A Manifold Interface for Kinesthetic Notation in High-Dimensional Systems

Insook Choi

Human-Computer Intelligent Interaction Laboratory
Beckman Institute
University of Illinois at Urbana-Champaign
ichoi@ncsa.uiuc.edu

1. Introduction

When children first encounter a piano, they begin exploring the object in untamed ways. They try different actions on various parts of the instrument with their own orientations unlike the ones prescribed in pedagogy such as Bayer or Hanon. There are differences according to the age groups. Children above age three, those who are in complete command of their own body orientations tend to show more aggressive behaviors. On the other hand, observe the toddlers, and notice how they begin discovering the interaction. First they discover the keyboard. They stand in one place and keep pushing two or three keys with a group of fingers. Their wrists rest along the body of the piano underneath the keyboard since they can barely place their noses above it. Next time and many more times they do the same. They smile when everything is there to reassure their previous experience. Then they discover the registers. They begin to move along the keyboard while translating the finger actions orthogonal to the straight line along the low and high register. This must be quite unnatural for their body orientation as hinted in their leanings for weight balance. One of the earliest encounters is to learn to compromise their body orientation with respect to the immutable physicality of the instrument.

This observation is seeded at the heart of the manifold interface project. When human beings explore external systems, the systems respond or react to the act of the explorers, and learning takes place for understanding the following:

- mutual systems of reference: how does the coordination work between two systems?
- divisions of labor: how much can performers influence the machine processes and how much of the processes are automated in the machine?
- requisite variety: is the interactivity between human and machine set up to be compatible?
- identification of external resources: what is the scope of additional performance gestures to be brought into the interaction to compensate the lack of variety when needed?

The first three are in the mode of closed loop introspection, the last is to look out for open loop references.

In music performance the learning is enforced by way of producing sounds. Thus the evaluation of the sound is not independent from the means to produce it. In computer music performance practice the foundation of aesthetics not only constitutes the sound as result also the technological construct that composers set up to condition the result. This amounts to introducing the development of auditory intelligence.

The manifold interface assumes the auditory intelligence of performers as a finely integrated intelligence. Auditory intelligence accounts for

- an ecological orientation towards the acoustic space when monitoring the resulting sounds,
- a sensibility towards the instruments, in other words, the sensory intelligence towards the systems in interaction, and

Reprint from : *Trends in Gestural Control of Music*, M.M. Wanderley and M. Battier, eds.

• an ability to adjust performance strategy with respect to the resulting sound.

Particularly, the performance strategy involves an adaptive kinesthetic energy control with respect to the physics of instruments and the resulting sounds. With computer music technology the physics of various instruments are often embedded in simulations. This very aspect asks for an abstract model of an interface that can provide access to the variety of simulations. Such an interface should provide ways of reconfiguring parameters, resolutions of numerical values, means to configure alternative coupling strategies among interactive parameters, and optional visualization of what is often referred to as a parameter space. The last, as much as it is optional, the author finds quite necessary since we lack in sound synthesis the physical aspect of action space which musical instruments naturally provide.

Overview of development of the manifold interface paradigm

The manifold interface enables multidimensional gesture-based interaction as alternative to two classic methods of control parameter variation. One method is known as *p-fields,* the practice of using arrays of parameter values to describe the state of a tone generator in the form of a procedural function call. The other is known as a *patch*, the practice of creating a control flow representation of a synthesis function which distributes control at selected time scales to nodes in a directed graph. Both of these computational representations of the state of a tone generator provide a machine-centric account of time and of the simulated space in the synthesis model. We wish to accompany these views with a human-centric representation that quantitatively couples the continuous time and space of human movement with the simulated continuity of the signal generator.

Problems of interaction with p-field and patch representations are notable when we consider that neither has developed into a practical notation system (MAX graphic patches notwithstanding), particularly for real-time performance. Most computer music works in the 20th century are non-repeatable realizations, thus avoiding problems of gestural interaction and notation for performance. Certain real-time performance works depend upon music notation for performance with traditional music instruments, to which the computer responds by receiving audio signals or MIDI-type control signals, or both. We seek an alternative that generalizes gesture-based tone generation without dependence upon an existing instrumental paradigm. Non-real-time computer music production arguably could also benefit from a notation system that provides standard indications for reproduction.

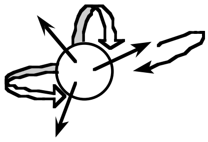
The desirable nonlinear attributes of contemporary synthesis algorithms such as physically-based models, further distance performers from an intuitive grasp of p-fields and patches. Nonlinear transient characteristics make it difficult to predict the state a tone generator will arrive in during a transition from one control value to the next. With most physically-based models, linear interpolations of control values will not produce reliable linear transformations in the audible signal. The linearity of the response depends upon the initial conditions, the order in which parameter values are modified, and the duration in which changes are applied. In nonlinear systems the audible characteristics of the tones are indicated by the transition characteristics between parameter values, rather than by the parameter values alone. In 1992 and 93, research in the interactive control of a chaotic circuit for tone generation led to the concept that transition regions in the parameter control space could be represented as surfaces (Choi 1994). Graphical surfaces were then used to represent transitions applied to multiple control parameters. Since the circuit control parameters numbered greater than three it became necessary to explore methods for projecting a "shadow" of n-parameter ranges onto the area defined by the surface. A cursor located on the surface indicated the state of multiple parameters. Changing cursor position created a control parameter transition. Edges of the surface indicated boundaries of valid parameter ranges. This was helpful considering many parameter combinations did not produce oscillations, and these parameter regions could be avoided by excluding them from the surface. In addition it was often the case that a particular order of parameter changes was necessary to prevent the circuit from entering basins of attraction. A basin is a region producing a steady-state periodic oscillation that tends to remain unaffected by further parameter variation. Basins often do not produce desired tones. To avoid entraining the circuit in these regions, specific paths through parameter space are required. Thus the quality of the tone is determined by sequential and duration aspect of parameter control transitions, not merely by arrays of parameter values. Neither p-field nor patch paradigms can effectively represent sequences of parameter transitions. Bundles of envelopes are possible but inefficient for coordinating multiple parameter variation.

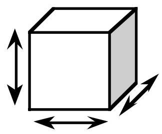
A surface representation creates an environment where paths may be drawn, traversing the surface. Paths provide an intuitive indication of transitions applied at different rates to multiple parameters. The development and formalization of these techniques are the subject of this paper. Section 2 discusses multiple definitions of spatial orientation in human perspectives and in machine specifications. Quantitative methods are described for converting control signals from one spatial embedding to another, and for calibrating the spaces to create a consistent relation from human movement to sound synthesis and back again. Section 3 presents data structures and management techniques for time domain operations. Two timescales are maintained in meaningful relation: the human-movement scale of physical gestures and musical events, and the signal processing scale of sound sample generation and synchronization between audio and graphical display and sensor input in a parallel processing architecture. Section 4 presents production examples of the manifold interface applied to the development of music compositions and performances.

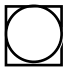
2. Space in Manifold Interface

The manifold interface project works towards refining a kind of interface with a particular projection technique. The technique involves computational geometric solutions for reducing the parameter space from *n* dimensions to 2 or 3 dimensions in representation of the space. The objective is to achieve an interactivity, not just an efficient controllability for *n* parameters. The difference between interactivity and control distinguishes a gesture performance from a mechanical functionality (Choi 1997). We refer to a *manifold interface* when emphasizing the interactivity, and a *manifold controller* when emphasizing the control signal flows. Since the interface visualizes a parameter space, the interface is often brought into a performance setting. This is analogous to the music performances where the ecology of various instruments has to be a part of the performance. Consequently the concepts of the spaces and various calibration techniques have been developed for the accountability of pre-compositional processes, notation, performance, and sensible representations for ecological orientation in Gibsonian terms of ecological psychology (Gibson 1966, 1979).

Human-centered coordinates and machine-centered coordinates

Human orientation to space and movement is organized like this: 

Machine orientation to space and movement is configured like this: 

The task of human-machine performance calibration is to reconcile these two different coordinate systems. Integration requires more than superposition of systems ( is not an integrated space). The geometric conversion between Cartesian and polar coordinates achieves a calibration of rotational and linear transformations in a space, but does not account for the perceptual orientation of human performers executing movements. This is a general problem statement for the incompatibility between human-centered coordinates and machine-centered coordinates.

The manifold interface provides a framework for normalization of human-machine performance space. A numerical technique is applied to calibrate the kinesthetic disposition of an observer to the dynamics of a sound synthesis engine. Calibration is achieved through a computational model of geometric space projection from two or three dimensions to *n*-dimensions. Normalization of a high-dimensional manifold is a general technique for expanding applications of control devices and motion tracking systems to gesture-based performance of parameterized sound synthesis. The space is subjected to a variety of temperaments. We describe three spaces for supporting a coherent presence of an observer with respect to a computing environment.

