University of California Santa Barbara

Standing Waves: a Multimodal, Motion-capture Composition in the AlloSphere

A Thesis Project submitted in partial satisfaction of the requirements for the
degree Master of Science in Media Arts and Technology

by Joshua Dickinson

Committee in charge:

Curtis Roads, Chair

Matthew Wright, Co-chair

JoAnn Kuchera-Morin

Clarence Barlow

March 2013

Standing Waves: a multimodal, motion-capture composition in the AlloSphere

Joshua Dickinson

March 28, 2013

Abstract: *An interactive, audio-visual composition based on the universal wave equation and additive synthesis is presented. Visualization, sonification, and motion-capture interfaces are discussed in relation to their use within the piece. The technique of "density plot synthesis" is introduced to describe the technique of controlling sound-source amplitude levels through the wave equation or similar data sources.*

Committee

Curtis Roads (chair) ______________________________

Matthew Wright (co-chair) _________________________

Clarence Barlow __________________________________

JoAnn Kuchera-Morin _____________________________

1 Introduction

Standing Waves is an audio-visual installation designed for the AlloSphere, an immersive multimedia instrument being built at UCSB. The piece presents an interactive visualization of two-dimensional wave propagation projected in three dimensions around the surface of a sphere. This simulation is then sonified through a variation of additive synthesis and spectral decomposition, and the resulting audio is spatialized around the perimeter of the performance space.

Users are able to interact with and control the combined audio-visual synthesizer through a motion-capture interface and gestural mapping system. The piece's form is structured through a series of modules that can run while being guided by user input, as well as in a semi-autonomous "installation mode" when limited or no user interaction is detected.

This document will discuss the artistic and technical components used to create the piece, while focusing specifically on aspects of tracker-based motion-capture interfaces and their expressive potential in media art. The technique of "density plot synthesis" will be introduced and described in terms of its implementation here. Previous installations and works created by the author will be described briefly with examples of how elements from each were combined and used in the final composition. The following issues will be addressed in relation to their role within this piece and multimedia composition more generally:

1. Visualization and sonification of a two-dimensional wave equation
2. Motion-capture controlled audio synthesis: additive, granular, density plot
3. Motion-capture controlled spatialization: literal position, division of axes
4. Multi-controller mapping strategies
5. Allocentric vs. Egocentric interface design

2 Motivation

Standing Waves was created in order to explore the synthesis possibilities of two dimensional wave propagation in audio and visual form. It was hoped that an immersive, appealing, and novel composition could be created by allowing users to interact with such a system through a motion-capture interface. While glove-based interactions have been possible in some form for decades, they are still a relatively uncommon experience for the average user, especially when used to control 3D environments at room-scale.

2.1 Perceptualization

The piece is a study on “perceptualization,” or the translation of numerical data to one or more sensory representations, as well as inter-sensory correlation and interaction.

> "The term ’perceptualization’...describes the translation of signals and information to modalities that appeal to any of the human senses. As such, it generalizes the terms ’visualization’ and ’sonification’ to include all other senses. We study such perceptualizations, with particular focus on how properties of a perception systems can be used to optimally convey information"[14].

The wave equation is perceptualized here, by using it as a synthesis tool, translating it into audio and visual form. This transformation is accomplished through the use of rule sets determining how to treat wave density plots created by the calculation as it is run in real time. Aspects of the resulting visual system are in turn used as a basis for the corresponding sonification and vice versa. As the resulting forms are projected around the installation space, the user interacts with the simulations by moving through and around them, which then alters their appearance around the room. In this way the installation re-perceptualizes movement as sight and sound. It was created in order to answer these research questions:

1. How can wave propagation be sonified?
2. What methods can be used to visualize sound synthesis techniques such as additive synthesis and spectral freeze?
3. Can such a visualization and sonification pair interact with and inform one another?
4. How can “perceptualizations” of this nature be interacted with using motion-capture?
5. Can these constituent elements be used as the basis of a composition?

2.2 The AlloSphere

The physical space for which the piece was composed is particularly important because of the site-specific nature of the installation. The AlloSphere is a "large sphere, ten meters in diameter, made of perforated aluminum, that is designed to provide multimodal representations of large-scale data in a fully immersive, 3D environment"[10]. It is currently equipped with a group of 12 projectors capable of displaying stereo-3D content seamlessly around its top hemisphere and stretching down to about 20 degrees below its horizon. It is also equipped with a 54.1 channel audio system, installed in three rings around its perimeter with the addition of a subwoofer beneath its base. Together these systems enable the projection of audio-visual content in any direction around the sphere, as the audience and participants stand at its center along a narrow bridge.

Because of its spherical structure, it was particularly well-suited for accomplishing one of the goals of this piece, which is to make the audience feel as if they are exploring the “inside” of a sound wave as it propagates through space and time. Audio is able to be projected from nearly any angle, which creates an intense illusion of space. Likewise, the spherical projection area eases the process of immersion; Visually, the piece’s opening sequence places the audience inside of a spherical wave-field roughly matching the real form of the dome. It was thought that mimicking the contours of the physical space before more complex and otherworldly shapes were displayed, would ease the transition into a virtual world and thus heighten its illusion.

