine Learning and Feature Computation

The range of machine learning techniques built into the Expressive Performance Extension System could easily be extended in a variety of ways to help support the use of the system in a rehearsal process and to add further layers of mapping capabilities to the system. Additional machine learning algorithms for continuous parametric evaluation could be added and more settings could be made available to the user for adjusting variables of individual algorithms. It might also be useful to integrate the machine learning evaluation and training node classes into a single node for more ease in switching back and forth between training and testing in the middle of rehearsal.

So far, this research has focused on the extent to which expressive performance can be analyzed via continuous qualitative parameters rather than through gesture or speech recognition techniques. The system could later be extended to incorporate more gesture recognition capabilities and discrete classification tools. The existing structure of EPES can be used for recognition of gestures (identifying the probability of specific gestures) if thresholds are set for each output parameter of a regression algorithm, but a next step of complexity would be to incorporate structures designed directly for gesture recognition and switching system behavior based on gestures. For example: if the gesture recognized is raising the hand, map the expressive information in one way; if it is pushing the hand down and to the side, map the expressive information in a different way.

However, the usefulness of integrating such gesture recognition techniques will be significantly affected by improvements in algorithms' abilities to predict a given gesture. If the desired behavior of the system is to interpret expressive data differently depending on the current gesture, only recognizing the gesture once it has been completed will not be particularly helpful. When a performer begins to raise her arm, how quickly can a system detect that she is raising her arm and branch into the desired mappings? If the system is inaccurate in its prediction, how quickly can it tell that and recover gracefully? As audience members, we watch and make predictions about what we think will happen next. We find it meaningful when our predictions are satisfied or thwarted. Our interactive systems will also need to be able to go on that kind of journey.

Another area in which the Expressive Performance Extension System can be extended is by adding interfaces to allow quicker collection and labeling of training data examples. The current training data gathering phase requires the user to manually segment training data by starting and stopping the system for each example. Automatic sample segmentation techniques could be easily developed for the training phase. For example, what if the user wanted a longer sequence of movements to be automatically separated into multiple training data examples labeled as "very fluid" movements? This separation could be performed either by segmenting a data stream on particular features (pauses in movement or vocal silences, for example), or by storing many overlapping windows of data as individual samples.

Additionally, tools could be added to the Expressive Performance Extension System to allow for rapid live annotation of live or previously captured data streams. Imagine that overlapping windows of data were continually added to the training data set, with their labels determined by a slider that is moved by the user in real time while watching the performer being measured. A performance-maker could sit in rehearsal and annotate many training data examples on the fly, without having to take time from the rehearsal explicitly for capturing labeled examples. Since parameters can be trained separately, it is likely to be an achievable goal for the user to track the changing value of one parameter at a time. What if the same process could be performed while playing back a sequence of sensor data that has captured from a performance or rehearsal, synchronized to a video? This would allow the user to perform additional system training offline, without the performers needing to be present. This would also allow the user to create labeled training data for a variety of parameters given the same example sequence of movement.

This extended system could also be adapted to assist in the handling of weakly labeled data, where new training data examples from a performance or rehearsal could be labeled automatically by the system based on the parameter labels that it calculates given an existing set of training data. What if a choreographer has trained a system on one a particular piece, but wants to refine those definitions for another piece? Imagine that the system is run during a rehearsal to automatically label new training examples as described above. With a video system added to play back the rehearsal video in sync with this weakly labeled data, a user could observe the system's guesses at labeling the performance along the specified expressive axes and then modify the labels where the user does not agree with the system.

7.2.3. Creative Modularity Through Expressive Parameters

The structure of the Expressive Performance Extension system allows for a several levels of modularity in creating performance works. For example, training data sets and trained models can be reused in other performance or rehearsal contexts. Suppose a choreographer has already created one expressive model. He can start with the original set of input data and mappings, then keep the core definition of expression but change mappings. Alternately, he can keep the mappings between a particular expressive parametric space and a particular set of output media, but change the definition of what those expressive parameters mean or how they are measured. One person's definition of expression in performance could even be applied to someone else's piece.

This flexibility supports practitioners in developing certain aspects of their interactive space independently over the course of different performance pieces. For example, the soundscape that I created for *Crenulations and Excursions* had its roots in one of the sonic worlds developed for *Four Asynchronicities*, which was itself inspired by samples I used in an earlier choreographic piece. The same soundscape and triggering software was incorporated into *Trajectories* and extended even further for use in *Temporal Excursions*. Through the exploration of this sonic output vocabulary, I have grown and refined the soundscape and the `OSCtoMIDIGenerator` tool separately from the individual pieces or mappings. Similarly, one could expand and develop a particular parametric set or pool of training data over a variety of pieces.

The persistence of training data is a particularly interesting concept: it is not a score, not a mapping, and yet it is a carefully designed part of a piece. One could take the same sensors, computed features, training data, and expressive parameters, but create very different mappings with the same or different output media elements. There is some essential concept of expression that is created by the performance-maker and encoded in the labeling of the training data, but that expression does not dictate the overall shape of the piece, the elements of the temporal structure, or the vocabulary used in performance. Indeed, that expressive encoding may be reusable in many different contexts.