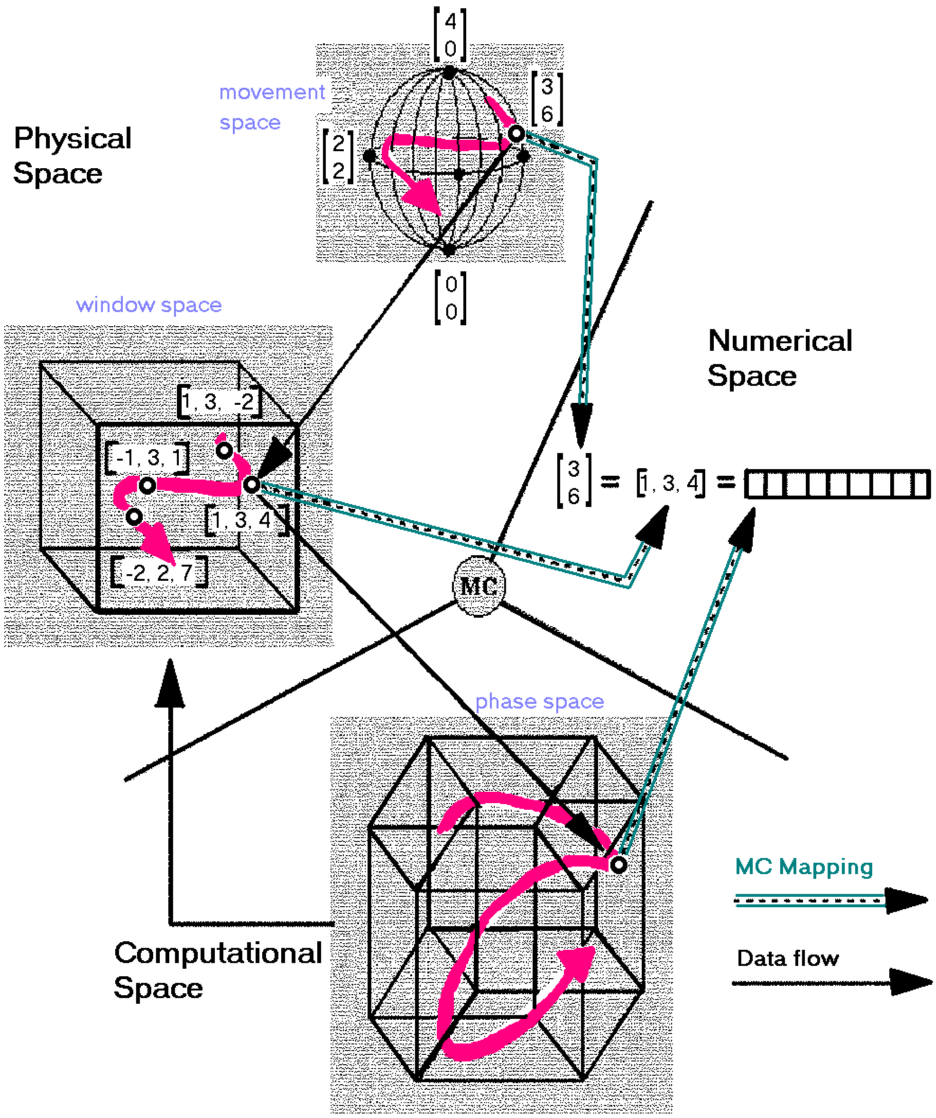


Fig. 1. Performance space calibration. A Manifold Control method is applied for mapping physical space to high-dimensional control space, with ecological organization of movement. The white arrow represents a Path event created by a hand gesture (see section 3). The same path is depicted in each of the spaces.

Figure 1 above shows how these three spaces may be defined in terms of computation or in terms of human movement. This distinction is important in virtual environments, where space acts as an immersive interface to computational processes, and control, display, and simulated dynamics appear to be occurring in the same virtual space. The differentiation and configuration of spaces in human-machine performance is a crucial part in achieving a coherent rehearsal competence for human performers.

Physical Space, Numerical space, and computational space.

Physical space is a pre-quantized space that is continuous within boundaries. The boundaries are determined by the space affordance inherent to the display apparatus. For example, the CAVE™ is a physical manifestation of a linear geometric model of space. Its physical space is bound to a 3.25m x 3.25m x 3m rectangle made of projection screens, within which a performer's actions take place. Spatial coordinates are represented from the point [0, 0, 0] (in x, y, z) at the center of the 10' cube, measuring the floor at [0, -5, 0] and the left wall at [-5, 0, 0] as observers face into the closed cube (the front screen). Appendix one provides further information about the CAVE environment and the software server for real-time sound synthesis applied in this research.

The use of a magnetic, video or mechanical tracking device to measure a person's position in 3-space provides an initial link to establish an embedding. The physical or electromechanical limits of these devices impose spatial constraints. For example the active field of present magnetic tracking systems become unstable at about 4 feet in any direction from the center of the system. Cables attached to magnetic head-tracking and hand-tracking systems encumber freedom of movement when a performer changes orientation or floor position. Cables are a notable and unavoidable intrusion in physical space.

In addition to imposing physical constraints, tracking devices only measure points in 3-space and not the organization of space implicit in an observer's movements. We tend to think of limbs and joints as motion constraints. An alternative is to describe their constraints as a manifold in a high-dimensional vector space. In other words, limbs and joints imply a structured movement space. Physical space can be viewed as a projection surface where movement space and phase space can be coupled.

Numerical space is a quantized physical space. The space translates position information to computable descriptions. The space displays extended three-dimensional views in a perspective projection in which the perception of the space may exceed the actual dimensions of physical space. In the CAVE while position and control information are bound to screen surfaces, the perception of this space is scaled according to the desired degree of geometric detail with respect to the simulated dimensions of the scene according to the viewing distance and the angle of view. For example one can view the entire circumference of the Earth in the CAVE. But to obtain a high degree of detail only a portion will be visible, as a closer viewing angle is required.

The quantization of numerical space is contingent to the digitization through measuring devices that move through the space while transmitting position coordinates in feet, and polar angles in radians. Quantization is time-dependent on the sampling rate of the measuring device (48 Hz for magnetic tracking) and numerical resolution is dependent upon the rate of motion through the physical space. Sensors also have limits of resolution that are less fine for ultrasonic devices than for magnetic devices, and finest for mechanical devices.

Numerical space may be re-oriented in physical space according to the position changes of an observer. When a tracking device measures an observer's head position and angle of orientation, perspective projection of three-dimensional geometric objects may be continuously relocated against the fixed physical screens, resulting in off-axis projection. At the same time the geometry appears stable to the observer compared to his or her body movements in physical space.

Computational space is where dynamical models are operated and their states are updated. The models are parameterized, their parameterization depending on the implementation choice. From the complete parameter space a selection of interactive parameters has to be made. Parameters can be visualized with an equivalent number of axes. Since we are bound to three-dimensional perception in space even with a VR display system it is not possible to visualize parameter values in spatial coordinates when the number of orthogonal axes exceeds three. In the manifold interface, visualization is provided by mapping points in three-dimensional space to control parameter vector arrays of size n. For this mapping we apply a geometric method to create a continuous, differentiable manifold passing through each 3D point and each value in the corresponding vector array. When a high-dimensional space is linked by a manifold to a 3D representation, we can traverse this manifold from physical space by placing a cursor at positions in the 3D representation. The 3D positions index the corresponding points in the manifold and transmit data from the manifold to control sound.

Action space and effective space in sound synthesis models

The performance with physically based models of musical instruments presents a unique philosophical problem. The idea behind such models is that we model the time domain physics of the instruments, not necessarily the physical properties of the instruments (Cook 1995). The wave behaviors from physical properties are approximated in effects with techniques such as waveguides (Smith 1992), sum calculations at nodes, and filtering to mention a few. Parameters of these models are often organized in similar classification with the real instrument performance parameters. There are two main advantages in these systems: 1) parametric exploration enables systematic documentation with accurate data storage and retrieval capacity, and 2) constraints are not bound to physical performance constraints such as human breathing capacity or the length of a bow. The unusual combinations of parameters often reveal unexplored acoustic potentials of the instruments, which can be only achieved by extremely extended performance techniques with real instruments. While these advantages open up unexplored possibilities and freedom they also present two problems: 1) how are we going to explore the vast parameter regions, and 2) how are we going to construct a performance practice paradigm? First we address the latter in terms of how performers play these instruments and how we are going to model the performance interactivity.

The problem is less obvious in other synthesis models that are not based on musical instruments. As there is no analogy to known performance paradigms the parameterization and access interface can be just about anything as long as the implementation is loyal to the models themselves. The anatomy of musical instruments is described as having two parts, an excitatory system and a vibrating body. From performers' point of view the parts correspond to action and effective space. *Action space* consists of the components such as keys, holes, and strings, *effective space* consists of the components such as vibrating surfaces and columns. Through the action space performers apply energy into the system and modify the geometry of the effective space. The columns are made longer or shorter by moving hands to cover and uncover holes or clamp and unclamp lengths of strings. With computational models of the instruments and other synthesis models, does the action and effective space paradigm still hold for interactive computer music performance gestures?

In one extreme one could easily indulge into absurd performance scenarios. One could indeed do lots of funny things with gadgets and tracking devices. It has been observed some composers tell performers to execute movements such as walking around on a stage or doing something with wired devices. These often result in invoking perceptive audiences to say, "who cares what you do there," or "I don't get it." The compositional problem for space is to construct a space proposition for an audience in a way that the space is conceivable. The main task is to establish a consistent coupling of space coordinates. This is the subject for the rest of this section. Human beings are not accustomed to perceiving inconsistent space. To establish a set of necessary invariants of spatial presentation is an accomplishment with computing technology. The compositional problem for movements is to construct an alternative proposition to traditional *tonmeister kinesthetic* (Choi 1998, 1999). A tonmeister kinesthetic in instrumental performance is that experience of coordinating sounds with performance movements observed with eyes. An audience would respond in negative way when the principle of tonmeister kinesthetic is absent or even violated. However, in computational sound production the spatial relations of a musical instrument are embedded in a sound synthesis algorithm. For interactive performance, the parameter space of the instrument simulation must be brought into a normalized relationship with a performer's movement orientation. This very aspect presents an explicit binding of space and time of which the implication of the binding has been present all along in performance tradition, yet becomes an explicit problem to solve with computer technology. There is no inherent common metric to host a wired musical instrument with respect to its spatial layout from a human orientation, therefor performance movements are not immediately transferable to a spatial framework modeled by linear geometry. We can affix a mouse or joystick to the moving parts of the instrument or directly to a musician in order to convert positions in physical space to coordinates in a computer model. We currently perform such measurements with magnetic tracking devices that detect 3D position and orientation of real-world objects. Nonetheless, movements imported to the computer and defined in Cartesian coordinates do not establish a computational definition of performance space with respect to the implicit space of a sound synthesis model.