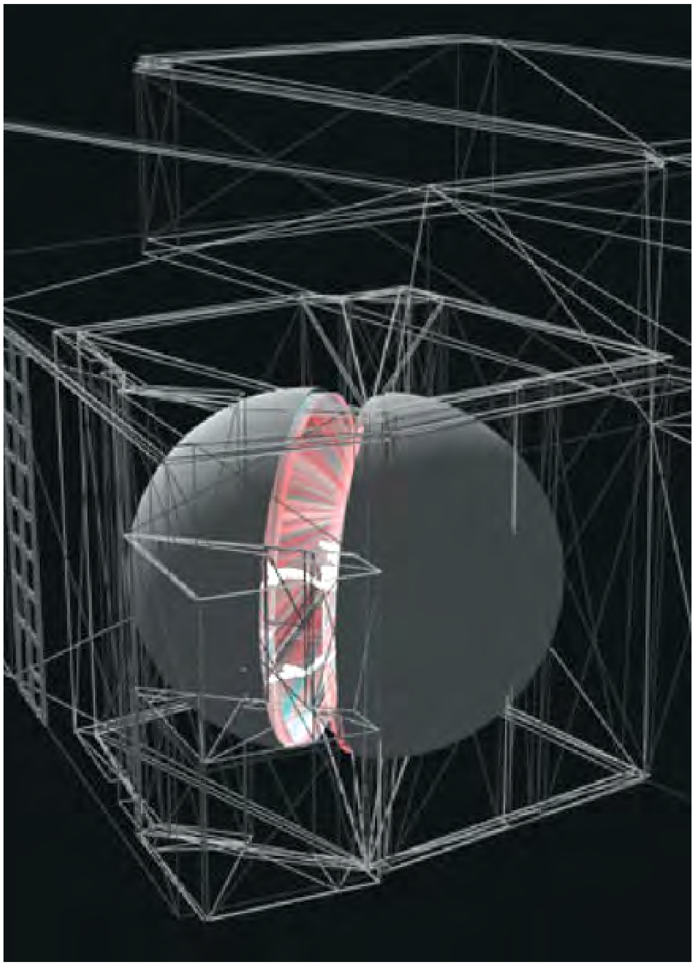


Figure 1: The AlloSphere

2.3 Phasespace Tracking System

The Phasespace-brand motion capture system used to control this piece includes gloves with 8 active LED markers on each hand: 4 on the finger tips, 2 on the thumb, and 2 at the base of the fingers, as can be seen in figure 2. Fourteen overhead cameras track the motion of each LED in three dimensions, differentiating them through a unique optical signature created through pulse-width modulation. The ability to track each point along the glove individually opens up the potential for very complex mapping strategies, such as utilizing the spatial relationship between any pair of points or from a single point and the room's center.

Figure 2: Phasespace gloves

In practice, it was discovered that the most successful interfaces created through this system made use of each hand's centroid, or its center point calculated by taking the average of all 8 markers along each glove (arithmetic mean). Centroids are a drastic reduction of the information provided by the entire glove, however they have proven to be effective and more reliable than working with individual markers because of the problem of occlusion: often times markers will become temporarily occluded by parts of the performer's body or other items within the room, thus preventing the system from providing positional information for every marker at every frame.

An interface that uses this system must compensate for these irregularities by either ignoring markers that are not updated by the system or by using a custom model to guess the location of unreported markers. There are many possibilities for such a system: a location could be guessed, for instance, by adding a distance vector to the centroid calculated by taking the difference between a marker's last confirmed location and the glove's centroid during this frame. This however, assumes that the hand's shape has remained constant between frames. Models based on higher derivatives of position such as velocity and acceleration, or an anatomical model are other possibilities for making positional estimates. Using centroid values as the main method of interaction is a simpler way to minimize the error caused by missed markers, and is the solution used in throughout this piece. One of the main disadvantages of this technique is that even if the hand is held still, the centroid will move as different markers are missed and re-added to the group, thereby affecting its calculated position.

Despite the problem of occlusion, there are many ways in which individual markers can be utilized. It was found that a “spread” value for each hand could be calculated by taking the average distance of each marker on the hand from the pre-calculated centroid (average absolute deviation). If a hand was completely outstretched, it would have a large spread value, and if it were crunched together, this value would drop. Using techniques for gestural detection developed with Ritesh Lala[15], it was possible to reliably know when a user touched his or her fingers together. Touches of the thumb and index finger, for instance, could then be used to trigger an event or change controller modes. False detection of touching or releasing two fingers was prevented by only allowing a touch or release event to be triggered during a frame in which both markers involved were reported by the system. This guaranteed that nothing was triggered because the index finger marker moved close to the last reported position of the thumb tip marker, for example.

2.3.1 Definition of Axes

Unfortunately it seems that many standards exist for defining X, Y, and Z axes in relation to the space in which the motion-capture system is installed, often causing confusion when referring to one axis or another. Throughout this text the axes are described as follows, so that the Y-axis always fights gravity and (where applicable) the Z-axis points in and out of the primary projection or screen. In the symmetrical AlloSphere, the “main” screen is taken to be the portion of the spherical screen perpendicular to the long axis of the bridge, so that the X-axis runs the length of the bridge, the Z-axis runs along its (shorter) width.

