A Multimodal System for Gesture Recognition in Interactive Music Performance

Dan Overholt,* John Thompson,
Lance Putnam,† Bo Bell,†
Jim Kleban,†† Bob Sturm,***
and JoAnn Kuchera-Morin†**

*Aalborg University,
Department of Media Technology
Niels Jernes Vej 14
DK-9220 Aalborg
dano@imi.aau.dk
**Georgia Southern University
Department of Music
PO Box 8052
Statesboro, GA 30469
jthompson@georgiasouthern.edu
†Media Arts and Technology
University of California
Santa Barbara, CA 93106
ljputnam@umail.ucsb.edu
bobell@mat.ucsb.edu
jkm@create.ucsb.edu
††Department of Electrical
and Computer Engineering
University of California
Santa Barbara, CA 93106
jim_kleban@umail.ucsb.edu
***Institut Jean Le Rond
d'Alembert (IJLRDA)
Équipe Lutheries, Acoustique,
Musique (LAM)
Université Pierre et Marie Curie (UPMC)
11 Rue de Lourmel
75015 Paris, France
bsturm@gmail.com

Music performance provides an elaborate research test bed of subtle and complex gestural interactions among members of a performance group. To develop a paradigm that allows performers to interact as naturally and subtly with automated digital systems as they do with other human performers, an interface design must allow performers to play their instruments untethered, using only natural cues and body language to control computer information.

This article presents a multimodal system for gesture recognition in untethered interactive flute performance. Using computer vision, audio analysis, and electric-field sensing, a performer's discrete cues are remotely identified, and continuous expressive gestures are captured in musical performance. Cues and gestures that are typical among performers are then used to allow the performer to naturally communicate with an interactive music system, much in the same way that they communicate with another performer. The system features custom-designed electronics and software that performs real-time spectral transformation of audio from the flute.

Our approach therefore makes use of non-contact sensors, specifically microphones, cameras, and electric-field sensors embedded in a music stand that we call the Multimodal Music Stand System, MMSS (Bell et al. 2007). The multimodal array of untethered sensors contained within the music

Computer Music Journal, 33:4, pp. 69–82, Winter 2009

stand provides data to an analysis system that identifies a set of predetermined gestures as discrete cues, while simultaneously capturing ancillary performance gestures as continuous control data for audio synthesis and transformation processes. This information is used to control various interactions, such as the entrance and exit of a virtual "digital performer," via the discrete cues. In this article, we describe our work in the context of an interactive musical work composed for flute by JoAnn Kuchera-Morin and performed by flautist Jill Felber.

Our primary goals are: (1) to enable the performer to cue the interactive music system using simple gestures that are natural to musicians; (2) to reinforce recognition of cueing gestures through the combination of multiple modalities; (3) to capture ancillary gestures of the performer and map them to real-time audio synthesis and transformation parameters; and (4) to accomplish this without altering the instrument and without requiring direct physical contact from the performer.

Background

This section summarizes the current state of the field in gestural control of interactive musical performance, focusing on multimodal detection and gesture recognition.

Instruments for Expressive Control

Overholt (2007) summarizes three basic groups of gestural controllers for music: (1) instrument-simulating and instrument-inspired controllers, (2) augmented instruments capturing either traditional or extended techniques, and (3) alternative interfaces, with the subcategories of "touch," "non-contact," "wearable," and "borrowed."

Instrument-simulating and *instrument-inspired* controllers are gestural interfaces that simulate the look and feel of traditional instruments, but do not include the original functionality of these instruments. For example, a guitar-controller that does not have strings but instead uses sensors along the fretboard would fall into the category of instrument-inspired controllers (because the technique used to play it is noticeably different from the instrument on which it was based). A keyboard-based synthesizer is also an example of this first classification, but it is an instrument-simulating interface, because its playing technique mirrors that of the piano. In the case of the flute, the Yamaha WX-7 or Akai EWI provide instrument-inspired options. (More specifically, certain fingering modes are instrument-simulating for a soprano saxophone, and other modes are inspired by the flute, yet the mouthpiece is quite different.)

Augmented instruments retain the full functionality of an original instrument by including mechanical workings of the traditional instrument; however, they have been modified to interact with a computer with the addition of sensors that are intended to capture either traditional or extended techniques. One example of an augmented instrument capturing traditional techniques is the Yamaha Disklavier. It includes all of the strings and mechanical workings of a traditional piano, and hence retains the functionality of the original instrument while gaining the ability to interact with a computer. There are several flutes that can be classified as augmented instruments capturing extended techniques, including IRCAM's MIDI-Flute (Pousset 1992), Ystad and Voinier's "virtually real flute" (2001), and Palacio-Quintin's Hyper-Flute (2003). Each of these instruments implements a set of extended techniques through extensive physical modifications to the original instrument (placing multiple sensors on the flute). In actuality, some of the sensors allow traditional techniques to be captured, while others are aimed at extended techniques. (For example, accelerometers measure tilt and other motions of the instrumentalist that are not normally used to consciously modify sound.) This approach of placing sensors on the instrument has the advantage of obtaining a wealth of accurate and detailed information about the performer's interaction with the instrument. Some disadvantages include the inability of many flute players to acquire these unique instruments, and the burdens associated with having the sensors on the instrument itself (weight, power requirements, wiring tangles, etc.). Other augmented instruments are based on different instrument families as well.

Tod Machover's Hypercello (Levenson 1994) is an example of an augmented string instrument.

Alternative interfaces take on forms not resembling traditional instruments. These types of controllers include instruments such as Sonami's "Lady's Glove" (2007), and Waisvisz's "The Hands" (1985). (These two are examples of the *wearable* subcategory.) When used in musical scenarios, the Wacom pen controller can also be classified as an alternative interface, but as a *borrowed* controller, because it was not originally intended to control sound or music. These interfaces also fall into the *touch* subcategory (because there is physical contact being made with them). There are, of course, many other alternative interfaces that have been invented in recent times.

An early precursor to these alternative interfaces is the Theremin, invented by Léon Theremin in 1919. As the world's first non-contact electronic instrument, the Theremin sensed the distance to a performer's hands using changes in the strength of the electric field caused by the human body. The MMSS adopts this method of sensing and expands it to be used in conjunction with audio analysis and computer-vision techniques. The music stand is a good candidate for an unobtrusive alternative controller owing to its common use in traditional music settings. While the MMSS can clearly be classified as an alternative interface, it can be said to fall into two different subcategories of *non-contact* and *borrowed*. Hewitt's Extended Mic-stand Interface Controller e-Mic (Hewitt and Stevenson 2003) is a related alternative interface, falling into the *touch* and *borrowed* subcategories. The e-Mic uses a microphone stand to provide interactive control to a vocalist. While similar to the MMSS in its use (borrowing) of a familiar musical stage item, the e-Mic requires the performer to manipulate controls attached to the microphone stand through physical contact, making it less useful for instrumentalists.