Movement space, window space, and phase space

In the previous section we described three spaces to engineer by coupling their coordinates. An additional distinction is needed to describe the system access in terms of human motion: from, through, and to where human motion signals are passed. In other words, in organization and representation of control parameter space we distinguish three spaces: movement space, window space, and phase space.

Figure 1 shows this distinction roughly corresponds to physical space, numerical space, and computational space accordingly.

Movement Space is that of the performer's kinesthetic orientation and actions.

Phase Space is the high-dimensional set of parameter ranges we desire to affect.

Window Space is a connection between the other two spaces.

The term *control space* implicitly refers to both phase and window space as a couple as we access the phase space through the window space.

Equation of Sample Computation:

$$s(t) = A\cos(\omega_c t + \beta \sin(\omega_m t))$$

Explanation of Terms:

$s(t)$ output signal (fn. of time)

A output amplitude

ω_c carrier frequency, in radians/sec

β modulation index

ω_m modulator frequency, in radians/sec

$\omega_c t$, $\omega_m t$ instantaneous car, mod phase, in radians

Independent variables in the phase space include A in the range [0,1], ω_c in Hz, β in some abtract range usually [0,~20], and ω_m specified as a ratio to ω_c . These can be seen in Figure 2b.

Fig. 2a. The mathematical equation for Frequency Modulation, and definition of terms

The *phase space* of a system is the traditional *n*-dimensional space where vector arrays -- n-tuples of real numbers -- amount to influence the *n* states of the simulated model. For many simulated models, multiple parameters and all combinations of their values present a massive space to explore. Figures 2a, 2b and 2c present three common representations of Frequency Modulation, a well-known synthesis phase space. It can be seen that none of these representations provide intuitive orientation for gesture-based interactions. Figure 2a presents the mathematical definition of the phase space and specifies the independent terms that can be brought under gestural interaction.

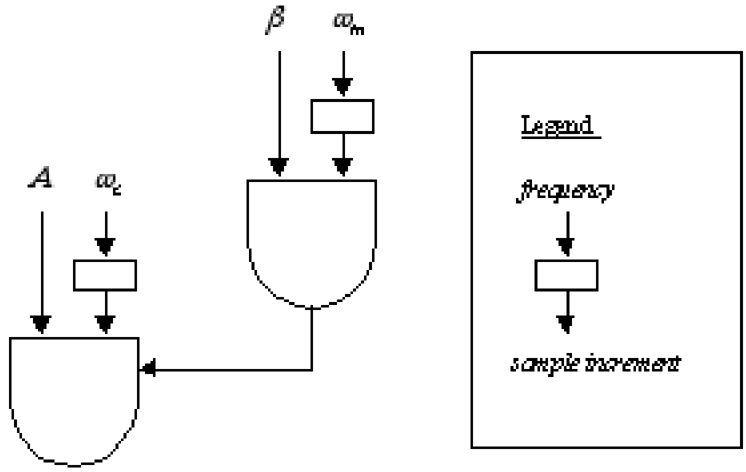


Fig. 2b. A visual programming patch-diagram representation of Frequency Modulation, with phase space represented as independent inputs. No concept of range nor intervals of relative transformations are provided.

Figure 2b shows a standard control flow diagram of the sound synthesis model. While the visual metaphor of the patch panel is helpful to understand the causality of the phase space, this representation does not provide a reference for a movement-based orientation to that space.

Figure 2c shows a common potentiometer-based version of the classical "p-field" representation of the control parameters. The linear layout of parameters is intelligible up to the limit that can be displayed in one screen space. To display and compare many different parameter settings using the representation in Figure 2c would be inefficient and visually burdensome. Regardless of the visual overload all of the parameters have an influence on the sound expression, and it is desirable that the entire system come under gesture-based interaction. The vertical potentiometers ("sliders") shown in Figure 2c do not support multiple-parameter variation in real-time (Hunt and Kirk 1999). Meta-sliders can bundle multiple parameters and support interaction within a fixed ratio relationship. In that case, the substantial part of parameter regions would be occluded due to fixed-ratio parameter coupling. In many exploratory cases meta-sliders constrain access to changing ratios between parameters and prohibit finding the optimal relationships between parameters. We seek for efficient system access by organizing control parameters so that one can easily manipulate them into alternative combinatorial couplings with rapid annotation capabilities to keep track of sequences of gestural inputs. Also we want the representation of the systems to have visual simplicity while maintaining an accuracy of its relationship to the actual states of the systems. This visual simplicity is an important factor to engage observers in an intuitive exploration.

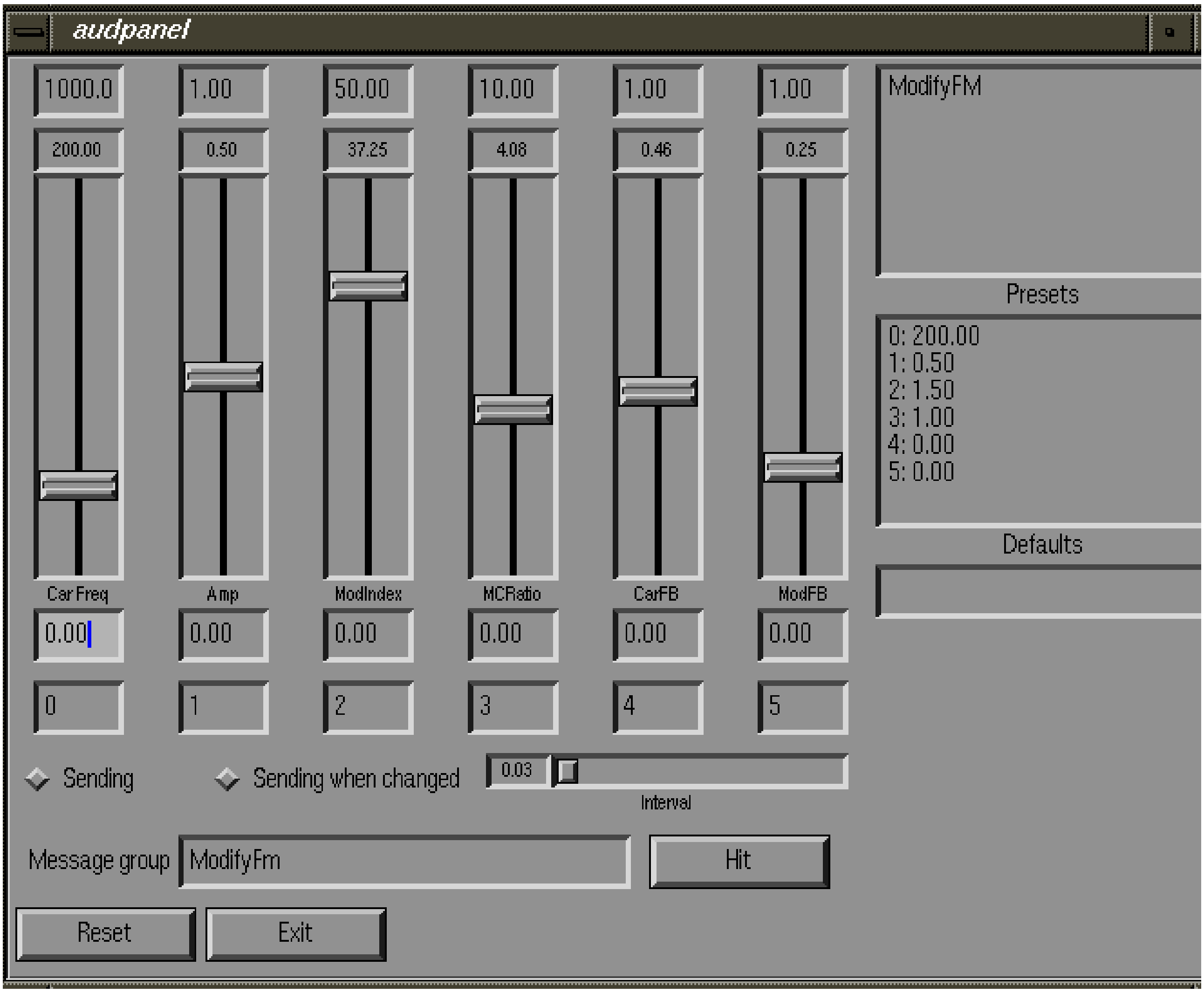


Fig. 2c. A potentiometer-based visual representation of parameters ranges for the Frequency Modulation phase space.

The phase space is all the permissible combinations of parameter values of an algorithm where trajectories of input gestures are encoded. A literal presentation of high-dimensional phase space will be visually undifferentiable resulting in loss of orientation. Thus we need a representation of the space with data reduction from arbitrary high-dimensional phase space to 3D space in perceptible form. We call this represented phase space a *window space*. The window space defines how a three-dimensional visual representation is embedded in the high-dimensional phase space as shown in Figure 3.