The Expressive Performance Extension Framework's focus on high-level parameters also encourages a different type of authorship of expressive interaction. An author of an interactive performance work or installation can create a piece that can be performed without a specific gestural vocabulary. The performers do not need to learn a particular choreography to interact with that particular creation, but the author still has a high level of control and shaping over the experience. There is a particular kind of artistry that is inherent in defining the expressive parameters and creating a mapping, which can remain constant even in a variety of different contexts. The Powers Sensor Chair is one example of a system that leveraged this strength of the Expressive Performance Extension System. This piece was designed to be on the installation side of the continuum of expertise discussed in Chapter 2, where those who encounter the instrument are novices at the experience. The system was able to support the creation of an instrument with a very particular character determined by its interaction design, but that did not have too many preconceptions about how it would be played.

This system's representation of expression through high-level parameter sets also allows for modularity in the division of labor around a performance or installation. It frees different members of a creative team to create different aspects of a piece, while keeping everyone focused on the same kind of expressive space. One set of parameters informs everyone's creative process. The choreographer may not need to know exactly how the movement may be extended into digital media, only that there is a particular parametric space to play with. The visual designer can begin creating interactive projections without yet knowing the specific choreography. A sound artist can compose a sonic piece that responds to particular high-level control parameters. All of this work can be developed simultaneously, connected and structured through a shared expressive vocabulary.

7.2.4. Expressive Extension in Other Performance Modalities

While the systems and frameworks discussed in this dissertation have been designed with the goal of describing and extending movement and the voice, these structures are also applicable for many other forms of performance. In particular, this framework's focus on regression rather than classification can be a general-purpose mapping concept for expressive performance. Similarly, the use of continuous expressive parameters as the core of a mapping is novel and generalizable to other performance domains.

For example, the same systems and workflow could be used in the development of a new extended musical instrument, such as an augmented instrument or an entirely new interface. Say that the desired instrument to be augmented was a piano. Performance data that could be sensed from this instrument might include the audio signal, which notes the performer plays (given a digital keyboard or some sensors on the keys of a traditional piano), how hard the performer hits the keys, which pedals are pressed, etc. Additional auxiliary data might come from sensors picking up the movement of the performer's upper body or head.

At the feature computation stage, many high-level musical features could be added, such as information about rhythmic structures, the deviation of a signal from a given rhythm, the variation of the tempo from moment to moment in relationship to a score, the amount of melodic tension, etc. These high-level and lower-level features could be used as inputs to an expressive parameter space of musical expression. While there are many different types of musical features that have been explored in expressive contexts, the final connection between these kinds of features and an expressive description of the performance is still subjective and would benefit from the sort of analysis explored in the Expressive Performance Extension Framework.

7.2.5. Performance Analysis and Training

The Expressive Performance Extension Framework and EPES can also be applied to the context of performance analysis. Regardless of whether a performance piece includes any technological performance extension techniques, the ability to measure aspects of the expressivity of that performance can still be used to study and analyze that performance in new ways. For example, imagine that a set of expressive curves are calculated throughout an entire performance, based on training examples. These curves are recorded in sync with video of that performance. By looking at the labels that now annotate the originally unlabeled performance, what information can be gained about the piece? What can we learn about the structure of the piece? We can use several labeled performances of the same piece to learn more about how that piece varies between performances, or how different performers vary in their interpretation of the same piece.

Similarly, these techniques can be useful as training tools. In the field of dance, transmission of the details of a particular piece is typically performed through a few limited strategies: direct training from an original performer, video, and/or notation systems. A system like EPES could easily be modified to serve as a complement to typical dance notation systems or video recordings. Could a performer learn not only the choreographic vocabulary of a piece but also the expressivity of its original performance? Could he learn not only the notes of the song but also the expressive vocal shaping? What if he could be evaluated by the system and get feedback about how accurate he was in reproducing an expressive model?

To do these sorts of training and analysis tasks, the system could be reasonably easily extended to record the values of a set of expressive parameters of an original performer, measure a new performer's expression along the same parameters, and compare the two sets of metrics. In cases where the overall timing of a piece is constrained by a musical recording (in the case of a dance performance to a specific piece of music), this comparison is straightforward. For performances where the timing varies more widely, some normalization of time across different sections would be necessary. A performer could then be given feedback about his or her performance either in real time or as a summary afterwards.

As a performer's expression starts to be used to affect more aspects of a piece, this kind of training may be particularly useful. For example, let us examine the Dallas performances of *Death and the Powers* and the replacement of our original baritone James Maddalena with Robert Orth in the role of Simon Powers. While the creative team chose not to dictate too many aspects of Orth's performance, there was a certain amount of consistency we hoped to obtain from the many visualizations and sound manipulations that would be affected by the performance of whoever was in the role of Simon Powers. We did not want to show Maddalena or Orth the direct visual results of their performance, lest they get in a feedback loop of attempting to make the system respond in a particular visual way. But what if the creative team could examine some representation of Orth's expressive performance, compare it to a representation of Maddalena's original performance, and through that discover ways to direct Orth's performance?

Many other interesting questions arise when discussing the analysis of performance through expressive parameters. What is the canonical form of a particular dance or vocal piece, and how can that form be shared? Is that form purely the sequence of choreographic motions or a musical score, or does it involve some concept of a particular performer's expressivity? Could someone learn the choreographic or performance style of a particular performer, given metrics of their expression? Do particular performers have an expressive signature that could be analyzed through these kinds of systems?