Two commercial augmented music stands are available for traditional musicians: the Muse and the e-Stand. These electronic music stands feature such attributes as graphical display of scores, the ability to annotate digitally and save the annotations, automatic page turns, built-in metronomes and tuners, and network capabilities. These stands serve a practical purpose, but are not designed as gestural interfaces.

Outside of the commercially available options, MICON (Borchers, Hadjakos, and Muhlhauser 2006) is a music stand for interactive conducting of pre-recorded orchestral audio and video streams. Amateur participants conduct using a baton that is tracked via a sensor built into the music stand. Through the tracking, an amateur is able to control the tempo of playback as well as the dynamic balance between synthesized instruments. MICON features graphical display of the musical score and automatic page-turning.

Aside from the interest in conducting gestures as opposed to expressive gestures of an instrumentalist, there are fundamental differences between the motivation of MICON and the MMSS. One is that MICON intends to allow people with no musical training to interact with a virtual orchestra through the music stand, whereas the MMSS is aimed at expert-level interactions with trained instrumentalists who wish to engage in interactive performances with a computer, without resorting to drastic modifications of their instrument or excessive prosthetic accoutrements.

Although the MMSS falls neatly into the classification of alternative interfaces, it could also be looked upon as a way to augment any existing acoustic instrument through extended playing techniques. It is a general-purpose interface that allows an instrumentalist playing a traditional instrument to control digital data (e.g., processing their own instrument's sound and/or playing back pre-composed sounds) by using intuitive (natural) gestures such as cueing, and potentially also a set of extended musical gestures to control the triggering and continuous control over synthesis and sound-processing parameters.

Gesture in Instrumental Music

There are three commonly accepted types of music performance gestures: *performative*, *communicative*, and *ancillary* (Cadoz and Wanderley 2000). The importance of tracking the first two types of gestures is evident: performative gestures produce sound, and communicative gestures (nods, eye contact, and similar cues) direct other performers. Ancillary gestures—intuitive body movements of the performer while playing—are expressive or emotive gestures that communicate musical meaning to the observer. Wanderley (2001) notes that ancillary gestures, although they don't alter the sound production of the instrument itself, do have an effect on the way the sound is heard by an audience. In addition, Cadoz and Wanderley (2000) have shown that certain ancillary gestures are repeatable and consistent within a particular piece. These types of musical performance gestures carry important details that can be used to inform an interactive music system. In the first performance with the MMSS, the system concentrated primarily on capturing communicative and ancillary gestures of the performer, thus enabling the performer to use these gestures to control and transform sounds as they would in a musical ensemble. (Performative gestures would have been redundant in the context of the particular composition, because the piece focused on processing the sound emanating from the performer's instrument.)

There have been many efforts to obtain ancillary gesture using computer-vision techniques. At the University of Genoa, Antonio Camurri and colleagues have developed a suite of software tools, EyesWeb, for gesture recognition. These tools are capable of identifying gestures on basic (syntactic) and advanced (semantic) levels (Camurri et al. 2005). Additionally, Camurri et al. (2004) explores using them to analyze dancers' movements, and thereby to control music parameters. Similar work has been attempted on the syntactic level by Qian et al. (2004), and on the semantic level by Modler and Myatt (2004).

Gesture recognition using audio features has long been a research topic in audio pattern recognition. Rowe (1993) gives an excellent overview of the subject. Tanaka (2000) discusses the use of sensors for interpreting gestures that are either essential or nonessential to music performance. To assure the most accurate gesture tracking possible, we combine these concepts in developing our MMSS.

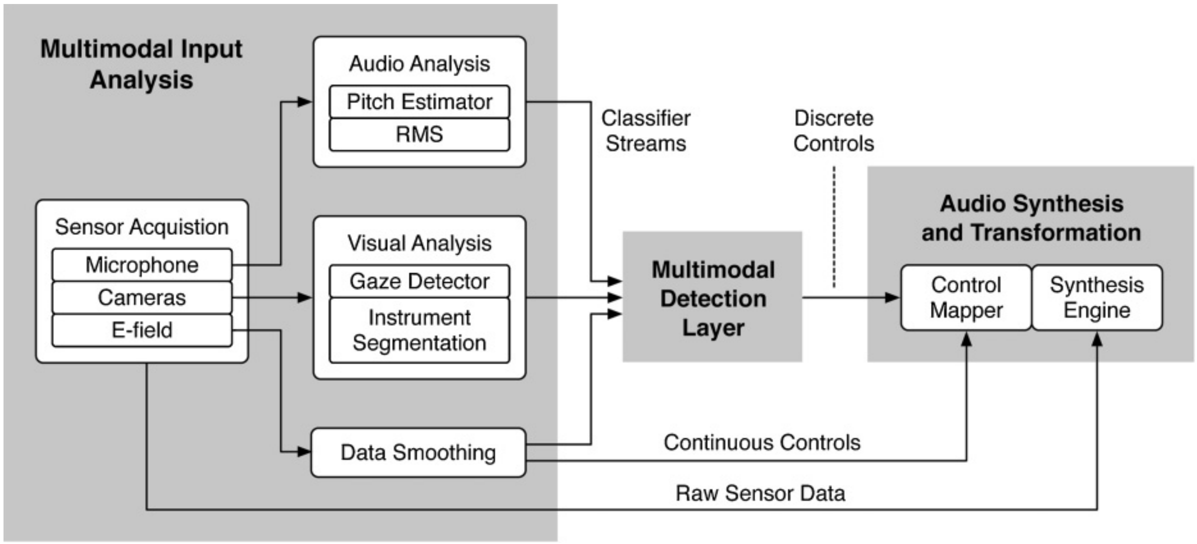
System Design

Our system design uses multimodality in a multimedia feature-extraction space as a means of phrase and motive synchronization. A key feature designed into the MMSS as compared to the systems mentioned herein is its use of multimodality. Certain of the aforementioned works primarily rely upon computer-vision and image-processing techniques, whereas the MMSS combines multiple cameras with audio analysis and electric-field sensors. Similar to the MacVisSTA system (Rose, Quek, and Shi 2004), we look for combinations of events in different modalities within concurrent time windows. Unlike that system, however, the MMSS operates dynamically in real time.