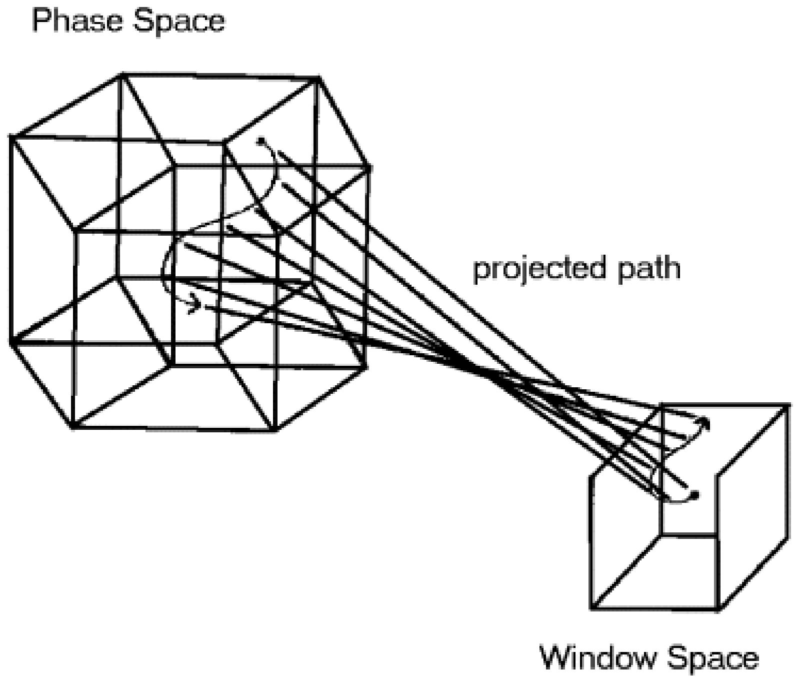


Fig. 3. An ordered set of contiguous points from a high-dimensional phase space, projected as a path in a three-dimensional window space.

We conceive of a three-dimensional visual display as a window onto the manifold so that an observer inputs changes to the system through the window space. An observer may effectively use the window space by panning and zooming in phase space in ways analogous to the use of lenses found in various observatory tools.

Temperament: Space calibration method

The coupling of window and phase spaces is achieved by defining a set of generating points. These points represent combinations of parameter values as user-specified, and they are associated with particular sounds.

The window space provides a domain for positioning generating points with respect to their sounds and to the human movement capacity. The association of the sounds in conjunction with positional orientation in window space enhances the ability to identify boundaries where character shifts occur in the states of the system.

Each generating point is determined in a phase space then brought into a window space. This requires a projection technique with data reduction and a smoothing function. We want to be able to visit these points and move smoothly between them in window space. However phase space may involve twists and bends during the embedding process, therefor the process should guarantee the representation to be continuous and "simple" while preserving a maximum amount of information.

To perform a continuous interpretation from window space to phase space, the image of the generating points in the window space is extended to a 3-dimensional lattice where lines through the generating points are more or less parallel to the principal axes of the space. All points in the lattice are then used to produce a corresponding lattice of similar geometry in the phase space.

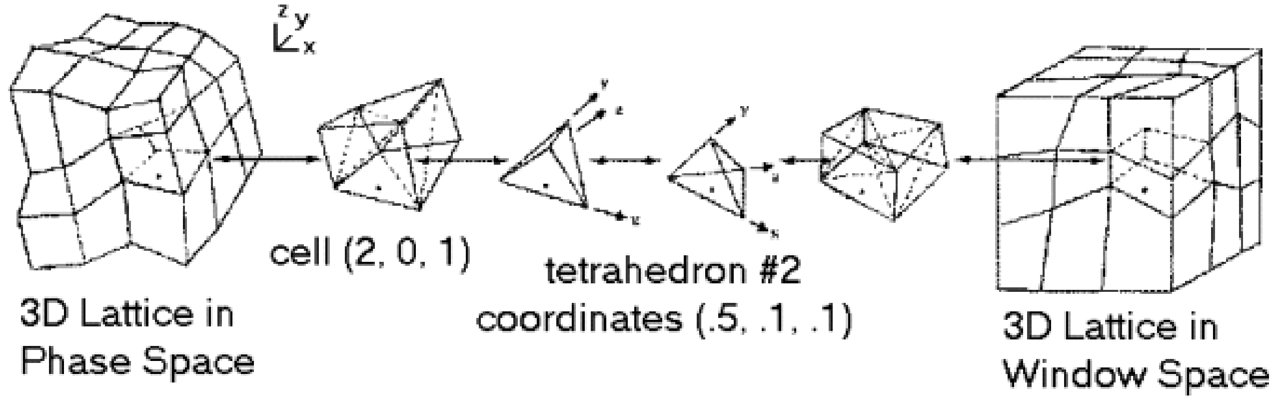


Fig. 4. A bijective mapping between Phase space and Window space.

Figure 4 shows the mapping of one point in window space to one point in phase space. First the lattice cell where the point belongs has to be searched. Then its coordinates in the cell are found based on a tetrahedral decomposition of the cell. The corresponding cell and coordinates in the phase space define the resultant point in the phase space. The inverse map is computed similarly. As a point's cell-coordinates exist and are unique under certain conditions which the cells satisfy (convexity, noncoincidence of corner vertices), this map from one space to cell-coordinates and back to another space exists and is bijective. As the map is a patch of linear functions continuously connected, it is continuous as well. This method is discussed in (Choi et. al. 1995).

3. Time in Manifold Interface

In musical tone production time is related to space through the physical structure of the musical instrument in two ways. First, the temporal structure of tone, known as frequency, is directly related to the spatial properties such as diameter and length designed to support vibrating surfaces and columns of air. Second, the physical structure characterizes the ecological condition in which performers action is applied. Performers manage time to modify the effective space of an instrument in order to change pitches and timbres. Each instrument presents characteristic discontinuities in action space layout. The characteristic continuity is often reflected in temporal articulations of the instrumental tones. Stylistic musical characteristics are founded upon these anomalies, bringing about the gesture repertoire of a performer. Virtuoso techniques to particular instruments have to do, in large part, with handling the physical discontinuity in a complete command to be able to achieve continuity when musical phrases demand it.

We might summarize by observing that the effective space is continuous while the action space layout is discontinuous. For example, a vibrating string is an effective space of continuous motion for sound production. When the string is depressed to a fingerboard, the depression is an action that alters the continuous vibrating length at a single point. The action of articulation makes a discontinuity. The depressed point of the string belongs to action space. In a sound computation algorithm we are facing the choice whether to model both the continuous space and the discontinuous space. Physically-based models have been successfully applied to the first problem while little attention has been given to the second, which is to model gestural space-time. The manifold interface brings an approach to models of gestural space-time by providing a normalized continuous interaction space. The parameters defining the models of continuous tone generation space-time, are projected into a continuous gesture time-space rather than the discontinuous space of traditional instruments.

3.1. Time management for Interactive data storage and retrieval with kinesthetic interaction

Interactivity in VR involves the discretization of a movement space into an abstract computational space as well as a temporal discretization. We refer to this discretization as spatio-temporal articulation. Articulation is determined both by the movements of an observer and the machine interpretation of the movements. The practice of articulation in the computer interface includes menus, graphical icons and manipulable graphical objects, speech or gesture pattern-recognition, and continuous parameter variation based upon a continuous input control signal. In each of these practices, interpretation of temporality of movements results from the software and hardware representation of an interacting observer's movement space.

Path, surface, and graphical notation

In the manifold interface observers apply kinesthetic energy to produce sounds. The interface captures the motion data and stores as trajectory. When we have ways storing and retrieving motion data we have an interactive notation method. The traditional concept of "score" is effectively an instructional data storage and retrieval system. However it would not be appropriate to refer to the data storage and retrieval method as notation. Symbolic representation, grouping, rearrangement capacity, and various transformation capabilities are adopted for visually articulating the ecology of the movement space (Eshkol 1993). The ecology is the positional organization of the shadowing from phase space to window space.

A interactive interface has been designed to enable a composer to stand in the CAVE and use the wand (a magnetic tracking device) to specify the location of generating points, by grasping each point and placing it at any location in 3D space. We arrange the generating points in the window space and recompute the projection to observe the sound differentiation that opens up between the points. As each generating point is positioned, the window space surrounding all the points reconfigures to accommodate their new relationship in all directions. By increasing the distance in window space between two generating points, we increases the degree of resolution of the parameter range between those points. The corresponding resolution in sound synthesis states is increased. By positioning generating points in window space, a synthesis algorithm can be "tuned" to focus the available window space on the fine control of regions of greatest auditory interest. The results can be immediately auditioned by using the wand to control sound synthesis, and manually scanning through the spatial neighborhood to test the parameter resolution.

Generating points may be positioned in a window space such that a hand or body movement tracing a gesture from one point to another supports an efficient performance, according to the natural disposition of the body in the space. When a manifold space has been configured, additional visualization and time management tools can be applied to compose notations for specific performance gestures and gesture sequences. Graphical paths and surfaces can be traced and shaped in the virtual space serving as a movement notation. The capability for tracing and shaping ensures a graphical and spatial representation of time and movement. Space-time calibration enables gestural notation and performance.

Paths and data management features

An observer making arm gestures with an input device such as a wand may draw traces in window space, as shown in Figure 5. We will refer to these traces as *paths*. The path is a sequence of positions of a cursor in the window space which correspond to the movement of a wand, thus scheduling the state changes in the system. The cursor position in the window space then maps to a point in a phase space through a callback function. A path through a phase space is a mapping from some time interval $[0, t_{max}]$ to the phase space. This map need not be bijective or continuous; a path can cross itself, or make abrupt jumps. The path is stored in the phase space, not in the window space. Thus a sequence of points of the path is defined with respect to the high-dimensional manifold, and its projection is defined with respect to the particular window space being used.