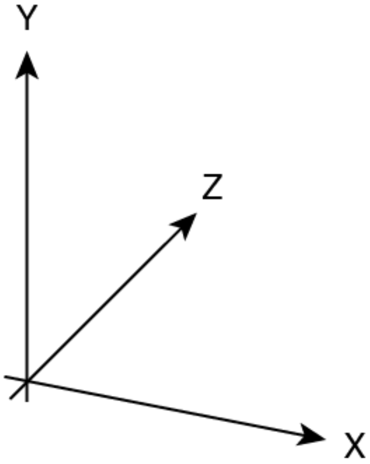


Figure 3: Standardized Axes, where y corresponds to gravity-up/down, and x and z form a 2-dimensional graph on the room's floor

3 Wave Propagation

> "The wave equation is a second-order linear partial differential equation that describes the propagation of a variety of electromagnetic, acoustic, and fluid waves" [19].

The engine of the piece is a one-hundred-node-tall by one-hundred-node-wide, two-dimensional wave propagation graph connected at each end in a toroidal structure. The height of the wave at each location is calculated in discrete time steps using the finite difference method (FDM), given by the formula:

$$\mu_{i,j}^{n+1} = 2\mu_{i,j}^{n} - \mu_{i,j}^{n-1} + c(dt/dx)^2 \left[\mu_{i-1,j}^{n} + \mu_{i,j-1}^{n} + \mu_{i+1,j}^{n} + \mu_{i,j+1}^{n} - 4\mu_{i,j}^{n}\right]$$

The formula can be summarized as follows: For each node in row i, column j, the node's next height is calculated by adding two times its current height, minus its previous height, plus a dampening coefficient c times $deltaT$ over $deltaX$ squared times the sum of the current heights of its four neighboring nodes (top, bottom, left, right) minus four times its current height. $DeltaT$ and $deltaX$ are the time step and distance step of the equation respectively, simply implemented here as two constant variables (0.001/0.003). The dampening factor c is a constant determining the amount of resonance in the system, or how long waves will continue propagating once triggered. Values ranging from 0.93 to 0.97 were used within this piece for different effects; the former to prevent energy from lingering too long within the system, and the latter to create long, resonant passages.

User interaction adds energy into the system; As he or she moves around the space, a graph representing the wave network is mapped to the dimensions of the AlloSphere's bridge along the X and Z axes and the location of the user's hands at each moment create an energy peak in the system at the corresponding point on the wave grid. During different sections of the piece, the size of the peak is either fixed at 1.0, or determined by the absolute sum of acceleration of the glove's centroid in each dimension. In "fixed mode," when standing still these energy peaks appear as stationary or "standing" points, however as they are moved their energy is propagated throughout the network following the rules of a two-dimensional wave system made to connect at each edge. In terms of the equation, this connection is created by applying a modulo operator to the row and column indices i and j, so that they become 0 when equal to the row or column length respectively. Because of this connection, waves that reach one edge of the graph continue in the same direction, wrapping around to i its opposite end.

A hand's location is determined by the centroid calculated through the positions of each marker along the glove. A roughly 3 meter by 1 meter area on top of the AlloSphere's bridge is assigned as the "bounds" of the field and the glove's centroid location is scaled and linearly interpolated to create an energy peak at the corresponding point in the wave node system. For instance, if the user were to extend his or her hand to the top-left corner of the bounded area, it would create a peak on the top-left corner (row 0, column 0) of the wave graph. As he or she walked to the opposite corner of the bridge, the peak would move diagonally across the system until it also reached the opposite corner (row 99, column 99). These bounds act as hard limits and therefore do not wrap around to the opposite end of the graph once exceeded. This relationship can be summarized as follows, where X, Y and Z refer to a glove's centroid and the height of the peak is determined by acceleration:

$$\mu_{i,j}^{n} = |x''| + |y''| + |z''|$$

$$j = (100.0/3.0) * x; 0 <= j <= 99$$

$$i = (100.0/1.0) * z; 0 <= i <= 99$$

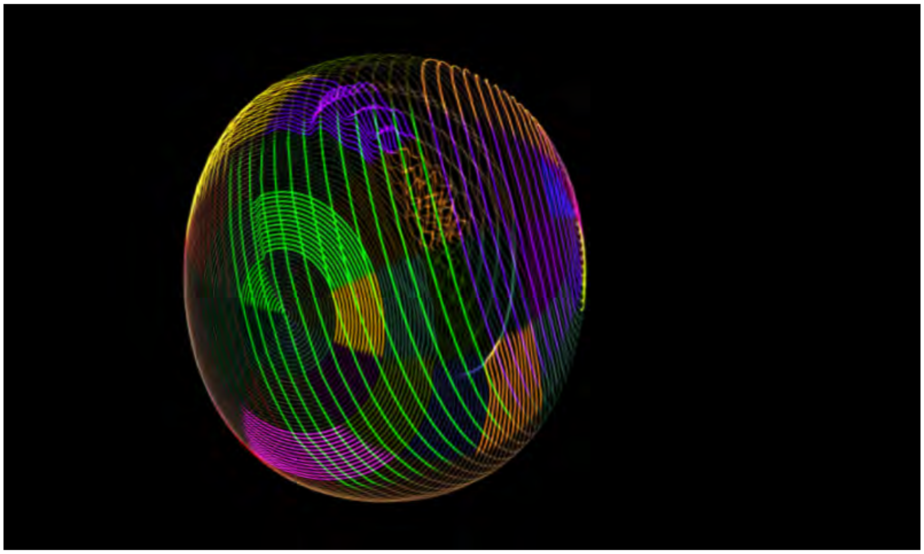
This simple rule of mapping an energy peak to the center position of a hand allows the user to "push and "pull" energy throughout the system, which then dissipates in complex patterns as it moves along the graph. Moving slowly will result in small ripples, while moving in quick, large gestures will cause torrents of energy to move through the system. Many variations on this basic mapping strategy were experimented with during the creation of the piece, each having a unique effect on the overall nature of interaction. One such variations, where energy spikes are quantized into discrete impulses, is used during the fourth movement.