On a related point, how can a system train people who are not yet expressive with their voices or bodies to be more expressive? The Vocal Vibrations project has begun to explore this field, with the goal of designing an experience that would inspire novices to explore the possibilities of their voices. What if you could get direct feedback from a system about the dimensions of expressivity of your voice? This kind of feedback about your own parameters of expression could be used not only for training purposes, but also for increasing your general awareness of your own expressive patterns. What are your styles of expression, given a particular vocabulary of movement or song?

Current interfaces or games that give feedback on singing, such as the Rock Band game, generally focus on giving the user feedback about how their pitch compares over time to the desired pitch. Interfaces for novices that measure and give feedback on physical movement, like the game Dance Central, similarly look at only whether your body is in the correct position at the correct moment. What if the question was not whether you could match the pitch or movement correctly at a specified time, but whether could you match a defined expressive arc of a piece? What if you were freed from using a specific vocabulary, but instead allowed to explore a range of vocabularies and styles while examining how those affected your performance? Or what if, while learning a particular song or dance piece, your focus was on learning the expressive shape of the piece in addition to (or even before) precisely learning the notes and movements?

7.2.6. Expressive Extension in Other Domains

The concepts and systems developed as part of this dissertation lay the groundwork for addressing problems of expressive representation in domains beyond extended performances and installations. For example, expressive gestural or vocal input could be used as parametric input into compositional tools. What might it mean to tell a system that you want the next part of the piece to be "like this," where "this" is a time-varying quality of movement, or a vocal phrase? How could compositional parameters be shaped by example? In systems such as Torpey's Media Scores (Torpey, 2013), the shape of compositional elements over time can be entered via an expressive drawing interface. What if the desired quality of a particular moment could be communicated to a system through a movement of the hand, or a vocalization?

In the field of Human-Computer Interaction, researchers have primarily focused on detecting individual gestures and words, with the interpersonal generalization goal of having one system be able to recognize many users performing the same gesture or saying the same text. As we have seen in this thesis and the related research, variation in movement and vocal examples is significant for expressive or emotional communication. There is a broad push toward detecting users' emotions, although generally the emotions detected are users' innate emotions, rather than emotions that the users are intending to express to the system. However, systems are not often created to allow interface designers to construct systems at a high level with that kind of information. Say you know how happy or energetic or calm or abrupt a user is: then what should be done with that information? The easier it is to obtain high-level representations of the expressive and emotional content of a user's behavior, the more that designers can design their interfaces to be shaped by those representations.

More broadly, this exploration of expressive movement and voice technologies will be relevant for designers of many different kinds of immersive experiences. As discovered with the Powers Sensor Chair, even though the world has become generally saturated with technology, people still do not expect their own movement to cause an effect in a space. Similarly, Vocal Vibrations showed how deeply visitors connected with an experience that was centered on their own voice. Movement and voice are two of the most personal, unique, and expressive ways that we can communicate with the world. Thus, using the body and the voice expressively can allow people to have very personal encounters with a space or with a technology.

7.3. The Future of Extended Performance

This chapter has shown how the work discussed in this dissertation has achieved the research goals outlined in the introduction: a creative framework and design principles for performance extension through technology, a computational system for high-level analysis and mapping of physical and vocal qualities, and several performances and installations that contribute to the repertoire of extended expression. These tools and principles support the main research question of how to transform movement and voice data into meaningful, expressive information. As the field of performance becomes increasingly technological, these kinds of methodologies will be necessary to help that field evolve in ways that support the humans at the center of the performance traditions.

7.3.1. The Multimodal Performance

We are now at a point where different performance disciplines are merging, creating hybrid performance experiences that cannot be classified solely as works of "dance" or "music" or "theater." We are perhaps moving closer to Wagner's concept of *Gesamtkunstwerk*, a model of a "unified artwork" that would combine all forms of art via the theater (Wagner, 1895). Interactive technologies are the next element that can be brought into these rich, boundary-defying performances. Beyond serving as an individual element that can be added to a performance, technology can also support the expressive integration and unification of many other modalities. An increasing number of theatrical systems are already under some level of technological and computational control, from stage lighting to sound manipulation to the movement of set pieces to projections. This gives us the potential to connect these different forms of media into unified expressive systems. However, the ability to connect these many systems in a live performance has so far generally been limited to envisioning that particular actions will occur simultaneously. The connection between multiple media elements is time, either in the form of shared timecode or shared cues. The media elements are composed in reference to time and perhaps a set of actions represented through a script or score.

Creating performances that integrate many types of media may become easier through the model of a shared representation of performance that can be extended into a variety of modalities. What are the possibilities when all of these systems can be linked to the expressive performance of an actor, dancer, or singer, to become completely connected into one multimodal experience? The opera *Death and the Powers* is an early example of such a performance extension, where visualizations, sound, and robotic movement of scenic elements are connected through the live behavior of the lead actor. Tools that capture and represent expressive performance in meaningful ways may be the key to creating these multimodal connections.

Through the use of a central vocabulary of expressive parameters, different members of the creative team for a particular piece can collaborate around a shared vision of what is important about a particular live performance. Visuals, sound, music, dance, scenic elements, and theatrical performance can be directly connected, with one computational model of expression linking many aspects of the performance in real time. An expressive quality of a performer's body can become a musical quality, a quality in an artistic visualization, and a quality of scenic lighting. The voice can transform the robotic scenery or the soundscape. Through careful design of mappings and mapping systems, all of these media can come together into a single expressive experience.