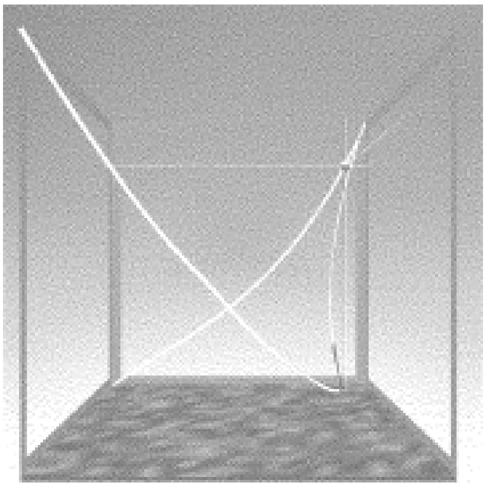


Fig. 5. A view of a path in a window space.

A path is initially recorded as a set of (n+1)-tuples, points in the Cartesian product of the n-dimensional phase space and one-dimensional time. This raw data is smoothed prior to be stored as a C++ path object. The smoothing is done by approximating the original path through this (n+1)-space with a sequence of spline curves. These splines are also in time as well as in "spatial" dimensions and are computed in the high-dimensional space. This smoothing may also be computed with a Genetic Algorithm (GA), where the bit vector representation of a sequence of spline segments is a vector of fixed-point control points and the fitness function approximates a least-squares error measure integrated over the original path. A GA provides an automated searching method to identify the shortest (smoothest) continuous mapping between path points in high-dimensional spaces (Goldberg 1989). In all approaches, smoothing acts as a low pass filter and it can eliminate the jitters originating in the magnetic tracking system.

In short we can say the path is drawn through a window space and encoded in a phase space. One of the important attribute of paths is a record/retrieval functionality which stores and returns temporal information as well as positional information from the user's activities. This enables users to reflect and revisit the previous decisions or movements in a time critical manner. There are many computer applications that record samples of control device position data at regular time intervals, in order to reproduce the temporal and spatial information in something roughly akin to tape recording and playback. The unique aspect of an MC path is they record and store path data in high-dimensional coordinates not in physical space coordinates. A visualization of a 3D path is projected from the high-dimensional path using the MC. In regular usage the MC environment continuously displays the 3D cursor as a projection from phase space to physical space, not the other way around.

Surfaces and fiducial points

To define sub-regions of interest and to simplify gestural interaction for three-dimensional control, we draw curved surfaces in the window space and constrain the sound control cursor to the surface. Surfaces articulate a window space as shown in Figure 6, and provide parameter regions tuned for interaction. A surface is also compatible with the locally two-dimensional behavior of a mouse. Paths may be recorded on a surface by gestures in two dimensions. The concept of *surface* is also useful in a 3D environment to provide regional differentiation with explicit geographical representation of subsets of control space.

In order to create surfaces we first investigate the window space and find a position producing interesting sounds. This position becomes an initial *fiducial point*, an identifier point located in both window space and phase space by recording parameter reference values embedded in each spatial representation. A fiducial point can be linked to other fiducial points in the window space by generating a surface through the points, creating a locally-displaced plane of continuous control values. A surface's edges and local degree of displacement may be altered by changing the positions of fiducial points or adding or removing points.

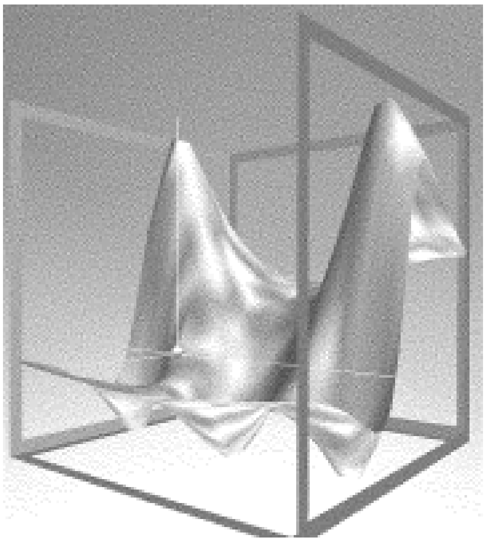


Fig. 6. A view of a surface in a window space.

Several methods may be employed for creating a smooth surface among a set of fiducial points. Splines are familiar tools in the graphics industry for constructing desired curved lines or surfaces. One drawback to their application for manifold surfaces is their potential need for multiple control points for assigning smoothing coefficients to a curve. By adding control points we lose the one-to-one relation between fiducial points and surface-defining points. To preserve this relation we use an averaging-displacement method for arriving at a smooth surface, passing through all the fiducial points and requiring no other surface-defining points. Beginning with a set of fiducial points defined by two planar dimensions and one perpendicular dimension, we apply the perpendicular displacement value of each fiducial point at regular intervals across the planar surface. The displacement at each interval across the surface is determined by weighting the displacement according to the distance from each fiducial point to a surface location, and averaging all of the weighted displacements for that location. This procedure is repeated at regular intervals across the surface.

Currently we constrain a surface to displacements in the direction perpendicular to the floor of the window space. This constraint means we do not permit two fiducial points to share the same horizontal (x,z) coordinates with different displacement values. This definition of a surface enables navigation on the surface in terms of purely horizontal movements, regardless of vertical displacement. The design prohibits the surface from overlapping itself along the vertical axis. Other features that are prohibited by this surface definition include complex surfaces containing folds or intersections, concave curves, and straight edges. These constraints are not inherent to the manifold projection between window space and phase space. Given enhanced navigation and geometric models, arbitrary surfaces could be used as control surfaces.

3.2. Tuning environmental clocks: Synchronization of Gestures and Computational Displays

In describing manifold interface components for composition and gesture-based performance we heretofore have made certain assumptions about the infrastructure for interactive graphics and sounds. Our discussion has implied there is a trivial connection between the following: real-time sound synthesis, graphical display in 3D space, and gestures measured in 3D space, such that gestures create instantaneous changes in sound and graphics. In practice this relation requires a painstaking design and implementation, and must be supported by an interface infrastructure that provides solutions to problems of latency and synchronization. It is impossible to implement a gesture-based performance interface without such an infrastructure. At the time of this writing few components are available from commercial suppliers or academic software packages. The majority have been designed and constructed in our lab in several stages. Below we briefly review the design principles and their realization in a virtual reality authoring environment where the manifold interface has been implemented.

The accumulation of latency between an observer's action and the system response, jeopardizes the observer's ability to construct mental models of interactivity. Interactivity is characterized by (1) how the observers access the control parameters linked to simulations, (2) how the observers monitor the coherence of the state change in simulations in response to the system-observer interplay, (3) how the observers can adaptively modify their performance strategies, and (4) whether the machine can be adaptive as well. Interactive models provide coherence laws for their internal dynamics. The coherence law particular to the given models under interaction governs the observable behavior of those models. This particularity supports an observer's ability to distinguish one interaction from another. Further, low-latency or "instantaneous" display supports intelligibility. The temporal criteria in VR authoring have to do with efficient reconfiguration of interactive capacity in a virtual scene and dynamics of services exchanged among parallel processes. The design process for defining manifolds and their spatial orientations for kinesthetic interaction requires an instantaneous VR authoring capability.

The rudimentary task in time management is the tuning of sampling rates of all component systems. This is to deliver coherent multi-modal feedback to observers with low-latency in order to support the observers' perception of causal links and recognition of feedback in the extended circularity in human-machine performance. We are accustomed to gesture-based real-time interactions with musical instruments. However, computers are not manufactured as real-time interactive instruments. To date, real-time computer-based displays are not completely reliable. In part this is because the temporal demands of human performance are not well understood, at least not by computing systems manufacturers. "Real-time" remains an ill-defined specification. For example, most operating systems do not provide Quality of Service measures for the CPU process scheduler, making task execution effectively probabilistic. In desktop applications this is a nuisance; for time-critical performance it is prohibitive. The problem is compounded by lack of synchronization in parallel processing. On a single processor many threads (separate computation tasks) can run "in parallel", but these tasks are processed serially and it is difficult to account for their relative load. Trail-and-error is the common method to determine if a processor will service parallel threads with enough regularity to keep abreast of real-time expectations. If a processor is not able to service all threads without latency, load balancing can be performed by hand, but this is usually performed offline after terminating operation of the application, not during a real-time performance. A calculated risk is always involved in real-time presentation.

Kinesthetic interaction demands a solution to preserve instantaneous feedback from actions to visual and auditory consequences. Our solution is to isolate each time-sensitive task on a separate thread, then to specify an internal clock rate for each thread, as well as a service rate for data flow between threads (Choi 1998a). This allows time to be prioritized for motion capture and audio control, since the channel from physical movement to auditory feedback is perceptually unforgiving of latency. On the other hand, motion picture displays can omit frames or delay the update of a frame without violating visual orientation. Graphics therefor provide for less time-critical display that sounds.

Figure 7 represents parallel processes in a performance system. For optimization, the service rate for the graphics thread is set at a lower frequency than the service rate for audio thread. To preserve the kinesthetic integrity of sound processes, gesture data is transmitted in parallel to the audio subsystem and graphical subsystem. Note in Figure 7 that control data does not flow from graphics to sounds. Control data flows from actions to numerical analysis then separately to each display modality. The temporal resolution of the graphical display is lower than the temporal resolution of the auditory display. Also, our ears are more sensitive to temporal changes than our eyes. In this parallel processing mode of synchronization various modalities provide time windows of different duration within which events may be perceived to correspond. We refer to "bounded synchronization" as the computational support for multi-resolution temporal relationship between modalities.