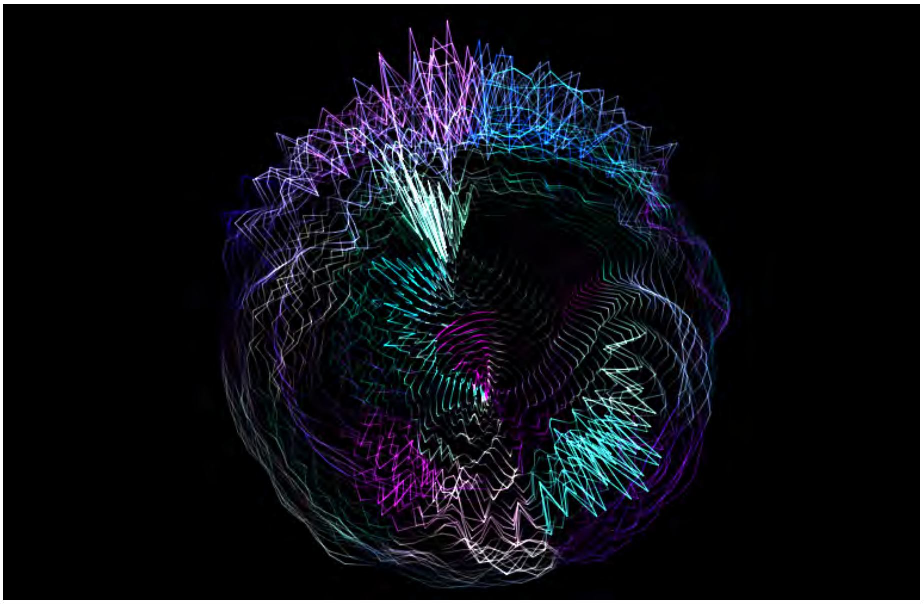
4 Visualization

The dynamic energy map created through this process is visualized by drawing the wave field as points connected by line segments and wrapped around a spherical 3D mesh. Because of this transformation, the row and column values i and j become latitude and longitude positions along the sphere, so the user can move "up" the sphere by moving forward in the Z-axis of the bridge and around the sphere by moving along the X-axis. Lines are drawn between neighboring nodes along the column j dimension, which is drawn as sliced circles of the sphere as seen in figure 4, image 1.

The instantaneous energy density or "height" of each each node is treated as if it were a literal height level. The radius of each node from the center of the sphere is multiplied by this value, causing waves to bounce up and down as they ripple outward. The absolute value of these energy levels determines the opacity of the line drawn at each node, with an absolute height of zero (no energy) corresponding to complete transparency. This helps to visually highlight interference patterns within the system as energy is distributed and dissipates throughout the network. When there is limited or no user input, and therefore little energy within the system, the simulated form fades to transparency and all that can be seen is the black background in "acceleration mode" or a single stationary energy peak for each glove in "fixed mode." Conversely, when a user makes quick, large movements the entire field lights up in motion. Projecting the toroidal graph around a sphere causes the waves to disperse along its surface connecting at the poles (due to the modulo operator), creating a relatively successful conversion from two dimensions into three.

At certain moments throughout the composition, the resting surface of the spherical mesh is warped by scaling its radius to reflect qualities of the accompanying sonification. An example of this can be seen in figure 4, image 3 and will be described in more detail below.