Many gestural nuances can be more accurately interpreted with the combination of audio, visual, and performer proximity data. The MMSS marks improvement over our previous work (Kuchera-Morin et al. 2005) in computer vision, allowing for natural gestures from a performer. Identifying those gestures using multiple forms of media ensures a more accurate recognition of a musical gesture. For instance, in the first test bed for the MMSS, a composition for flute and computer, one of the performance gestures requires the flautist to hold the flute in a downward angle while playing a low B-natural. This performance gesture was identified by both a visual cue and an aural cue in combination with an electric-field proximity to the lower-right antenna, thus double-checking flute angle and the flute pitch. This gesture allowed the flautist to control the entrance of a sound file, and it could also be mapped to control a processing algorithm, transforming the flute.

Using multiple forms of data in this multimedia feature-extraction space, a performance gesture is accurately identified. Within the MMSS, a camera, microphone, and electric-field sensors allow for three different types of data to be combined for gestural recognition. The resulting rich gesture can then simply trigger a sound file or a processing algorithm, synchronizing to a score at the phrase level, or it can perform multimodal control over a multidimensional parameter space. However, our

Figure 1. Multimodal interface system model.



current system still lacks precise, dynamic synchronization of the acoustic instrument and computer algorithm at smaller time scales. This problem will be addressed in future versions of the MMSS.

MMSS as an Intelligent Space

Interactive music frequently addresses the issues of interactivity by requiring performers to work with additional physical elements to which they are unaccustomed. These physical elements include foot pedals (discrete and continuous), worn sensors, and devices attached to the instrument. These allow performers to control computer-generated sounds, but they do not give performers the opportunity to interact with the machine in the same way they interact with other performers. Many performers find these elements to be distractions that interfere with their ability to perform (McNutt 2003).

To achieve more intuitive interactivity, an intelligent space is needed—a space that can sense many aspects of the performer's movements and infer their intentions, thereby interacting accordingly. A space such as this should be designed as a "person," with eyes, ears, and a sense of dynamic three-dimensional motions. Of related interest is the MEDIATE environment (Gumtau et al. 2005). MEDIATE is a responsive environment replete with sensors (microphones, interactive floor, cameras, interactive wall, and objects) that promotes multi-sensory interaction with the space. MEDIATE evaluates the measurements of its sensors and makes decisions about the novelty or repetitiveness of participant actions to tailor media feedback accordingly.

The MMSS with its camera, microphone, and electric-field sensing begins to embody the meta-concept of an intelligent, sensing, general performative space that can be expanded and further developed for precise dynamic, interactive control of any type of data. Whereas the current system uses a standard video camera, microphone, and custom electric-field sensing, future MMSS versions may incorporate three-dimensional "depth-of-field" cameras (e.g., the "Swiss Ranger" optical imaging system that provides real-time per-pixel distance data at video frame rates), or directional microphone arrays, for example.

System Overview

Figure 1 describes the multimodal interface system model. The MMSS consists of three parts:

the Multimodal Input Analysis segment, which includes electric-field sensing, visual analysis, and audio analysis, and gathers information about the musician's performative space; the Multimodal Detection Layer, which analyzes this input data, sending triggers and control messages according to user-defined conditions; and the Audio Synthesis and Transformation engine, which listens for triggers and continuous controls that affect how it creates and/or alters the musical accompaniment.

Analysis Techniques

This section describes the techniques used in each analysis method.

Visual Analysis

By observing flautists playing in duet, we were able to determine a lexicon of gestures in the visual domain. The following gestures were selected for computer vision analysis: flute dipping and raising, and head turning.

Visual analysis has the dual foci of segmenting and tracking the flute, and detecting and tracking the motion of a face in frontal pose. In the MMSS, two cameras are used to detect multiple gesture types. One camera is mounted on the top of the stand and is used to capture the movement of the flute. The other camera is placed to the side of the stand and detects a flautist's gaze.

The gestures of flute dipping and raising, as shown in Figure 2, are defined, respectively, as the angle of the flute being below or above a certain threshold. The first step in determining the flute angle is to isolate the flute in the scene. We began to tackle the problem of flute segmentation by noting that the flute itself is often the only reflective object in the scene. To isolate the most reflective object, we first perform standard background subtraction on the scene to isolate the flute and performer. We convert the resulting video from RGB to HSV color space, and then apply a threshold to filter out all but the brightest objects. Because the flute is not purely reflective and there are occlusions due to the performer's hands, the result of the thresholding operation consists of blobs on the line of the flute along with noise. We reduce these blobs to their centroids, and then we perform linear regression using the random-sample consensus algorithm. The angle of this resultant line is used as a feature to inform the gesture recognition. The processing is carried out using our custom Max/MSP/Jitter object, jit.fluteends, available at https://svn.mat.ucsb.edu/svn/mmms/code/max.jit.fluteends.

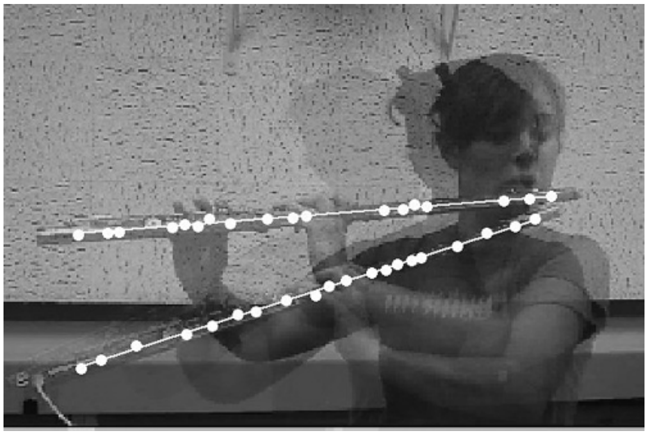


Figure 2. Flute segmentation as outlined by the algorithm.

The second gesture, head turning, involves motions for cueing that include gaze and eye contact. Detecting an eye-contact gesture using a retinal tracking system would prove unwieldy in a performance environment, so instead the gesture is detected using a face-detection algorithm from a laterally placed camera (to the side of the music stand). An OpenCV implementation of the Viola-Jones method (Viola and Jones 2001) detects the frontal pose of the performer's face as the performer turns slightly to make a cue.