7.3.2. The Multi-Spatial Performance

Additionally, as performances stretch outside of a single performance space, the question of extending expression becomes even broader. For an experience like the *Sleep No More* Extension's online component, or the *Powers* Global Interactive Simulcast, what does it mean to have a live performance connected to experiences that are located at an extreme distance from the theater? How can something on a mobile device feel like it resonates with a live performer's actions thousands of miles away? How can an experience taking place online feel like it is connected expressively with a real performance? How can multiple performance spaces in different locations be connected together to create larger performance experiences? High-level representations of a performer's physical expression will assist performance-makers in transforming that expression not only into other modalities within the theater, but also into modalities that can allow that expression to be experienced remotely in a variety of ways.

Modeling a performer's expressivity with regards to space and abstracting expression away from the form of a particular space may be important components in translating that performance across spaces. The majority of the systems described in this dissertation have focused on the expressive qualities of a performer within his or her personal kinesphere. New parameters of expression could be added and trained on features of a performer's physical relationship to the performance space or his relationship to other performers within a space, examining spatial connections such as those explored in Section 4.1.4. Once there is a model of the performer's expressivity with regards to the performance space, can we transmit a sense of this connection in order to link very different types of spaces? A performer in relationship to one particular space evokes a particular feeling. That feeling can be transmitted and then transformed into elements that evoke the same feeling in the context of another space (a different theater, a cinema, a tiny room, an outdoor location, a giant warehouse, a specially architected space, an online world, a mobile screen...).

It is important to note that in these cases where a performance is sent from one space to a variety of other spaces or transformed into alternate modes of experience, having a high-level representation of expression provides a compact method of communication between the main performance venue and other venues. The analyzed expressive performance information can be the primary information transmitted, rather than a large amount of raw sensor data. This performance information can then be interpreted in individual ways by each of the different spaces receiving that information.

7.3.3. The Human Performance

As we continue to incorporate technology into these multimodal, multi-spatial performances and installations, it is vital that we keep our focus on the human at the center, the most important part of a performance. Our systems should both serve the goals of a particular performance work and extend the nature of live, human, performance. Is it clear that there is live control, affected by moment-to-moment variation of a performer, different every night? Is the technology a core part of the storytelling and the performance experience, rather than being included for the sake of the technology? Technology in performance too often takes the form of giant projection screens with content divorced from the live performers, or giant speakers with sound distant from a performer's actions. Our technologies need to support the performers, rather than overwhelm them.

Imagine, in a time not far from now, that a performance is taking place. At first, when the audience arrives, they find a large empty space. As the audience members start to tentatively walk around the room, they find that some level of their nervous movement is echoed in the space by thin threads of light that agitatedly shift across the walls. A woman smiles and waves her hand rapidly, and one thread of light playfully darts around the space. As other audience members pick up her movements, walk with different tempos, or gather together in clusters, patterns start to emerge from the strands of light. The lines shift, becoming more confident, complex, and smooth as the audience relaxes and tries different kinds of movements. Suddenly, a man dressed all in white enters the room and holds up his hand, suspended. The threads of light cluster behind him into an excited ball. With a sharp flick of his hand, the light scatters around the walls and a drone slowly becomes audible. The man begins to sing a sad, wordless melody, the emotion in his voice reflected through the threads of light dimming and drooping. He starts to pace through the space, passing audience members and momentarily interacting with some of them. Their reactions to his encounters create ripples of light through the space, some startled, some excited, some calm. As the performance goes on, the space continues to envelop the audience in light and sound, reflecting and enhancing the performer's actions and emotions as well as the audience's own behavior. The audience members do not care about how this experience is implemented. Instead, they are captivated and brought into the story unfolding before them.

The theater has the ability to make magic out of thin air, out of nothing more than talented people onstage with perhaps scraps of paper or a piece of fabric or some lights or a piece of furniture. Technologies need to become a fully integrated part of the magic of a complete performance experience, in the expressive service of the performer and the performance-creators. That, after all, is the fundamental goal for a future of technological performance: systems that are true extensions of and complements to a live performance, by recognizing and responding to subtleties of timing, articulation, and expression that make every performance fundamentally unrepeatable and unique.

8. References

Abe, S. (2005). *Support Vector Machines for Pattern Classification*. New York, NY: Springer.

Anagnostopoulos, C. N., Iliou, T., & Giannoukos, I. (2012). Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review*.

Avilés-Arriaga, H. H., & Sucar, L. E. (2002). Dynamic Bayesian networks for visual recognition of dynamic gestures. *Journal of Intelligent and Fuzzy Systems*, *12*(3), 243–250.

Aylward, R., & Paradiso, J. A. (2006). Sensemble: a wireless, compact, multi-user sensor system for interactive dance. *NIME '06: Proceedings of the 2006 Conference on New Interfaces for Musical Expression*.

Bahl, L. R., Jelinek, F., & Mercer, R. (1983). A Maximum Likelihood Approach to Continuous Speech Recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2), 179–190. doi:10.1109/TPAMI.1983.4767370

Baker, J. K. (1975). *Stochastic Modeling as a Means of Automatic Speech Recognition*. Carnegie Mellon University.

Banse, R., & Scherer, K. R. (1996). Acoustic Profiles in Vocal Emotion Expression. *Journal of Personality and Social Psychology*, *70*, 614–636.