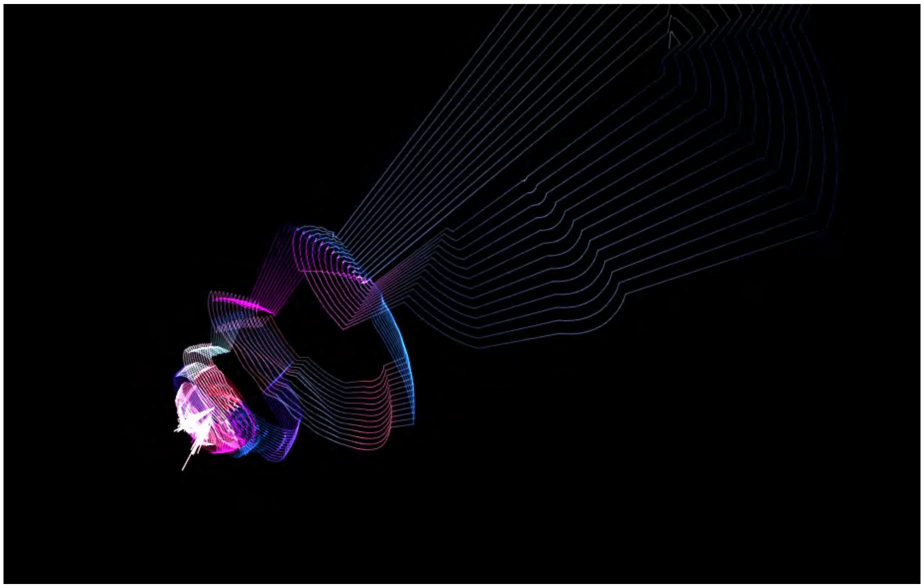


Figure 4: “Standing Waves” screenshots

5 Sonification

The audio portion of the piece has been designed as a novel approach to sonify wave propagation, and to allow for the exploration of sounds in frequency-space and time through physical movement. To create this system, small samples taken from audio files are broken up into their constituent frequencies and mapped onto the physical dimensions of the room. The user can add energy to and excite these frequencies by creating movement around them, translating kinetic energy into sound. Each tone is spatialized in three-dimensions around the AlloSphere’s dome, further emphasizing the illusion of being "inside" of a sound wave.

The process of generating these frequency “landscapes,” starts with pre-selected sound files that are divided into spectral frames of 65536 samples, corresponding to roughly 1.5 seconds of single channel audio sampled at 44100hz. Each spectral frame is processed through a short-time Fourier transform (STFT) function using four overlapping Hanning windows. This deconstructs the sample into an array of frequency and amplitude bins. The bins are sorted from largest to smallest in terms of amplitude, and the frequencies of the 54 highest amplitude bins are calculated and used to tune a sine wave oscillator assigned as its partner. A diagram of this system can be seen in figure 5, where white dots represent wave-field nodes and red dots represent oscillators distributed throughout the graph.

The oscillators are then imagined as being spread out in rows and columns along the original wave graph and likewise along the X and Z axes of the AlloSphere's bridge; The amplitude of each oscillator is calculated by taking the absolute value of the wave field corresponding to its assigned location at each moment. When the area of the graph that an oscillator rests on is flat, then it will be completely silent. When this part of the graph is stirred up, the oscillator will sound. This connection allows users to hear energy distribution throughout the system as waves move back and forth exciting different members of the oscillator array. Put simply, the user is able to control sonic activity by moving his or her hands throughout the space in the same method as that used to control the visual components of the piece.

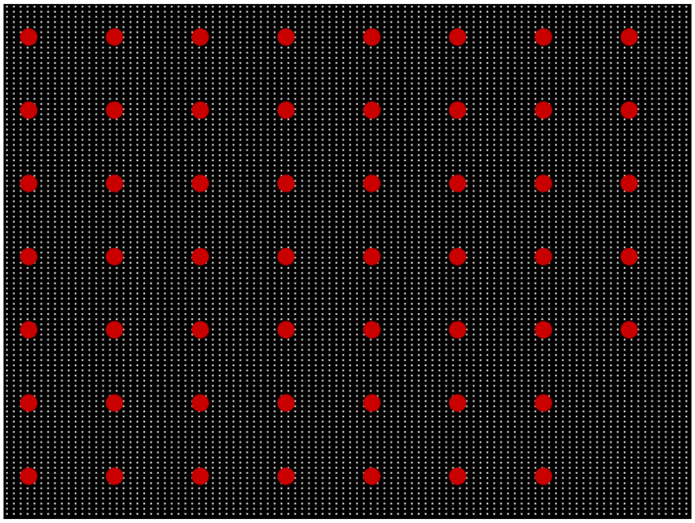


Figure 5: Fifty-four oscillators positioned along the wave field graph.

The spectral makeup of each moment in the audio file is approximated through the use of 54 sine tone oscillators, which can continue playing for an arbitrary length of time. Remaining fixed on a single frame in this fashion is essentially a "spectral freeze," allowing the frequencies that make up a sound to be paused and explored at length:

> "We may select a window (or sets of windows) and freeze either the frequency data or the loudness data which we find there over

> the ensuing signal (spectral freezing). If the frequency data is held constant, the channel amplitudes (loudnesses) continue to vary as in the original signal but the channel frequencies do not change. If the amplitude data is held constant then the channel frequencies continue to vary as in the original signal" [wishart]

This is an example of the first model, where the frequency data is held constant and the amplitude values are changed, however the amplitudes are controlled by the wave propagation system as opposed to the audio file itself, which is why this is not spectral freeze in its traditional definition. Additionally, only the 54 highest amplitude frequencies are used, which while able to convey a sense of the original audio, are far from an accurate recreation. It's worth noting that the sample rate of the wave system runs on the visual timescale at roughly 30 FPS, so amplitude values which are sent to the oscillators 30 times a second are really used as a target which is linearly interpolated to over 5 milliseconds at audio rate.

In the third movement, spectral frame analysis progresses along a fixed tempo, but in all other sections of the piece, the user must manually move forward in time through the audio file by creating sudden spikes in acceleration. This movement in turn changes the contents of the frozen spectral frame and the frequencies being played.