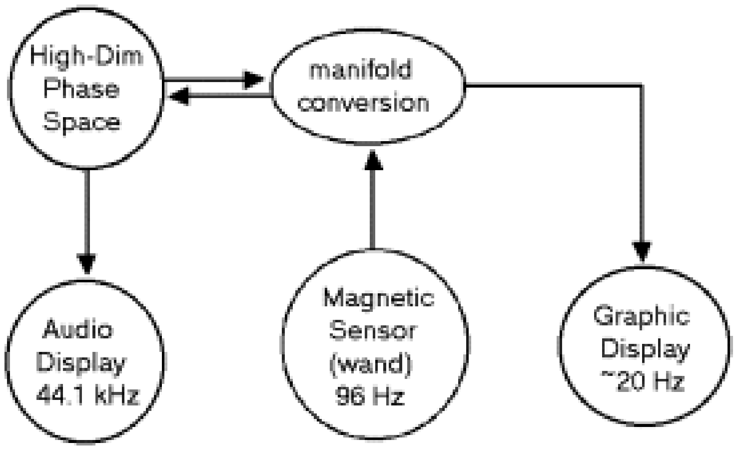


Fig. 7. Schematic representation of parallel processes requiring different processing rates for an input sensor controlling image and sound. Note there is no control flow from image to sound.

These solutions have been implemented in a virtual reality authoring system for multi-process time management. We refer to the system as a ScoreGraph because it provides support for specification of both linear (deterministic)and interactive (non-deterministic) time-sequenced events (Choi 1998b). ScoreGraph software consists of a kernel, a library of dynamic shared objects, and a configuration protocol that declares the run-time connectivity of objects. The kernel includes a scheduler that manages the CPU allocation to support all dynamic processes, which are registered as separate threads so that they can be independently managed according to priority of interactivity. Performance examples presented in (Choi 99) in this volume, were implemented using the ScoreGraph architecture.

4. Composition example of the MC space for gestural interaction

Figure 8 summarizes the Manifold Interface architecture. The architecture of the MC control flow can be summarized as a double-loop feedback and rendering cycle.

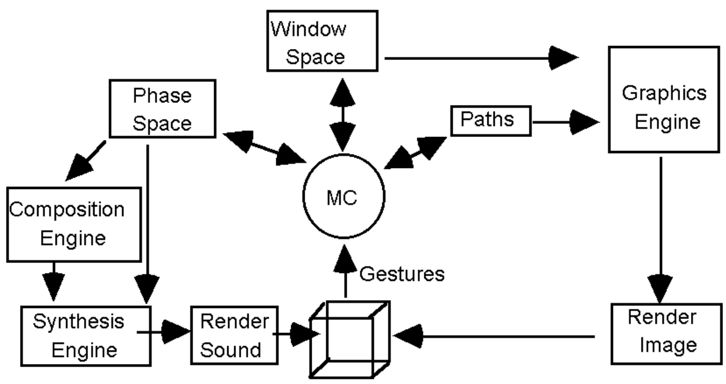


Fig. 8. The control flow of the MC environment.

The coherent presence of an observer with respect to a computing environment is supported by the calibration of the performance space according to an observer's orientation. Calibration is achieved when MC points in window space establish a correspondence between sound transformations and control trajectories accessible by a performer's movements. Listening to sounds generated by preceding movements, an observer's cognitive responses are in turn reflected in her or his next movements. Thus each movement is an articulation of decision-making based upon the evaluation of a preceding acoustic consequence and a proposition for the next.

The manifold interface has been applied to a large variety of sound synthesis models and provided for interaction in different performance and composition scenarios. This sections presents three example applications, two involving paths for tone generation, and one involving a surface notation for performance of a virtual environment composition.

4.1. Dynamically controlling vowel formant (FOF) synthesis

The CHANT algorithm synthesizes sound from a description of frequency spectrum characteristics and a simulation of the output of an excitor-resonator system (Rodet 1984). CHANT requires the specification of seven parameters for each formant in a spectrum. For best results the spectrum should vary over time. We installed the CHANT tone generator in our sound server to enable the control of CHANT tones in real time using the manifold interface. To define a window space we associate specific vowel sounds with generating points and position these points at regular intervals in the CAVE. This process involved four steps:

1. Identify sets of control parameter values that produce specific vowel sounds.
2. For each vowel, create a generating point in order to associate the control parameter values with a unique 3D position in a window space.
3. Compute the embedding such that all positions in the window space have acoustic properties consistent with those of the generating points (smooth transitions occur between generating points).
4. For the examples in Figure 9, create a path in the window space that visits each generating point.

Figure 9 presents audio signals made up of three formants, which appear as three peaks in each FFT display. Three formants require 21 CHANT parameters. We decided to hold some parameters fixed while others varied along the control path. For each generating point we defined eight variable parameters: the center frequency and bandwidth of the first formant, and the center frequency, bandwidth and amplitude of the other two formants. Amplitude is measured in dB and center frequency and bandwidth in Hertz. Each generating point was made up of an array of values for these eight parameters. Four generating points were created, each producing a unique vowel sound (/u/, /i/, /e/, or /a:/) and each vowel point was positioned at a unique corner in the window space.

After the generating points were positioned a path was created passing once through each of the generating points. Signals from five locations on this path are presented in figure 9 (next page).

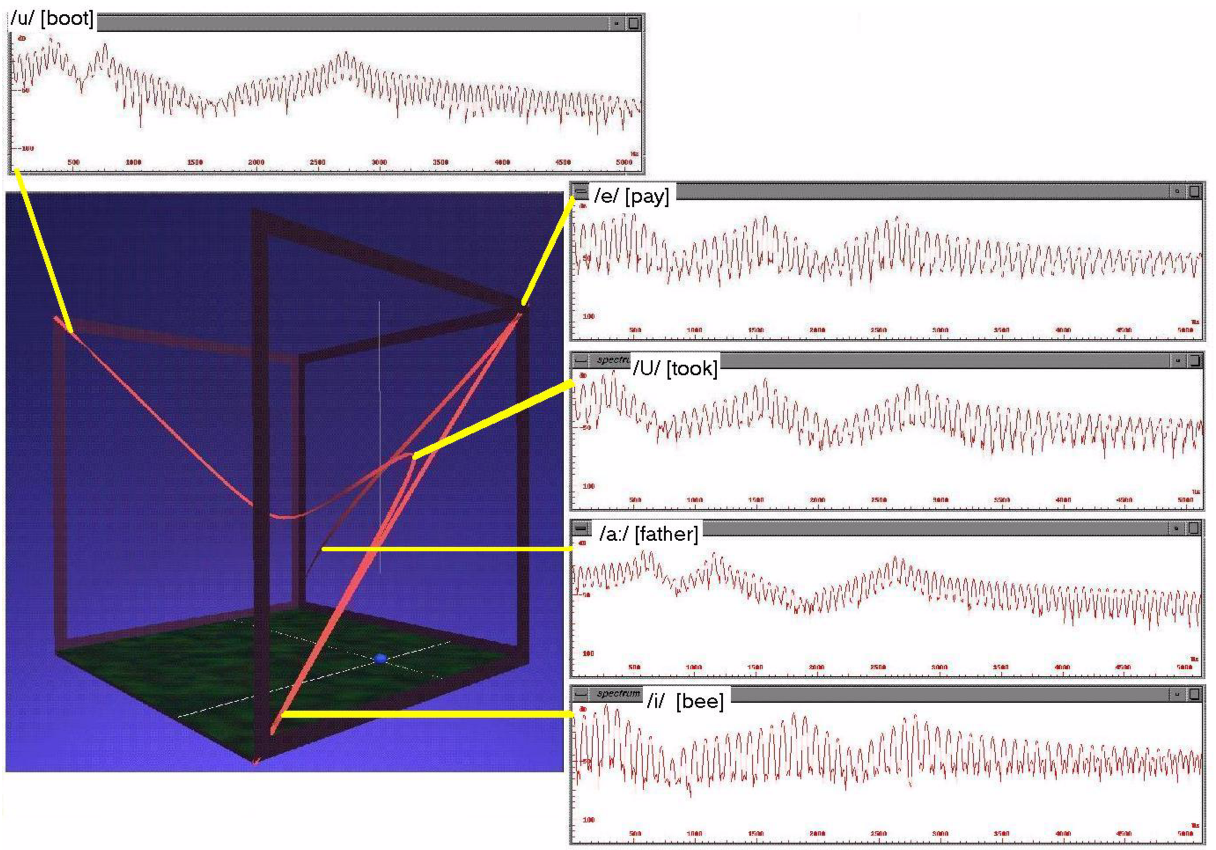


Fig. 9. Vowel sounds create Vowel sounds created by a path controlling CHANT (International Phonetic Alphabet [English]).

Intermediate positions on the path produce intermediate vowel sounds, such as the /U/ which occurs in a location toward the center of the window space. In figure 9 the cursor on the floor is positioned so that its vertical axis intersects the path at the point of the intermediate vowel, /U/. When the path is traversed a continuous tone transforms from one vowel to the next. The interpolation generated by the path produces intermediate vowel-like tones. Neither the CHANT parameters nor the configuration of the window space provide articulation of the continuous tone generator.

4.2. Creating and playing a manifold path to control two tones from a physically-based flute model

Figure 10 shows generating points in a window space in the CAVE manifold interface. Each generating point represents two parameter sets for two tone generators, each applying a physically-based flute model. The control parameters for each tone are frequency, breath pressure, jet delay (angle of breath stream), noise gain, vibrato rate and vibrato gain, making the phase space 12 dimensions for the pair of tones. The arrays of values for each generating point are indicated in Figure 10. Each generating point is tuned to a specific duet relationship between the two tones. Articulations and rhythms are created by the vibrato gain and vibrato rate parameters.