Benbasat, A. Y. (2000). *An Inertial Measurement Unit for User Interfaces*. Massachusetts Institute of Technology.

Benford, S., Schnädelbach, H., Koleva, B., Anastasi, R., Greenhalgh, C., Rodden, T., et al. (2005). Expected, sensed, and desired: A framework for designing sensing-based interaction. *Transactions on Computer-Human Interaction (TOCHI*, *12*(1). doi:10.1145/1057237.1057239

BigEye | STEIM. (n.d.). BigEye | STEIM. *Steim.org*. Retrieved June 8, 2014, from http://steim.org/2012/01/bigeye-1-1-4/

Billon, R., Nédélec, A., & Tisseau, J. (2008). Gesture recognition in flow based on PCA analysis using multiagent system. *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, 139–146.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bokowiec, M. A. (2011). V'oct (ritual): An interactive vocal work for bodycoder system and 8 channel spatialization. *Proceedings of the International Conference on New Interfaces for Musical Expression*.

Bokowiec, M. A., & Bokowiec, J. (2005). The Suicided Voice: The Mediated Voice. *Proceedings of the International Computer Music Conference*.

Bongers, B. (2000). Physical interfaces in the electronic arts. *Trends in Gestural Control of Music*.

Brook, P. (1996). *The Empty Space*. Simon and Schuster.

Brown, A. K., & Parker, M. (1984). *Dance notation for beginners*. Princeton Book Co Pub.

Cahn, J. E. (1990). The Generation of Affect in Synthesized Speech. *Journal of the American Voice I/O Society*.

Campbell, N., & Mokhtari, P. (2003). Voice quality: the 4th prosodic dimension. *15th ICPhS*, 2417–2420.

Camurri, A., Canepa, C., Coletta, P., Ferrari, N., Mazzarino, B., & Volpe, G. (2008). Social active listening and making of expressive music: the interactive piece the bow is bent and drawn. *DIMEA '08: Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*. doi:10.1145/1413634.1413701

Camurri, A., De Poli, G., Leman, M., & Volpe, G. (2001). A multi-layered conceptual framework for expressive gesture applications. *Proceedings of the International MOSART Workshop*.

Camurri, A., Hashimoto, S., Suzuki, K., & Trocca, R. (1999). Kansei analysis of dance performance. *1999 IEEE International Conference on Systems, Man, and Cybernetics*, *4*, 327–332.

Charles, J.-F. (2008). A tutorial on spectral sound processing using max/msp and jitter. *Computer Music Journal*, *32*(3).

Clynes, M. (1977). *Sentics: the Touch of the Emotions*. New York, NY: Doubleday and Co.

Coniglio, Mark. (2004). The importance of being interactive. *New Visions in Performance: the Impact of Digital Technologies*, 5–12.

Cottin, R. (n.d.). *Laban Effort Graph*. Retrieved from http://commons.wikimedia.org/wiki/File:Laban-effort-graph.jpg

Credo Interactive Inc. (n.d.). Credo Interactive Inc. *Credo-Interactive.com*. Retrieved June 8, 2014, from http://www.credo-interactive.com/gallery/index.html

Cromer, D. (Ed.). (2013, January). *Our Town*. Boston: Huntington Theatre.

d'Alessandro, N., Babacan, O., Bozkurt, B., Dubuisson, T., Holzapfel, A., Kessous, L., et al. (2008). RAMCESS 2.X framework—expressive voice analysis for realtime and accurate synthesis of singing. *Journal on Multimodal User Interfaces*, *2*(2), 133–144. doi:10.1007/s12193-008-0010-4

David Rokeby: Very Nervous System. (n.d.). David Rokeby: Very Nervous System. *Davidrokeby.com*. Retrieved June 8, 2014, from http://www.davidrokeby.com/vns.html

Death and the Powers. (n.d.). *Death and the Powers*. Retrieved June 7, 2014, from http://powers.media.mit.edu

Death and the Powers DVD (in progress). (2014). *Death and the Powers DVD (in progress)*.

Delaumosne. (1893). Delsarte system of oratory.

Dixon, S. (2007). *Digital performance*. The MIT Press.

Downie, M. N. (2005). *Choreographing the Extended Agent*. Massachusetts Institute of Technology.

Efron, D. (1972). *Gesture, Race and Culture*. The Hague: Mouton.

Eickeler, S., Kosmala, A., & Rigoll, G. (1998). Hidden Markov model based continuous online gesture recognition. *Fourteenth International Conference on Pattern Recognition, 1998 Proceedings*, *2*, 1206–1208. doi:10.1109/ICPR.1998.711914

Fagerberg, P., Ståhl, A., & Höök, K. (2003). Designing gestures for affective input: an analysis of shape, effort and valence. *Proceedings of the 2nd International Conference on Mobile Ubiquitous Multimedia*.

Fernandez, R. (2004). *A Computational Model for the Automatic Recognition of Affect in Speech*. Massachusetts Institute of Technology.

Fiebrink, R. A. (2011). *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. Princeton University.

Frick, R. W. (1985). Communicating emotion: The role of prosodic features. *Psychological Bulletin*.

g-speak - oblong industries, inc. (n.d.). g-speak - oblong industries, inc. *Oblong.com*. Retrieved June 8, 2014, from http://www.oblong.com/g-speak/

Gillian, N. E. (2011). Gesture recognition for musician computer interaction. *Queen's University Belfast*.