5.1 Density plot synthesis

The result of this mapping is a variation of additive synthesis, where many primitive waves are summed together to approximate more complex tones, and vector synthesis, where 4 or more sound sources are laid out on a vector plane and the amplitude of each source is determined by the X and Y position of a controller inside of it. Here the vector plane is replaced with a 2D density plot reflecting the amount of energy density at each point in the graph as calculated by the wave equation, controlling the amplitude of 54 sources spread throughout it. The same method could be applied to many different types of sound sources and equations for generating energy levels in N-dimensions. Therefore I propose the term “density plot synthesis” to describe the generalized technique of mapping sound source amplitude values on a grid while an equation producing a density plot is calculated on top of it.

The formula, given an arbitrary univariate dataset or simulation in N-dimensions is as follows:

1. Choose a type of sound sources such as primitive oscillators, grain generators, etc.
2. Assign each sound source to a location in the data by spacing the group evenly throughout the same number of dimensions as the input data source.
3. Tune each sound source to a frequency determined either stochastically, from an external input, or from a second characteristic of the system if the input dataset has more than one variable (multi-variate).
4. Change the amplitude of each sound source in relation to the density or value of the data field at its location as the equation is run or time slices of the data move forward.
5. Optionally spatialize each source to reflect its assigned location.

For example: the density plot synthesis scheme implemented in "Standing Waves" uses sine-tone oscillators as sound sources, which are tuned through sound files (external input), and the amplitudes are controlled by the (univariate) wave equation. This formula could be used to sonify an assortment of related equations: Buddhabrot maps, heat maps, reaction-diffusion systems, pixel histograms, etc.

5.2 Spatialization

Each oscillator is assigned to and one of the 54 separate speakers present in the AlloSphere, providing a simple but effective method of spatialization. Oscillators in close proximity on the wave graph are generally output through neighboring speakers, however the conversion from a 2D grid to a three ring speaker array prevents a direct spatial mapping. None the less, an intriguing sense of depth is created as sound peaks literally move throughout the room. A diagram of the speaker rings can be seen in figure 6, where oscillators 1-12 are projected through the bottom speaker ring, oscillators 13-41 are projected through the middle speaker ring, and 42-54 are projected through the top ring.

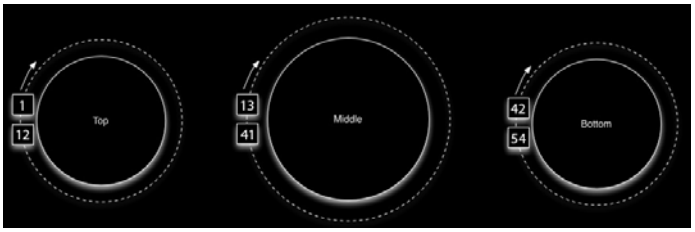


Figure 6: Three ring speaker array inside of the AlloSphere and oscillator assignments.

5.3 Resonance

Because of the resonant damping inherent to the system, a distinct reverb effect is heard as energy dissipates throughout the room. This epiphenomenon is an important characteristic of the sonification, pointing perhaps to its effectiveness at translating the qualities of the underlying equation into sound. A decaying 2D wave equation network is closely related to damped vibrating plates commonly used to simulate reverb, so the reverberant nature of its sonification makes sense intuitively. There are also many less expected qualities contained in the sonification. For instance, rhythmic patterns at varying rates can sometimes be heard as waves constructively and deconstructively interfere, creating spikes and valleys in the oscillator's amplitude envelopes.

6 Audio-visual interaction

The visual and sonic components of this piece play into and affect one another both in the composition and design process as well as in real time as the piece is performed. For instance, the choice to determine oscillator amplitudes by pretending that the oscillators were "buoyed" in a graph throughout the wave field was realized while prototyping different visualization possibilities and trying to develop a method to convey this dissipation of energy along the field. This amplitude-tethering metaphor became the basis of density plot synthesis as it is described here. Likewise, once the sonic portion of the piece was included it became apparent that line segments would be an interesting method to render the visual waves, due to the fact that they resembled audio waveforms commonly seen in music editing software and other visualizers.

6.0.1 Cohen's Color Selection Algorithm

The visualization is also directly affected by the contents of the corresponding sonification during every spectral frame. The color hue of each section of the waveform is determined by the 54 chosen frequency values, with lower frequencies corresponding to blue and higher to red. To determine this hue, each of the 10,000 nodes uses the frequency of its nearest assigned oscillator; Specifically, the row and column indices of the node graph are downsampled to the row and column indices of the 54 oscillator bank and the normalized hue of each node is then set as the frequency of that oscillator divided by 2000.0 and constrained between 0.0 and 1.0 (where 0.0 is red and higher values move forward through the spectrum). This mapping only provides frequency differentiation between 0.0 and 2000.0 Hz, however in practice this was where a majority of spectral variation appeared.