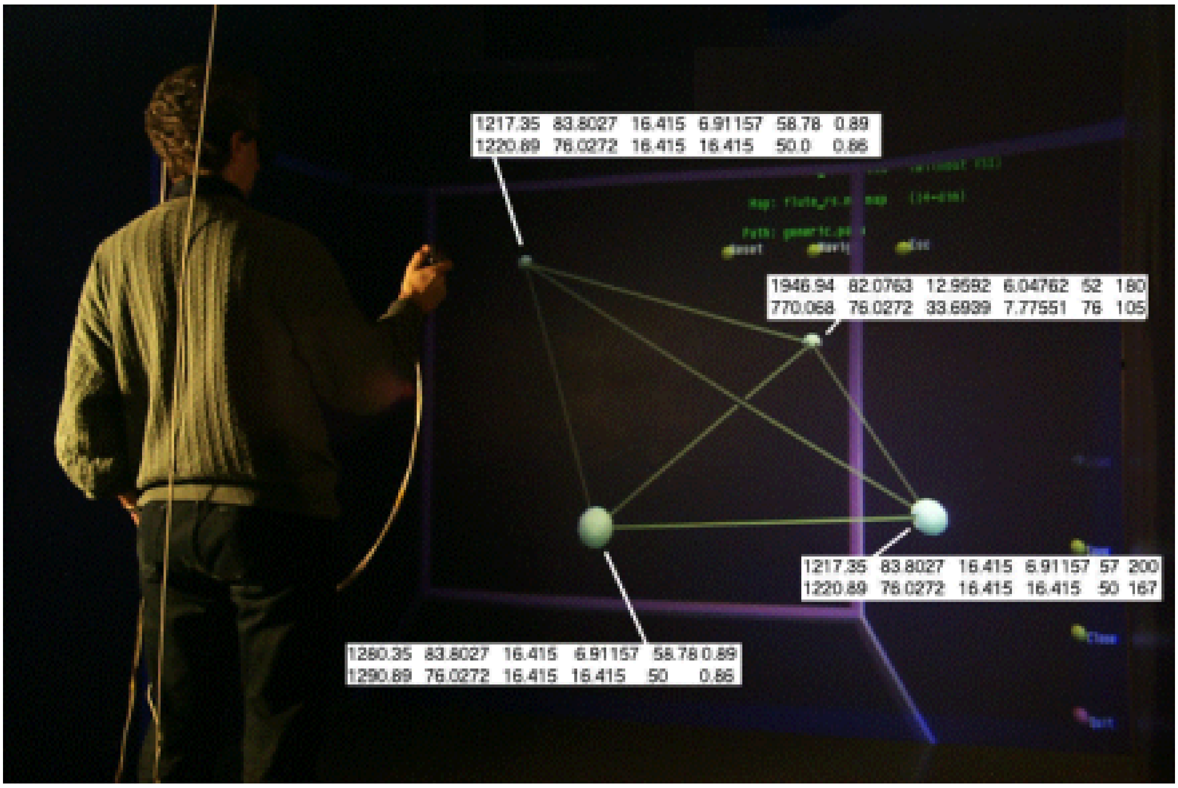


Fig. 10. Four generating points in the manifold interface (in a CAVE). The array of parameter values at each point is mapped to the parameters of two physically-based flute models, playing a duet. The flute control parameters are Frequency, Breath Pressure, Noise Gain, Jet Delay, Vibrato Gain, and Vibrato Frequency (Cook 1999). Frequency and Vibrato are in Hz; other parameters are in normalized floating point range [0, 127].

Figure 11 shows the creation of a manifold path in the CAVE. The path is drawn in the CAVE by controlling a spherical cursor using the wand, a 3D magnetic tracking device with indicator buttons,. The path records orientation as well as position coordinates. The path appears in the CAVE as a ribbon, with the ribbon surface normal indicating the angle of the wand orientation. The composer may select any point in the CAVE space to begin and end a path, and a path can cross over itself, revisiting any point. The rate of path traversal affects the rate of transitions in pitch, timbre and rhythm.

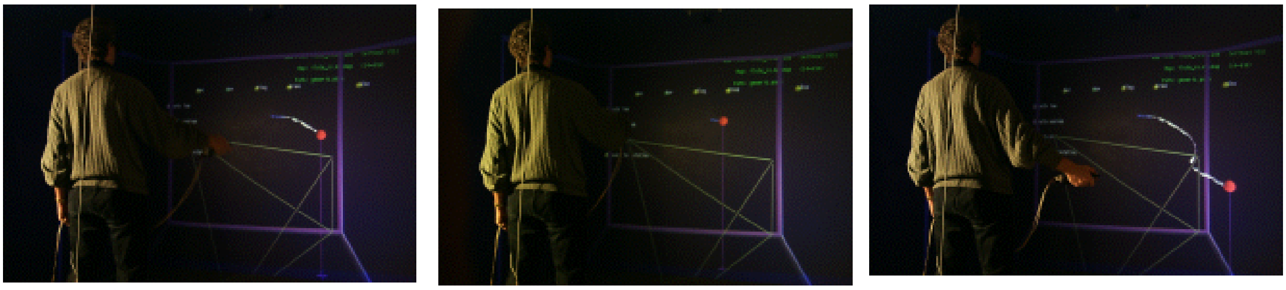


Fig. 11. A manifold path created in the CAVE using the manifold interface. Cursor position is controlled using the wand, a hand-held magnetic sensor with indicator buttons.

Figure 12a and 12b present animated graphic sequences from the CAVE, recorded by converting the image from the front wall of the CAVE to video frames. The video shows in a monocular image what is displayed on the CAVE wall in stereo for a user wearing stereo glasses. The CAVE users are not visible in these examples. Figure 12a video demonstrates the sound of a generating point and neighboring area, as the cursor is moved in 3D space exploring the region surrounding this point. Figure 12b video demonstrates a path being created and replayed. There are two modes for replaying a path, automated playback and free play. Automated playback reproduces the time information associated with the original performance sequence of drawing the path. Free play enables the path spatial positions as control data without the original time information. In free play the performer interactively traverses the path in either direction using the wand to control the path cursor position. Figure 12b video concludes with an automated path playback.

Fig. 12a. Videos available in the original CD-Rom version.
(left). Video demonstrating audio feedback from Generating Points and surrounding window space regions, controlling two sound synthesis voices using a physically-based model of a flute.
Fig 12b. (right). Video demonstrating the creation of a Path in the window space
defined by generating points in Figure 12a.

4.3 Editing a manifold surface in a window space and navigating the surface to control granular synthesis

Figure 13 is a video demonstrating generating points applied to the control of granular synthesis. Each generating point controls three granular synthesis tone generators that are mixed into a single audio output. As in classical granular synthesis, a constrained random number generator is used to select parameter values for individual grains, based upon a specified Average (center value) and Range (minimum and maximum value). There are six control parameters for each generator: Grain Duration Average, Grain Duration Range, Onset Rate Average, Onset Rate Range, Tuning Average, and Tuning Range. Given three parallel grain generators mixed to the audio output, altogether each generating point controls an 18-dimensional space. The grain generators use sound sample wavetables for sound source waveforms. Each generator provides a sound source of running water on a scale from small stream to ocean. The generating points are arranged in the CAVE so that the small scale water sounds are towards the floor and in the center, while the large bodies of water are at shoulder height and away from the center.

Fig. 13.Video available in the original CD-Rom version.. Video demonstrating generating points in a window space, applied to the control.

Figure 14 shows the deformation (editing) of a manifold surface using the wand. Figure 15 is a video demonstrating editing a manifold surface for performance with auditory feedback from the granular synthesis space. The surface is initialized from a pre-existing form, although it is also possible to create a surface from scratch. The surface is reshaped by re-positioning the surface control points (the fiducial points discussed in Section 3). These points are visualized as spheres attached to the surface. The camera view of the surface moves up and down controlled by a tracking device attached to the stereo glasses worn by the composer in the CAVE. When the composer looks under the surface, the camera view causes the surface to appear to rise in space. When the composer stands up, the surface lowers. Eventually the surface is re-formed to locate the small scale water sounds in a low center pocket. The sounds increase in scale as you climb the slopes of the surface, with different details and tunings at different positions around the circumference of the surface. The peaks and elevated ridges provide the water sounds of the largest scale. Figure 16 is a video demonstrating the creation of a granular synthesis sequence using the completed surface. The composer navigates a cursor attached to the surface, and we hear the granular synthesis in smooth transitions, from small stream, to lake, to ocean waves.

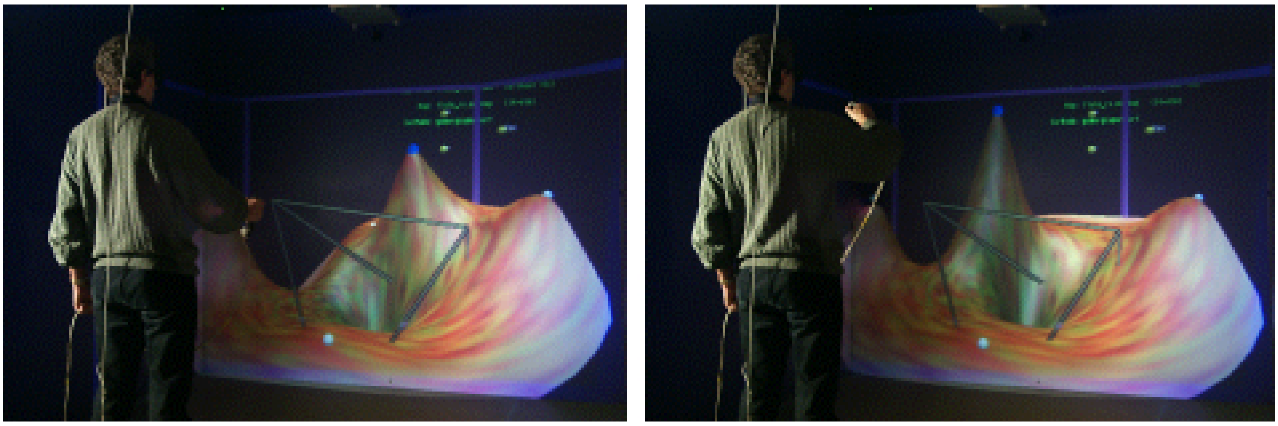


Fig. 14. Editing a manifold surface in the CAVE.