Audio Analysis

A condenser microphone mounted on the stand sends audio from the flute into a computer running Max/MSP. Using a robust combination of temporal and spectral analysis, the system obtains pitch and root-mean-square (RMS) amplitude data, along with an estimation of note attacks. The audio analysis uses Tristan Jehan's analyzer~Max/MSP object (Jehan 2007). The audio-analysis features are sent via Open Sound Control (OSC) to the Multimodal

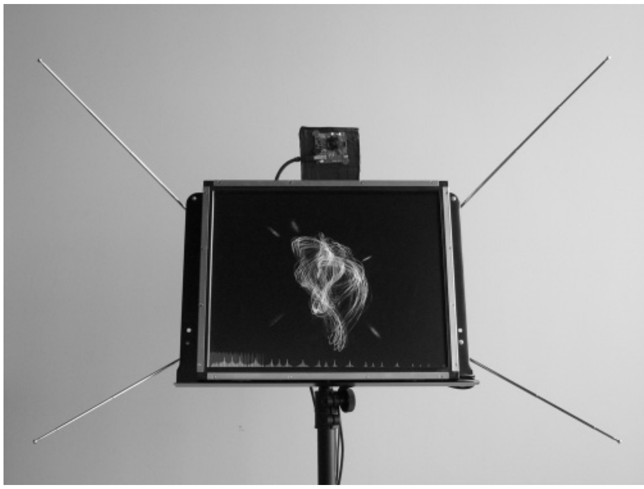


Figure 3. Prototype MMSS with electric-field-sensor antennas mounted at the corners of the stand.

Detection layer, where higher-level gestures are derived.

Electric-Field Sensing

The Multimodal Music Stand system incorporates four electric-field sensors (Mathews 1989; Boulanger and Mathews 1997; Paradiso and Gershenfeld 1997), as shown in Figure 3. These are used as part of the multimodal gesture-detection system and also as input sources for the control of continuously variable musical parameters, such as sound brightness or density. The electric-field sensors capture bodily and instrumental gestures (they are sensitive to both) made by the performer, which are tracked via the four sensor antennas.

The electric-field-sensing technique is based on the original Theremin circuit topology (Smirnov 2000), but all timing calculations are done entirely in the digital domain. Whereas Theremin's circuit designs utilized analog heterodyning techniques, the MMSS only uses the "front end" of this type of analog circuit. The remaining logic is accomplished through the measurement of high-frequency pulse widths using custom-designed firmware on the CREATE USB Interface (Overholt 2006).

The data from each of the independent electric-field sensors is received in Max/MSP/Jitter, as pictured in Figure 4, mean-filtered, and sent on to the multimodal detection layer via OSC. Using four channels of sensing makes it possible for the MMSS to provide full three-dimensional input. In contrast with the Theremin's dual-channel approach, three-dimensional proximity can be sensed. The overall intensity of all four antennas is used to determine the *z*-axis of the gestural input, as this mixture corresponds directly to the performer's overall proximity to the stand. The independent antenna's signal strengths correspond to each quadrant of the *x*–*y* space, so they are used to gather up/down and left/right gestures, and visualized using a simple display window in Max/MSP/Jitter.

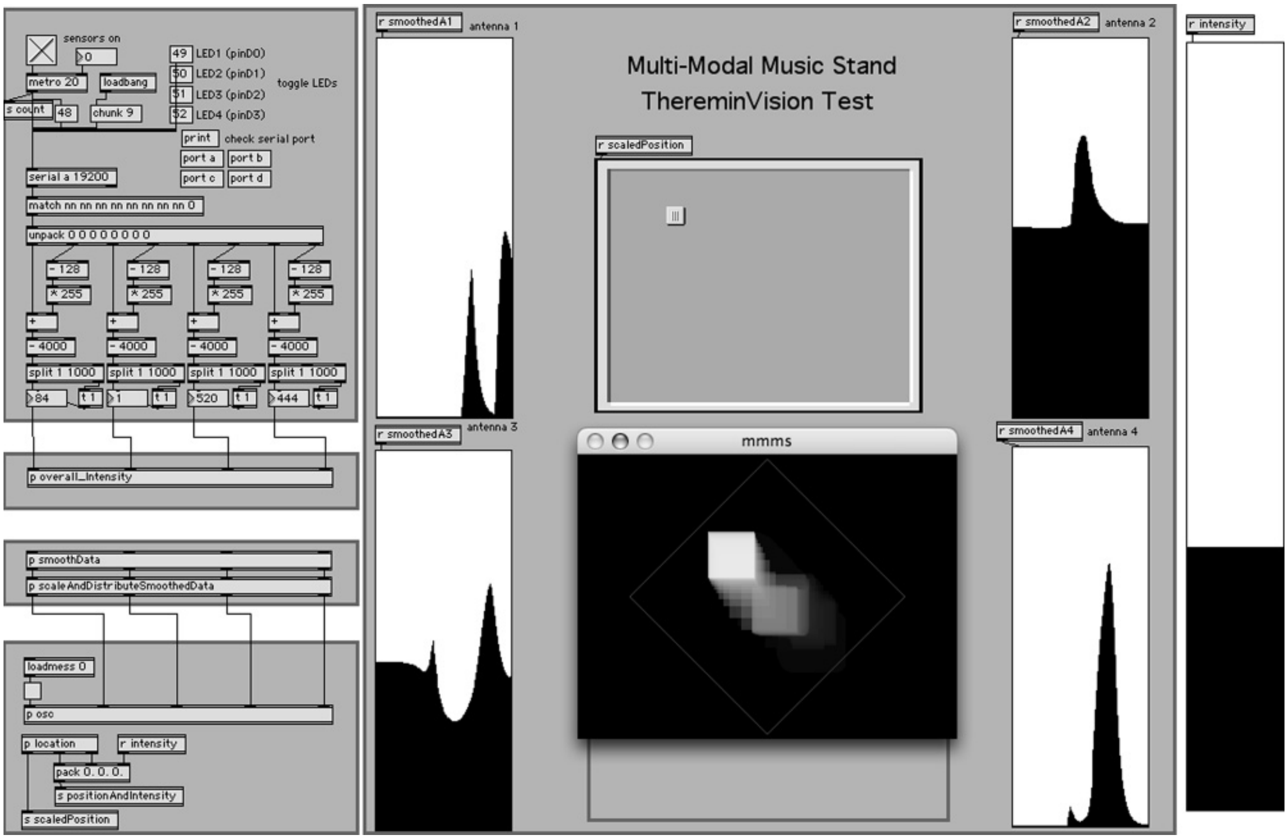
Multimodal Detection Layer

The Multimodal Detection layer integrates audio and visual classification results (pitch and audio amplitude estimates, face detections, angle estimates) as well as proximity of the performer to the electric-field sensors for gesture detection. A GUI allows composers to define the types of gestures occurring in the piece, either asynchronously (occurring at any time in the piece) or synchronously (ordered by timing). Gestures can be defined to occur in a single modality alone (e.g., the occurrence of one particular note), or more robust combinations by requiring that gestures occur in multiple modalities together within a short time period. For example, a gaze to the side-mounted camera, along with a certain loudness of playing and/or proximity to one antenna, can be required for a particular gesture. Upon detection of the pre-defined gesture, a trigger is sent to the synthesis machine in the form of an OSC message.