Color saturations and lightnesses (values) were chosen using a modified version of Harold Cohen's coloring formula, as used extensively in his AARON algorithmic painting system[16]. This algorithms can be summarized as follows: three normalized number ranges are chosen, corresponding to low, medium, and high values. As an example, Cohen suggests 0.15 to 0.35 for the low (L) range, 0.4 to 0.65 for the medium (M) range, and 0.8 to 1.0 for the high (H) range. These are then set up in a 3x3 matrix, each corresponding to a possible saturation-lightness pairing. For example, a low-low (LL) pairing would provide both saturation and lightness values chosen randomly from within the low range (0.15 - 0.35), whereas a HL pairing would provide a saturation value chosen from the high range (0.8 - 1.0) and a lightness value chosen from the low one.

During initialization, a probability value is assigned to each of the 9 pairing possibilities inside of this matrix. These probabilities determine the statistical frequency of each saturation-lightness pair, and therefore have a profound effect on the algorithm's final appearance. Cohen suggests only using 2-3 of these pairings throughout a single composition, so an example of a typical probability map would be 20%-LL, 40%-HL, and 40%-MH. Whenever a new color is desired, the algorithm selects one of these ranges based on its assigned probability, and then a specific saturation-lightness pair is chosen randomly from within each of the selected pair's ranges. An example of the results can be seen in figure 7.

In "Standing Waves," saturation-lightness values generated through this technique are used in conjunction with hues determined by the oscillator frequencies to color the waveform. Classes for generating colors through Cohen's algorithm were written in C++ in preparation for their inclusion into the AlloCore synthesis library.

Figure 7: Harold Cohen's color selection algorithm at work

In movements II and V, the amplitudes of these frequency bins also affect the overall shape of the spherical mesh over which the 2D field is wrapped (before the wave heights are added). The same quantization method used to color each region of the sphere by first dividing it into 54 sections is used to multiply its radius by the amplitude of each oscillator's assigned spectral amplitude bin. An amplitude value of 0.0 would multiply this sections radius by 0.0, causing it to rest in the absolute center of the sphere, whereas higher amplitude values push it outward. This system, combined with the section's frequency-dependent coloring, enable viewers to get a sense of its spectral makeup and the ratio of magnitudes between its 54 most prominent component tones. This can be seen in figure 8.

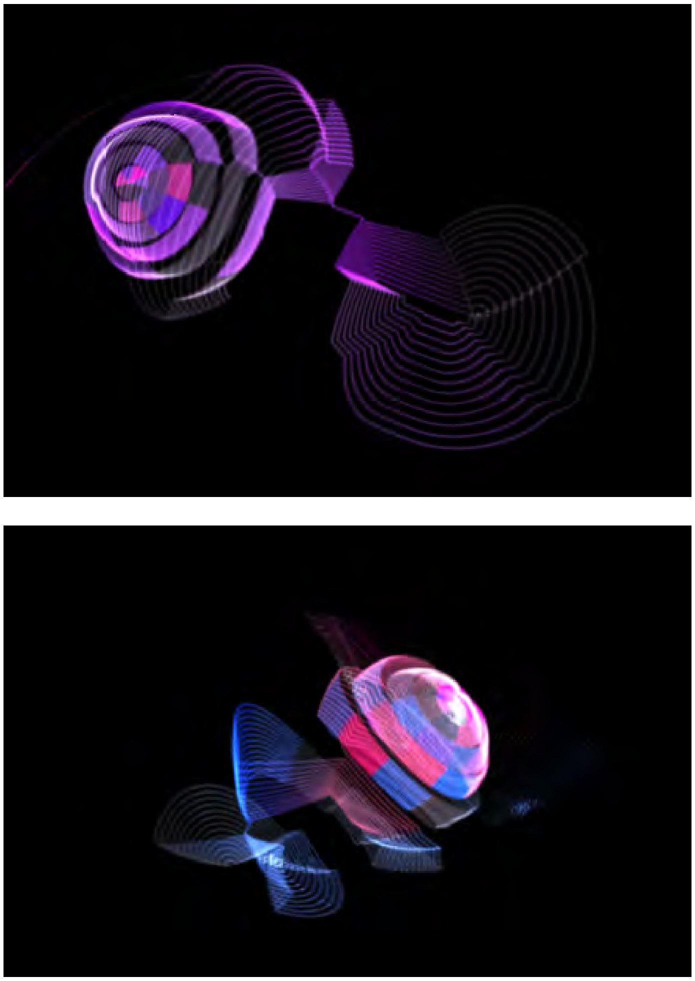


Figure 8: STFT bin amplitude values affecting the shape of the sphere

7 Interaction

Marker-based motion capture systems, such as the Phasespace system used for this piece, are uniquely suited for the task of joining physical and virtual worlds because of their origin as real movement through real space. They are especially appealing as a control interface because of their inherently flexible range of movement, high bandwidth, and low-latency. They can track objects at sub-millimeter accuracy and stream this data at over 100 Hz, however the application of this information is only as effective as the mapping system used to translate it into parameters of control. The difficulty then, is how to map kinetic information into intuitive and effective methods of interaction.

> *The success or failure of a live computer music instrument is determined by the way it maps performers' control gestures to sound. We believe that the best mappings are informed by metaphors... [**?**]*

In this piece, the metaphor of "dragging" was used in order to mimic the familiar action of dragging one's finger or hand through liquid, causing ripples. Here the "liquid" is the energy field, and the ripples extend outward in waves of sight and sound. The user can create larger energy peaks by using fast, exaggerated movements, or smaller waves by standing almost still and only slowly extending his or her hand. Setting up the interaction in this way is meant to make the controller feel as intuitive as possible. A first time user should immediately notice that stillness results in a diminished amount of motion within the system and movement has the opposite effect, stirring up the projected form.