Gillian, N., Knapp, R. B., & O'Modhrain, S. (2011). A machine learning toolbox for musician computer interaction. *Proceedings of the International Conference on New Interfaces for Musical Expression*.

Godøy, R. I., & Leman, M. (2009). *Musical Gestures*. New York: Routledge.

Grichkovtsova, I., Morel, M., & Lacheret, A. (2012). The role of voice quality and prosodic contour in affective speech perception. *Speech Communication*, *54*(3), 414–429. doi:10.1016/j.specom.2011.10.005

Gritten, A., & King, E. (2006). *Music And Gesture*. Hampshire: Ashgate Publishing, Ltd.

Guest, A. H. (1984). *Dance Notation*. Dance Books.

Gunther, E. (2001). *Skinscape: A tool for composition in the tactile modality*. Massachusetts Institute of Technology.

Hatten, R. (2006). A Theory of Musical Gesture and its Application to Beethoven and Schubert. In A. Gritten & E. King, *Music and Gesture* (pp. 1–23). Hampshire: Ashgate.

Heaton, J. (2008). *Introduction to Neural Networks with Java*. Heaton Research, Inc.

Hewett, I. (2008, March 25). *Lost Highway:* Into the Dark Heart of David Lynch. *Telegraph*.

Hodgson, J. (2001). *Mastering Movement*. Psychology Press.

Hunt, A., Wanderley, M. M., & Paradis, M. (2002). The importance of parameter mapping in electronic instrument design. *Proceedings of the 2002 Conference on New Instruments for Musical Expression, Dublin, Ireland*.

Isherwood, C. (2014, January 23). Off Off Off Broadway (at Your Multiplex). *The New York Times*.

Itakura, F. (1990). Minimum prediction residual principle applied to speech recognition. *Readings in Speech Recognition*.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. MIT Press.

Jessop, E. (2009). The Vocal Augmentation and Manipulation Prosthesis (VAMP): A Conducting-Based Gestural Controller for Vocal Performance. *Proceedings of NIME 2009*.

Jessop, E. N. (2010). *A Gestural Media Framework*. Massachusetts Institute of Technology.

Jessop, E., Torpey, P. A., & Bloomberg, B. (2011). Music and Technology in Death and the Powers. *New Interfaces for Musical Expression*.

Juslin, P. N. (2003). Five Facets of Musical Expression: A Psychologist's Perspective on Music
Performance. *Psychology of Music*, *3*(3), 273–302.

Kendon, A. (2004). *Gesture*. Cambridge University Press.

Kim, Y. E. (2003). *Singing Voice Analysis/synthesis*. Massachusetts Institute of Technology.

Knapp, B. (1992). BIOMUSE: Musical Performance Generated by Human Bioelectric Signals. *Center for Computer Research in Music and Acoustics*.

Knapp, R. B., & Cook, P. R. (2005). The Integral Music Controller: Introducing a Direct Emotional Interface to Gestural Control of Sound Synthesis. *Proceedings of the International Computer Music Conference (ICMC)*, 4–9.

Ko, T., Demirdjian, D., & Darrell, T. (2003). Untethered gesture acquisition and recognition for a multimodal conversational system. *Proceedings of the 5th International Conference on Multimodal Interfaces*, 147–150.

Kurtenbach, G., & Hulteen, E. (1990). Gestures in Human-Computer Communication. In B. Laurel, *The Art and Science of Interface Design* (pp. 309–317). Reading, MA: Addison-Wesley Publishing Co.

Laban, R. (1980). *Mastery of Movement* (4 ed.). Northcote House.

Ladd, D. R. (1985). Evidence for the independent function of intonation contour type, voice quality, and F0 range in signaling speaker affect. *The Journal of the Acoustical Society of America*, *78*(2), 435. doi:10.1121/1.392466

Laver, J. (1980). The Phonetic Description of Voice Quality. *Cambridge Studies in Linguistics*

London, *31*, 1–186.

Lee, M., Freed, A., & Wessel, D. (1992). Neural networks for simultaneous classification and parameter estimation in musical instrument control. *Aerospace Sensing*, 244–255.

Levin, G., & Lieberman, Z. (2004). In-situ speech visualization in real-time interactive installation and performance. *NPAR '04: Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*. doi:10.1145/987657.987659

Levitin, D. J., MacLean, K., Mathews, M., & Chu, L. (2000). The perception of cross-modal simultaneity. *International Journal of Computing Anticipatory Systems*.

Lewis, G. (2007). The Virtual Discourses of Pamela Z. *Journal of the Society for American Music*, *1*(01), 57–77. doi:10.1017/S1752196307070034

Machart, R. (2010, September). Wagner goes digital at the New York Met. *The Guardian*.

Machover, T. (1992). *Hyperinstruments*. Massachusetts Institute of Technology.

Machover, T. (2004). Shaping Minds Musically. *BT Technology Journal*, *22*(4).

Machover, T. (2010). Death and the Powers. New York: Boosey and Hawkes.

Machover, T. (2014a). *Chapel Music*. *Vocal Vibrations*.

Machover, T. (2014b). *Cocoon Music*. *Vocal Vibrations*.

Maes, P.-J., Leman, M., Kochman, K., Lesaffre, M., & Demey, M. (2011). The “one-person choir”: A multidisciplinary approach to the development of an embodied human-computer interface. *Computer Music Journal*, *35*(2).