Audio Synthesis and Transformation

The sound synthesis component of the MMSS system is based on a client–server model (McCartney 2002). The synthesis server receives network commands that are mapped into control logic and/or direct modulation. The client–server model was deemed necessary to support distributed processing in cases where more complex analysis and synthesis algorithms are required.

Figure 4. The software in Max/MSP/Jitter, to receive, filter, and process electric-field sensor data, sending it via OSC.

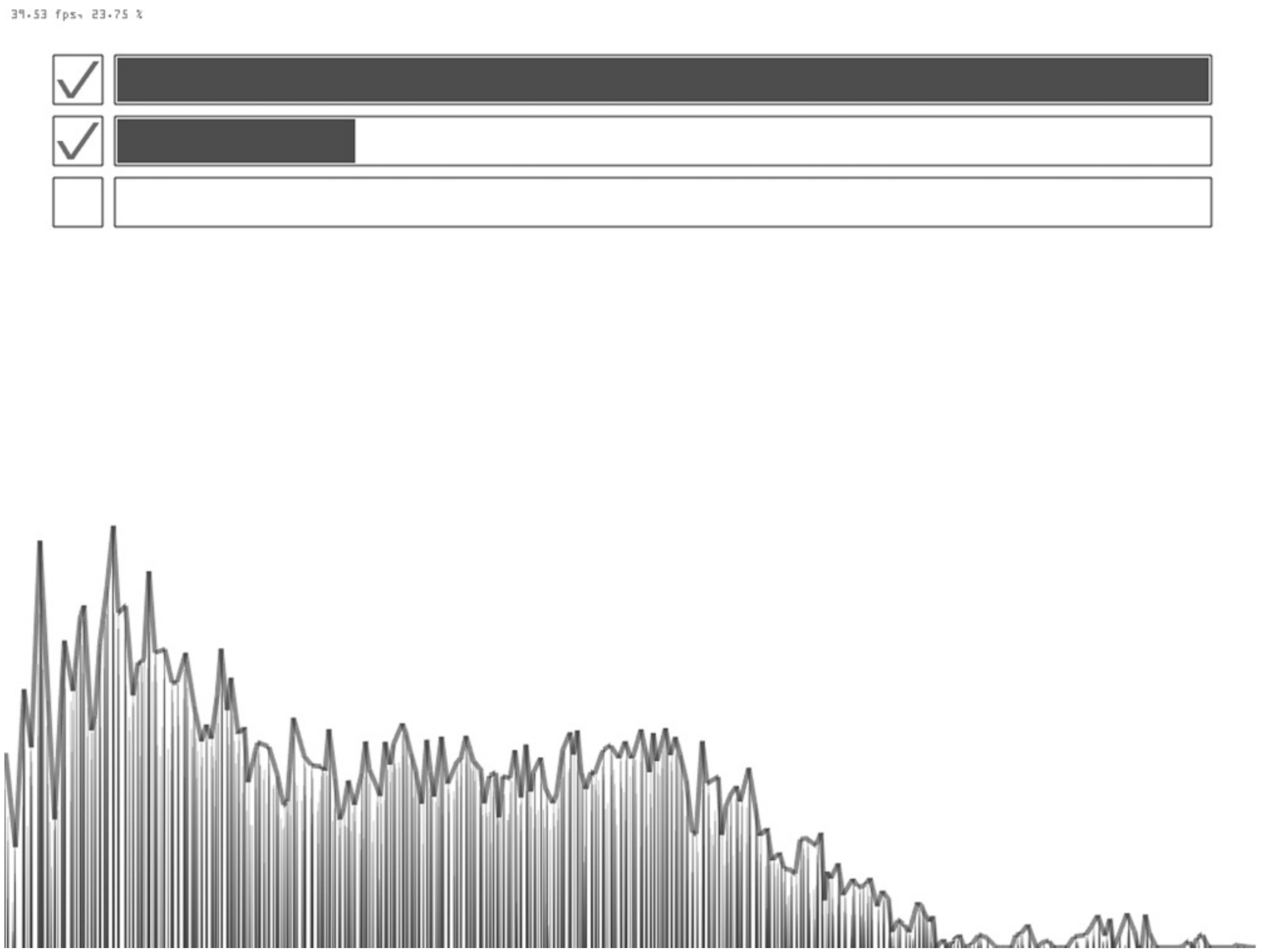


In theory, the synthesis server can receive and act upon any arbitrary command or signal; however, it was found that modes of interaction tended toward being either discrete or continuous. Discrete controls are used to trigger well-defined actions within the synthesizer, such as starting playback of a specific sound routine or randomizing a set of control parameters. Discrete commands are a way to encapsulate a group of actions that can be executed with minimal effort and ambiguity. They allow the performer to "set and forget" specific processes, giving the performer the ability to execute several distinct musical expressions in parallel. Continuous controls, on the other hand, offer a direct means for modulating parameters of the synthesis system. This connectedness between the human's intention and the computer's reflection enables nuance and spontaneous articulation that would be difficult to obtain by other means. In addition, continuous control allows the operator to more freely direct movement in time without the rigid constraints of predetermined actions and signals.

Continuous commands need to be filtered before they are used in synthesis algorithms, to avoid clicks and control-rate artifacts in the audio. As a general solution to this, we pass each control stream through a one-pole low-pass filter within the sample loop. Conceptually, we found it useful to think of all synthesis processes as contained within a separate temporal partition whose only means of access is through filtering ports. Low-pass filtering has the additional side benefit of allowing brief control gestures (quick movements) to be expanded over time, thus introducing the simultaneity advantages of discrete commands while maintaining the performer's expressive inflections.