Fig. 15.Video available in the original CD-Rom version. Video demostrating editing a manifold surface in the CAVE, with auditory feedback from granular synthesis.

Fig. 16.Video available in the original CD-Rom version.. Video demonstrating the performance of audio sequences by navigating on the surface with a cursor.

5. Conclusions

The usage of the manifold interface can be summarized in two ways. First, the control points feature the ability to constantly reorganize and customize parameter space. One can stretch and compress the distance between the control points while monitoring the results from altered resolutions. This has been applied suitably for artistic compositions. The second, the automated algorithms provided in the interface such as the projection technique features a consistency offering ways of preserving integrity of numeric values of parameters. One can preserve the geometric relationship internal to the system under investigation with the rigorous use of the automated algorithms. This is more suitable for most scientific investigation.

The utility of the manifold interface has been demonstrated in two composition approaches. The following functions are visually projected in the composition and take an active part with live performances:

graphical notation

interactive performance interface

computer graphic representation of interactive performance space

Additional functions are active but hidden from the performance scene, and are usually engaged directly in non-performance mode:

pre-compositional assistant for investigating *n*-parameter space

interactive device coupling with sound synthesis models

interactive space coupling with sound synthesis models

The manifold paradigm continues to provide a reliable general procedure for orientation to high-dimensional parameter control spaces, in the creation of compositions for sound alone, and compositions integrating movement with sound synthesis for real-time performance, with or without visual display.

Research and development is underway for porting the manifold engines from a stand-alone tool and directly incorporating their functions into the graphical and sound synthesis libraries supporting composition and performance systems.

6. Appendix 1

Anatomy of an Immersive Performance Environment and peripheral devices

Though the manifold interface project is not bound to the high-tech environment such as virtual reality most of the present research has been performed in the CAVE™ (DeFanti 1992). Since the discussion of the manifold interface in this paper involves the CAVE as a case platform, this section introduces the anatomy of the system and peripheral devices. CAVE is a virtual reality theater consisting of a 10' cube built from projection surfaces. Computer-based graphics are rendered in a geometric model of 3D space that shares the same virtual dimensions as the physical CAVE. Graphics hardware and software libraries provide basic 3D rendering and display functions for projection of perspective views onto screen surfaces. Stereo images are projected on the walls and floor and synchronized in the corners and along the edges of the screens (Currently there are roughly 50 CAVES in the world, each having three to six walls.) Observers are located inside the cube, surrounded by images, wearing stereo glasses for 3D depth cues. One observer wears a head-mounted magnetic tracking device, allowing the CAVE application to determine the position and orientation of that person's stance and point of view. The stereo and 3D perspective of the images can be computed to match the exact line of sight of the tracked observer. Other observers in the CAVE will see an approximation of correct stereo projection, depending upon how close they are to the active observer. The stereo glasses electronically shutter each eye in turn, synchronized to the computer's alternating display of left and right-eye image frames. An infrared light beam controlled by the computer keeps the glasses in sync with the graphical frames. The shutter rate is 90 Hz to avoid a flicker that can be visible at 60 Hz. For the Manifold Interface application, the accurate 3D depth cues enhance the normalized visual 3D representation of a manifold surface. Graphics frame rates vary from 15-30 Hz, depending upon the complexity of the scene, roughly corresponding to the number of polygons in the image or the complexity of equations of motion. When gestures are guided by graphical feedback displayed in 3D stereo, the frame rate has an affect on the resulting accuracy of motion data capture, which in turn affects the resolution of gesture-based control signals.

Gestural data can be input from a number of control devices. The standard controller is the wand, a hand-held baton-like 3D mouse with 3 buttons and a magnetic tracking device. The tracker measures three position values and three orientation values at a rate of approximately 48 Hz. Latency in the current system is less than 10 msec. The wand can measure hand position changes and arm movements, but captures no information about the posture of the hand or fingers. The CAVE layout encourages observer's gestural interactions in all directions, and full arm motions are not unusual during a working session. The Manifold Interface can be configured as a full-size CAVE display, supporting large-format projection of the interactive graphical representations of manifold surfaces. The visual scale of these interactive surfaces is compatible with full-rotation arm movements, full-body leaning and facing, and walking by a performer. Multiple projection screens preserve the visual field when an observer walks around an object to view it from different sides. Head-tracking supports circumference tours of 3D objects by relocating the object image on different screens to accommodate the observer's orbital path in the CAVE. This display format provides a good view for creating and editing Manifolds and interactive surfaces for performance works. The manifold surfaces preserve their 3D space definition and interactive capability when they are exported from the CAVE and presented in a single-screen format. Additional control devices are available, including tablets, joysticks, gloves, wireless palmtop computers, video-based hand motion tracking (Pavlovic et. al. 1997, Choi 1999), and foot-mounted pressure-sensitive Cyberboots (Choi and Ricci 1997). All of these can be applied to the Manifold Interface control paradigm. We refer to the use of the wand as the normalized manifold interaction.

Audio is generated by the Virtual Sound Server (VSS), a research tool developed by the authors through the NCSA Audio Development Group (Bargar et. al. 1994). VSS provides message-passing, scheduling, data mapping, software synthesis and synchronization services. VSS supports real-time interactive sound synthesis and algorithmic composition as well as non-real-time synthesis for offline sound rendering tasks. We apply the term "Sound Authoring" to refer to the layer of mediation between interactive events, graphics-based events and sound synthesis. For example, the Manifold Controller described in this paper is implemented as a high-dimensional mapping algorithm in VSS. A performance application links with VSS libraries at compile time. The application registers messages with VSS requesting sounds for dynamic and interactive processes. At run-time these processes pass messages to VSS and corresponding sounds are synthesized in real-time, synchronized according to a pre-determined degree of detail. VSS has been applied to the performance control of external synthesis engines including Ircam's jMax software, and MIDI-based hardware devices.

Acknowledgements

I wish to thank Camille Goudeseune for his contributions toward implementation of the manifold interface software, and Alex Betts for bringing the manifold interface into computer graphics and virtual reality modeling and performance environments. Thanks also to Robin Bargar for research coordination.

7. References

Bargar, R. 1997. "Authoring intelligent sound for synchronous human-computer interaction." In *Kansei, The Technology of Emotion*. Proceedings of the AIMI International Workshop, A. Camurri, ed. Genoa: Associazione di Informatica Musicale Italiana, October 3-4, 1997, pp. 177-188.

Choi, I. 1994. "Interactive exploration of a chaotic oscillator for generating musical signals in real-time concert performance." *Journal of the Franklin Institute*, 331B(6): 785-818.

———. 1997a. "Interactivity vs. control: Human-machine Performance basis of emotion." *Proceedings of the AIMI International Workshop, Kansei: The Technology of Emotion*, A. Camurri, ed. Genoa: Associazione di Informatica Musicale Italiana, pp. 24-35.

———. 1998a. "Human - Machine Performance Configuration for Multidimensional and Multi-modal Interaction in Virtual Environments." *Proceedings of the 4th Annual Symposium on Human Interaction with Complex Systems*, Dayton, Ohio, pp. 99-112.

———. 1998b. "ScoreGraph: dynamically activated connectivity among parallel processes for interactive computer music performance." *Proceedings of the International Computer Music Conference*, San Francisco: International Computer Music Association, pp. 527-35.

———. 2000. "Gestural Primitives and the context for computational processing in an interactive performance system." In this volume.

Choi, I., Bargar, R., and Goudeseune, C. 1995. "A Manifold Interface for a high dimensional control space." *Proceedings of the International Computer Music Conference*, San Francisco: International Computer Music Association, pp. 385-392.

Choi, I., and Ricci, C. 1997. "Foot-mounted gesture detection and its application in a virtual environment." *Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4248-4253.

Cook, P. 1995. "Hierarchical System for Controlling Synthesis by Physical Modeling." *Proceedings of the International Computer Music Conference*, San Francisco: International Computer Music Association, pp. 108-109.

Cook, P. 1999. "STK: a set of audio signal processing C++ classes and instruments for music synthesis." Synthesis Tool Kit new release announcement. http://www-ccrma.stanford.edu/CCRMA/Software/STK/.

DeFanti, T., C. Cruz-Neira, D. Sandin, R. Kenyon, R., and J. Hart. 1996. "The CAVE," *Communications of the ACM*, V. 35, No. 6.

Eshkol. N. 1973. *Moving Writing Reading*, Holon, Israel: The Movement Notation Society.

Gibson, J. J. 1966. *The senses considered as perceptual system*. Boston: Houghton-Mifflin.

———. 1979. *The ecological approach to visual perception*. Boston: Houghton-Mifflin.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Hunt, and Kirk. 2000. "Mapping Strategies for Musical Performance." In this volume.

Pavlovic, V., Sharma, R., and Huang, T. 1997. "Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review". *IEEE Transactions on Pattern Analysis*.

Rodet, X., Y. Potard, and J.B. Barrière. 1984. "the CHANT Project: From the Synthesis of the Singing Voice to Synthesis in General." *Computer Music Journal*, 8(3): 15-31.

Smith, J. O. 1992. "Physical Modeling Using Waveguides", *Computer Music Journal* 16(4): 74-87.