Maestre, E., Bonada, J., & Mayor, O. (2006). Modeling musical articulation gestures in singing voice performances | Music Technology Group. *Proceedings of the AES 121st Convention*.

Martin, J. (1933). *The Modern Dance*. Princeton, NJ: Princeton Book Company.

Mathews, M. (1991). The Radio Baton and Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music. *Computer Music Journal*, *15*, 37–46.

Max « Cycling 74. (n.d.). Max « Cycling 74. *Cycling74.com*. Retrieved 2014, from http://cycling74.com/products/max/

Mazo, J. (1977). *Prime Movers: The Makers of Modern Dance in America* (2nd ed.). Hightstown: Princeton Book Company.

McBride, S. (1997). Sing the Body Electronic: American Invention in Contemporary Performance. *Sycamore: a Journal of American Culture*, *1*(3).

McCaw, D. (Ed.). (2011). *The Laban Sourcebook*. New York, NY: Routledge.

McNeill, D. (1992). *Hand and Mind*. Chicago, IL: University Of Chicago Press.

Merrill, D., & Paradiso, J. A. (2005). Personalization, expressivity, and learnability of an implicit mapping strategy for physical interfaces. *Proceedings of the CHI Conference on Human Factors in Computing Systems, Extended Abstracts. 2005.*

Mestres, J. J. (2008). *Singing-driven Interfaces for Sound Synthesizers*. Universitat Pompeu Fabra of Barcelona.

Mitchell, T. J. (2011). Soundgrasp: A gestural interface for the performance of live music.

Modler, P., Myatt, T., & Saup, M. (2003). An experimental set of hand gestures for expressive control of musical parameters in realtime. *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, 146–150.

Nakra, T. A. M. (2000). *Inside the Conductor's Jacket*. Massachusetts Institute of Technology.

Nam, Y., & Wohn, K. (1996). recognition of space-time hand-gestures using hidden markov model. *ACM Symposium on Virtual Reality Software and Technology*, 51–58.

Oliver, W. D. (1997). *The Singing Tree*. Massachusetts Institute of Technology.

Oliverio, J., & Pair, J. (1998). Design and implementation of a multimedia opera.

opensoundcontrol.org. (n.d.). opensoundcontrol.org. *Opensoundcontrol.org*. Retrieved 2014, from http://opensoundcontrol.org/

Overholt, D., Thompson, J., Putnam, L., Bell, B., Kleban, J., Sturm, B., & Kuchera-Morin, J. (2009). A multimodal system for gesture recognition in interactive music performance. *Computer Music Journal*, *33*(4), 69–82.

Paradiso, J. A. (1999). The brain opera technology: New instruments and gestural sensors for musical interaction and performance. *Journal of New Music Research*, *28*(2), 130–149.

Paradiso, J. A., & Gershenfeld, N. (1997). Musical applications of electric field sensing. *Computer Music Journal*.

Pliam, S. L. (2007). *The Chandelier: towards a digitally conceived physical performance object*. Massachusetts Institute of Technology.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Rabiner, L. R. (1993). *Fundamentals of Speech Recognitions*. Prentice Hall.

Ramakrishnan, C., Freeman, J., & Varnik, K. (2004). The architecture of auracle: a real-time, distributed, collaborative instrument. *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, 100–103.

Ricci, A., Suzuki, K., Trocca, R., & Volpe, G. (2000). Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*.

Robert R. Morris: Gesture Guitar. (n.d.). *Robert R. Morris: Gesture Guitar*. Retrieved June 2014, from http://www.robertrmorris.org/gestureguitar.html

Rowe, R. (1993). *Interactive Music Systems*. MIT Press (MA).

Rowe, R. (2004). *Machine Musicianship*. MIT Press.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*.

Saffer, D. (2008). *Designing Gestural Interfaces*. O'Reilly Media, Inc.

Sakoe, H. (1979). Two-level DP-matching--A dynamic programming-based pattern matching algorithm for connected word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, *27*(6), 588–595. doi:10.1109/TASSP.1979.1163310

Sargin, M. E., Aran, O., Karpov, A., Ofli, F., Yasinnik, Y., Wilson, S., et al. (2006). Combined Gesture-Speech Analysis and Speech Driven Gesture Synthesis. *2006 IEEE International Conference on Multimedia and Expo*, 893–896. doi:10.1109/ICME.2006.262663

Scherer, K. R. (1986). Vocal affect expression: A review and a model for future research. *Psychological Bulletin*, *99*(2), 143–165. doi:10.1037/0033-2909.99.2.143

Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008). Gesture recognition with a Wii controller. *The 2nd International Conference*, 11. doi:10.1145/1347390.1347395

Siegel, W., & Jacobsen, J. (1998). The challenges of interactive dance: An overview and case study. *Computer Music Journal*, *22*, 29–43.

Soerensen, E. B., & Lyng, T. (2005, December). How long does the subject linger on the edge of the volume. *Www.Artificial.Dk*. Retrieved 2014, from http://www.artificial.dk/

Sparacino, F. (1996). Directive: Choreographing media creatures for interactive virtual environments.

Sparacino, F., Wren, C., Davenport, G., & Pentland, A. (1999). Augmented performance in dance and theater. *International Dance and Technology*, *99*, 25–28.