Although no specific means for sound transformation is defined for the MMSS, we have developed

Figure 5. Graphical display of audio-synthesis and transformation-processing cues.



a custom synthesis library, Gamma (Putnam 2009a), and a programming interface using Graphics Library of Views (Putnam 2009b) for the creation of specialized synthesis systems. We required a cross-platform, easy-to-use synthesis library that had both computational efficiency and expressive structuring of signal flow. The synthesis library allows flexible structuring of unit generators by means of single-sample evaluation, and it contains a large collection of generic tools for performing sound analysis, transformation, and synthesis in both the time and frequency domains. The audio synthesis machine also graphically displays the activity of sound synthesis and transformation cues, as shown in Figure 5.

The cues are ordered sequentially in time from top to bottom, with a check box to indicate that the cue has been received and a status bar to display its temporal envelope. The cue visualizations, along with a spectrogram, inform performers about the state of the synthesis machine so they can gauge and/or adjust their performance timings for better synchronization.

MMSS and Its Use in the Composition *timeandagain*

The first proof of concept for the Multimodal Interface was the premiere by Jill Felber of JoAnn

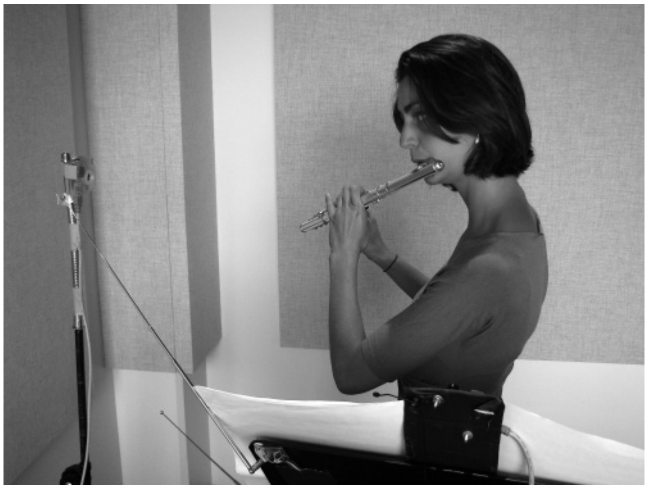


Figure 6. Original gesture #1: head turn and RMS level of audio input.

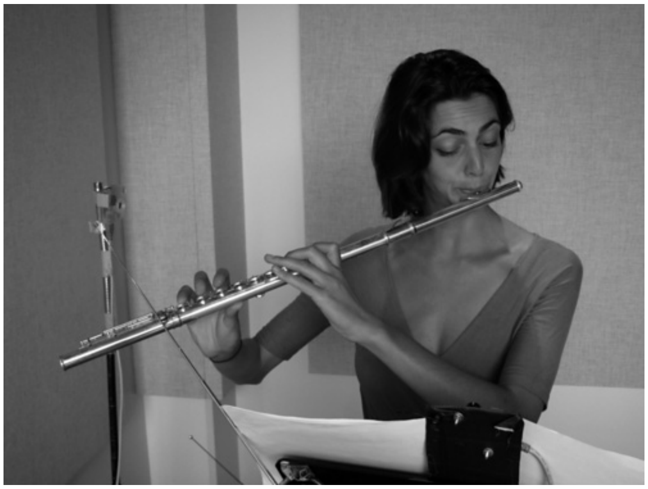


Figure 7

Figure 7. New gesture #1: electric-field proximity and RMS level of audio input.

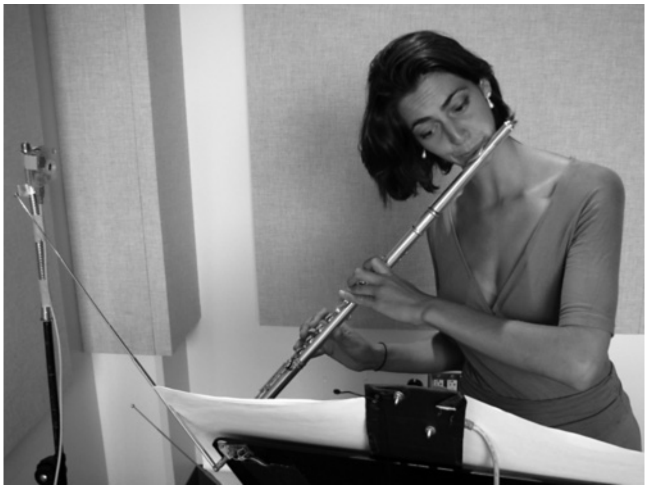


Figure 8

Figure 8. Flute angle down and low flute note.

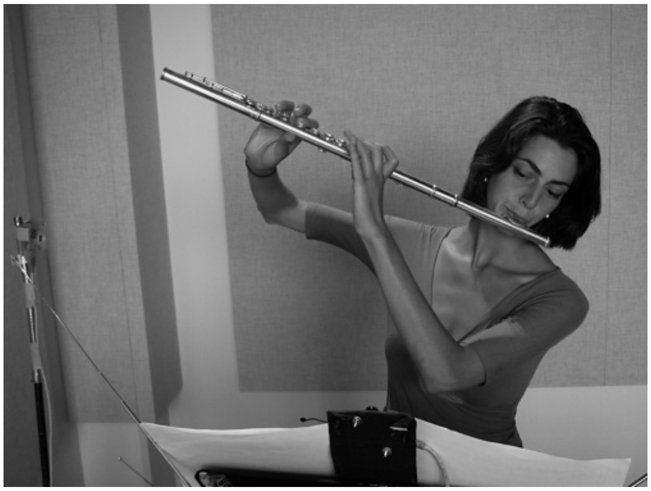
Kuchera-Morin's *timeandagain* for flute and computer, which took place on 22 March 2007 at the Rochester Memorial Art Gallery Auditorium during the 25th Anniversary of Computer Music Conference at the Eastman School of Music in Rochester, New York. The piece is constructed so that the computer-generated portion of the work is derived from the first 21 seconds of the live flute part. The processed sounds are stored in sound files that are controlled and triggered by a set of gestures that the performer uses during the performance of the piece; thus, the performer interacts with the computer much in the same way that she would cue other performers—with visual, audio, and movement cues.