Starner, T., Weaver, J., & Pentland, A. (1997). A wearable computer based American sign language recognizer. *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, 130–137. doi:10.1109/ISWC.1997.629929

Stoppiello, D., & Coniglio, M. (2003). Fleshmotor. In J. Malloy, *Women, Art, and Technology* (pp. 440–452). Cambridge, MA: MIT Press.

Stowell, D. (2010). Making music through real-time voice timbre analysis: machine learning and timbral control. *Queen Mary University of London*.

Strachan, S., Murray-Smith, R., & O'Modhrain, S. (2007). BodySpace. *CHI '07 Extended Abstracts*, 2001–2006. doi:10.1145/1240866.1240939

The Gloves Project. (n.d.). *The Gloves Project*. Retrieved June 8, 2014, from http://theglovesproject.com/

The Magic of NeoPixels | Adafruit NeoPixel Überguide | Adafruit Learning System. (n.d.). The Magic of NeoPixels | Adafruit NeoPixel Überguide | Adafruit Learning System. *Learn.Adafruit.com*. Retrieved July 20, 2014, from https://learn.adafruit.com/adafruit-neopixel-uberguide

Thibodeau, J., & Wanderley, M. M. (2013). Trumpet Augmentation and Technological Symbiosis. *Computer Music Journal*, *37*(3), 12–25. doi:10.1080/09298210500124208

Torpey, P. A. (2009). *Disembodied Performance*. Massachusetts Institute of Technology.

Torpey, P. A. (2012). Digital systems for live multimodal performance in Death and the Powers.

International Journal of Performance Arts and Digital Media.

Torpey, P. A. (2013, August). *Media Scores*. Massachusetts Institute of Technology.

Torpey, P. A., & Jessop, E. N. (2009). Disembodied performance. *CHI EA "09: CHI "09 Extended Abstracts on Human Factors in Computing Systems*. doi:10.1145/1520340.1520555

TOTEM Set Design and Projections. (n.d.). TOTEM Set Design and Projections. *Www.Cirquedusoleil.com*. Retrieved June 8, 2014, from http://www.cirquedusoleil.com/en/~/media/press/PDF/totem/TOTEM_Set_Projections.pdf

TROIKATRONIX | live performance tools. (n.d.). TROIKATRONIX | live performance tools. *Troikatronix.com*. Retrieved June 8, 2014, from http://troikatronix.com/

Ursonography - Interactive Art by Golan Levin and Collaborators. (n.d.). Ursonography - Interactive Art by Golan Levin and Collaborators. *Flong.com*. Retrieved June 8, 2014, from http://www.flong.com/projects/ursonography/

Vickery, L. R. (2002). The Yamaha MIBURI MIDI jump suit as a controller for STEIM's Interactive Video software Image/ine. *Proc Australian Computer Music Conference*.

Volpe, G. (2003). Computational models of expressive gesture in multimedia systems. *InfoMus Lab, DIST–University of Genova*, *12*.

Wagner, R. (1895). The Art-Work of the Future and Other Works.

Waibel, A., & Lee, K.-F. (1990). *Readings in Speech Recognition*. Elsevier.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, *37*(3), 328–339. doi:10.1109/29.21701

Waisvisz, M. (1985). The Hands, a Set of Remote Midi-Controllers. *Proceedings of the International Computer Music Conference*, 313–318.

Wakin, D. (2008, November 7). Techno-Alchemy at the Opera: Robert Lepage Brings his "Faust" to the Met. *The New York Times*.

Wakin, D. (2010, September 19). The Valhalla Machine. *The New York Times*.

Wanderley, M. M. (2001, June). *Performer-Instrument Interaction: Applications to Gestural Control of Sound Synthesis*. University Paris VI.

Wanderley, M. M., Schnell, N., & Rovan, J. (1998). Escher-modeling and performing composed instruments in real-time, *2*, 1080–1084.

Weinberg, G. (2008). The Beatbug–evolution of a musical controller. *Digital Creativity*.

Westeyn, T., Brashear, H., Atrash, A., & Starner, T. (2003). Georgia tech gesture toolkit: supporting experiments in gesture recognition. *Proceedings of the 5th International Conference on Multimodal Interfaces*, 85–92.

Wilkinson, S. (1997). Phantom of the Brain Opera. *Electronic Musician*.

Wilson, A. D., & Bobick, A. F. (2000). Realtime online adaptive gesture recognition. *Proceedings 15th International Conference on Pattern Recognition*, *1*, 270–275.

Winkler, T. (2002). Fusing movement, sound, and video in Falling Up, an interactive dance/theatre production. *Proceedings of the 2002 Conference on New Interfaces …*. Retrieved from http://www.brown.edu/Departments/Music/sites/winkler/research/papers/fusing_movement.pdf

Wu, J., & Chan, C. (1993). Isolated word recognition by neural network models with cross-correlation coefficients for speech dynamics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *15*(11), 1174–1185. doi:10.1109/34.244678

Young, D. (2002). The Hyperbow controller: real-time dynamics measurement of violin performance. *NIME '02: Proceedings of the 2002 Conference on New Interfaces for Musical Expression*.

Zhang, X., Chen, X., Wang, W.-H., Yang, J.-H., Lantz, V., & Wang, K.-Q. (2009). Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. *IUI '09: Proceedings of the 14th International Conference on Intelligent User Interfaces*. doi:10.1145/1502650.1502708

Zhao, L. (2001). *Synthesis and Acquisition of Laban Movement*. University of Pennsylvania.