A set of gestures was constructed using multimodal data to give an accurate detection of the gestural cue. Three different gestures were used in the piece, for a total of six trigger events, for cueing sound files. Figure 6 displays the first attempted cue in the piece. The cue was originally made up of the performer turning her head to the side camera for face detection, while playing the flute for audio RMS-level data.

This gesture proved unreliable owing to instability with lighting conditions for the gaze-detection algorithm. Gaze detection was hence dropped for this particular composition, and the problem actually facilitated system optimization, as only one computer and one camera was then needed for the visual gestures without the head-turn cue. The first gestural cue now did not contain visual data. It contained electric-field sensor data and RMS audio data, a gesture in which the flautist moved toward the electric-field sensor while playing, shown in Figure 7. This cue proved to be more reliable in performance.

The second gestural cue in the piece used visual data from flute segmentation by having the flautist angle the flute down for a visual cue while playing a low note for the audio data, as shown in Figure 8.

Figure 9. Flute angle up and high flute note.



The third cue used a visual angle up and high note for the audio, as shown in Figure 9. There were three more triggers of these cue types throughout the piece.

Dynamic control of spatialization was accomplished through the four electric-field sensors on the music stand. The sensors created a three-dimensional sensing space that allowed the performer to move the sound to a particular loudspeaker or group of loudspeakers by moving closer to one of the electric-field sensors on the music stand.

The stand proved effective in performance. The cues were accurate and intuitive for the performer, giving her reliable, untethered control and freedom of interaction with visual, audio, and movement data, much in the way a performer interacts with another performer.

Future Work

Some aspects of the MMSS are being further developed to improve the overall performance of the system. The next steps for making the MMSS a more robust and intuitive tool for gestural control in musical performance are to concentrate on dynamic control at the note level and within the note. This will allow the performer's expressivity to increase beyond that of any traditional instrumentalist who is cueing an interactive music system that lacks these supplementary features of the MMSS.

Our major goal of future research is to develop further methods for expressive gesture tracking. Many features (audio frequency and amplitude, instrument position and angle, player proximity and motion vectors, etc.) contribute to a large "feature space" with dynamic changes that can be tracked over time. Using unsupervised machine learning, we hope to better identify feature clusters in the space that can be tagged with user-defined metadata to link the clusters to expressive performances. In this way, we hope to make the system not only responsive, but also extremely expressive.

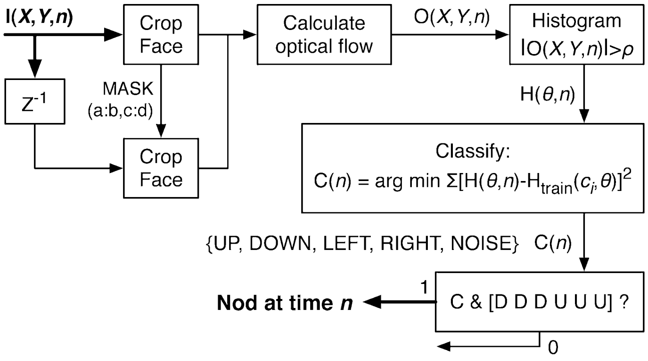
The Use of Score-Following for Synchronization

To achieve synchronization at smaller units of time than the phrase, we attempted score following within the system design. We attempted to work with a state-of-the-art score follower, Suivi, from IRCAM (Orio, Lemouton, and Schwarz 2003). This algorithm uses a Hidden Markov Model to represent the score elements (notes and rests), with transition probabilities found through training. One of the features Suivi uses is something similar to harmonicity, which is only mildly based on pitch. Our plan was to see how this algorithm worked on a musical piece based on extended techniques. It was not robust enough to be successful.

As computerized score-following continues to evolve, the technique still has limitations that impose unsatisfying constraints on composers and performers of contemporary music. The composer must stick to events that would be obvious to a computer, and this entails a pitch-focused writing. The performer is forced into what flautist Elizabeth McNutt (2003) calls a "prison of perfection." Although the ideal computerized score-follower will allow the computer to adjust to the performer, in practice, the performer usually must adjust to the score-follower. This loop then can completely alter the way of playing to suit an algorithm.

Score-following is a continuing research area, but at present, it is not reliable for performance of some contemporary music. As research efforts continue, robust score-following software may become available. In the meantime, we will attempt

Figure 10. Nod-detection flow diagram. If the optical flow moves downward over several frames, followed immediately by several upward classifications, then a nod is detected.



to move our multimodal research to the smallest level of control by capturing ancillary gestures of the performer and establishing a gestural vocabulary that can be controlled at the motivic level. By combining dynamically varying multimodal streams of data that precisely identify a gesture and allow continuous control, we can define new extended techniques for performers to master.

Ancillary Gestures: Head Nodding

To control ancillary gestures, we attempted detection of the performer's head nods as part of our vision-recognition system. Whereas others' work has used retinal tracking with cameras with custom infrared light-emitting-diode circuitry (Kapoor and Picard 2001), improved face-detection algorithms allow direct nod detection from the optical flow (tracking the overall direction of motion) in the region around the face. Figure 10 shows a system diagram of the nod detector.

When a face is detected in the same region when transitioning from one frame to the next, the optical flow in the transition between sub-windows is calculated using the Lucas-Kanade hierarchical method. The optical flow vectors within a certain magnitude are then arranged according to a histogram of their angles. These histograms are then classified using an L2 distance measure between possible classes of vertical, horizontal, and non-cohesive motions. A nod is defined as a series of downward-motion classifications followed directly by upward ones. Owing to the stringent design requirements for performance and the non-subtle motions required for the algorithm to reliably detect a head nod, this nod-detection system is not currently implemented in this version of the MMSS. However, we believe further refinement will lead to the detection of ancillary gestures and finer gestural control, and the implementation of head nods will be an important contribution to more fine-grained gestural control in the MMSS.

Conclusion

Control of digital information through natural gestures that humans typically make when working, communicating, performing, and playing an instrument facilitates intuitive human-computer interaction. Musical performance provides an elaborate research test bed of subtle and complex gestures that facilitate this research into new techniques. The Multimodal Music Stand System, an untethered interface for dynamic gestural control in music performance, allows for interactivity that neither hinders the performer nor requires extraneous action, and advances intuitive musical human-computer interaction. We continue to test the system with different instruments and in various performance situations, and we look forward to further research gains.

Acknowledgments

Thanks to Carla Sturm for assisting with early system prototyping, Jill Felber for helping test the system and for her performance of the premiere of *timeandagain* at the Eastman School of Music, and Rebecca Kleinmann and Matthew Wright for assisting with photo documentation. Support was provided by IGERT NSF Grant DGE-0221713.

References

Bell, B., et al. 2007. "The Multimodal Music Stand." *Proceedings of the 2007 International Conference on New Interfaces for Musical Expression.* New York: Association for Computing Machinery, pp. 62–65.

Borchers, J., A. Hadjakos, and M. Muhlhauser. 2006. "MICON: A Music Stand for Interactive Conducting." *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression.* New York: Association for Computing Machinery, pp. 254–259.

Boulanger, R., and M. Mathews. 1997. "The 1997 Mathews Radio-Baton and Improvisation Modes." *Proceedings of the 1997 International Computer Music Conference."* San Francisco, California: International Computer Music Association, pp. 395–398.

Cadoz, C., and M. Wanderley. 2000. "Gesture-Music." In M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music.* Paris: IRCAM, pp. 1–55.

Camurri, A., et al. 2004. "Multimodal Analysis of Expressive Gesture in Music and Dance Performances." In A. Camurri and G. Volpe, eds. *Gesture-Based Communication in Human-Computer Interaction*: *Fifth International Gesture Workshop.* Berlin: Springer-Verlag, pp. 20–39.

Camurri, A., et al. 2005. "Communicating Expressiveness and Affect in Multimodal Interactive Systems." *IEEE Multimedia Magazine* 12(1):43–53.

Gumtau, S., et al. 2005. "MEDIATE: A Responsive Environment Designed for Children with Autism." *Proceedings of Accessible Design in the Digital World Conference*. Dundee, Scotland: BCS, pp. 1–8.

Hewitt, D., and I. Stevenson. 2003. "E-Mic: Extended Mic-Stand Interface Controller." *Proceedings of the 2003 International Conference on New interfaces For Musical Expression.* New York: Association for Computing Machinery, pp. 122–128.

Jehan, T. 2007. "Tristan Jehan's Max/MSP Stuff. Available online at web.media.mit.edu/∼tristan/maxmsp.html. Accessed November 2007.

Kapoor, A., and R. W. Picard. 2001. "A Real-Time Head Nod and Shake Detector." *Proceedings from the 2001 Workshop on Perspective User Interfaces*. New York: Association for Computing Machinery, pp. 49–54.

Kuchera-Morin, J., et al. 2005. "Out of the Ether: A System for Interactive Control of Virtual Performers Using Video and Audio Streams." In *NSF Interactive Digital Multimedia IGERT Annual Research Review*. Santa Barbara: University of California, Santa Barbara, pp. 13–17.

Levenson, T. 1994. "Taming the Hypercello." *The Sciences* 34(4):15–23.

Mathews, M. 1989. "The Radio Drum as a Synthesizer Controller." *Proceedings of the 1989 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 42–45.

McCartney, J. 2002. "Rethinking the Computer Music Language: SuperCollider." *Computer Music Journal* 26(4):61–68.

McNutt, E. 2003. "Performing Electroacoustic Music: A Wider View of Interactivity." *Organised Sound* 8(3):297–304.

Modler, P., and T. Myatt. 2004. "A Video System for Recognizing Gestures by Artificial Neural Networks for Expressive Musical Control." In Antonio Camurri, Gualtiero Volpe, eds. *Lecture Notes on Computer Science.* New York: Springer-Verlag, pp. 541–548.

Orio, N., S. Lemouton, and D. Schwarz. 2003. "Score Following: State of the Art and New Developments." *Proceedings of the 2003 International Conference on New Interfaces For Musical Expression.* New York: Association for Computing Machinery, pp. 36–41.

Overholt, D. 2006. "Musical Interaction Design with the CREATE USB Interface: Teaching HCI with CUIs Instead of GUIs." *Proceedings of the 2006 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 425–430.

Overholt, D. 2007. "Musical Interface Technology: Multimodal Control of Multidimensional Parameter Spaces for Electroacoustic Music Performance." Ph.D. Dissertation, University of California, Santa Barbara, Department of Media Arts and Technology.

Palacio-Quintin, C. 2003. "The Hyper-Flute." *Proceedings of the 2003 International Conference on New Interfaces for Musical Expression.* New York: Association for Computing Machinery, pp. 206–207.

Paradiso, J., and N. Gershenfeld. 1997. "Musical Applications of Electric Field Sensing." *Computer Music Journal* 26(2):69–89.

Pousset, D. 1992. "La Flûte-MIDI, L'histoire et Quelques Applications." Ph.D. Dissertation, University of Paris-Sorbonne.

Putnam, L. 2009a. "Gamma – Generic Synthesis C++ Library." Available online at mat.ucsb.edu/gamma. Accessed January 2009.

Putnam, L. 2009b. "Graphics Library of Views." Available online at mat.ucsb.edu/glv. Accessed January 2009.

Qian, G., et al. 2004. "A Gesture-Driven Multimodal Interactive Dance System." *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 2004.* New York: Institute of Electrical and Electronics Engineers, pp. 1579–1582.

Rose, R. T., F. Quek, and Y. Shi. 2004. "Macvissta: A System for Multimodal Analysis." *Proceedings of the 6th International Conference on Multimodal*

Interfaces. New York: Association for Computing Machinery, pp. 259–264.

Rowe, R. 1993. *Interactive Music Systems: Machine Listening and Composing.* Cambridge, Massachusetts: MIT Press.

Smirnov, A. 2000. "Music and Gesture: Sensor Technologies in Interactive Music and the THEREMIN Based Space Control Systems." *Proceedings of the 2000 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 511–514.

Sonami, L. 2007. "Lady's Glove." Available online at www.sonami.net/lady_glove2.htm. Accessed January 2007.

Tanaka, A. 2000. "Musical Performance Practice on Sensor-Based Instruments." In M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music.* Paris: IRCAM, pp. 389–406.

Viola, P., and M. Jones. 2001. "Robust Real-Time Object Detection." *Technical Report February 2001/01*. Cambridge, Massachusetts: Compaq CRL.

Waisvisz, M. 1985. "The Hands: A Set of Remote MIDI-Controllers." *Proceedings of the 1985 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 86–89.

Wanderley, M. 2001. "Interaction Musicien-Instrument: Application au Contrôle Gestuel de la Synthèse Sonore." Ph.D. Thesis, University of Paris 6.

Ystad, S., and Voiner, T. 2001. "A Virtually Real Flute." *Computer Music Journal* 25(2):13–24.