The use of the Y or "gravity" axis, brings up an interesting point regarding Allocentric vs. Egocentric interfaces. Allocentric interfaces define all position and orientation information relative to to the room's center as defined during calibration of the system. An *egocentric* system on the other hand, defines spatial information relative to the user's current Pose (position and orientation). The Y-axis is an interesting example of where the two models blend together. The axis itself is described in allocentric terms according to the orientation of the room, however because humans have an inherent gravitational sense, movement along this axis can almost be thought of as egocentric, since the up/down orientation of the room will mirror the up/down orientation of the user generally.

> *Allocentric: Relating to spatial representations: linked to a reference frame based on the external environment and independent of*

> *one's current location in it. For example, giving the direction as "north," as opposed to "right" (egocentric).*

Mapping strategies which utilize the relative distance between the set of gloves can also be thought of in Egocentric terms. Although it is measured allocentrically, the distance between the two is the same in any coordinate system. In practice it seems that the most successful gestural mapping strategies employ aspects of both allocentric and egocentric models. The former allow the user to change the piece by moving through the room, while the latter provide instant and perhaps more intuitive feedback.

8 Form

8.1 Audio Specified Movements

Transitions between sound files signify "movements" within the piece. The unique frequency signatures of the underlying audio have a profound affect on the spectral composition of the synthesized audio, and therefore serve as a very simple formal structure within the work. The sound files were chosen for their spectral "signatures," each corresponding to a desired affect.

8.1.1 Movement I

The first movement begins with the virtual camera placed directly in the center of the sphere. Complete darkness opens up into lush harmonies of sound and color using the song "Verbena Tea With Rebekah Raff" by Teebs as the driving audio file. As the movement progresses, the camera slowly pans outward as its movement boundary is expanded, allowing the audience to see the sphere's structure from outside. The user's hand acceleration data controls the height of the wave peaks introduced to the system, so quick movements make large spikes and vice versa.

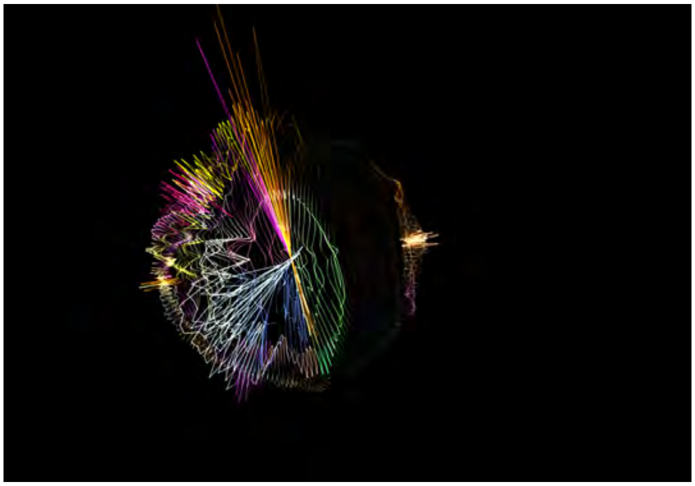


Figure 9: Movement I seen from outside the sphere

8.1.2 Movement II

During the second movement the sphere is warped according to the amplitudes of each spectral bin, creating an amplitude map of the input signal. Opacity values are not allowed to fall to zero, guaranteeing that the structure of the sphere can be observed even when no energy is propagating through it. Visually, the virtual camera is placed toward the center, creating a sort of "tunnel" effect. Glove acceleration is used to trigger changes in rotation.

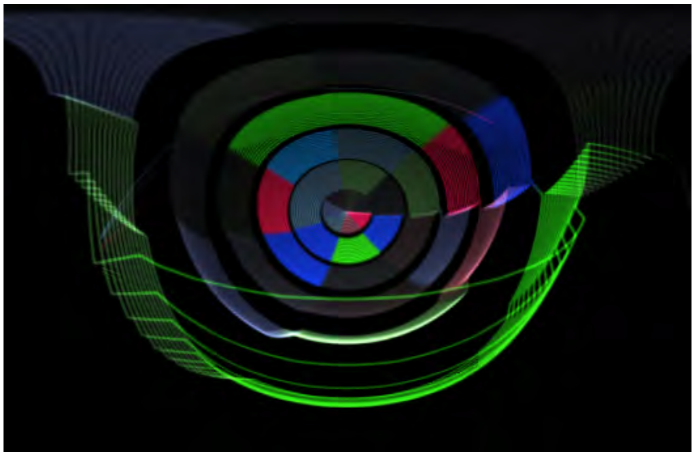


Figure 10: Tunnel Visualization

8.1.3 Movement III

The third movement creates a steady, even tempo as the progression of spectral frames is controlled by a fixed timer. The audio file used in this section is a recording of the fifth movement of Gustav Holst's "The Planets" suite "Saturn: the Bringer of Old Age." Saturn, or Chronos, the god associated with the passage of time seemed like an apt subject for this tempo-driven portion of the piece. During this section, the user's position directly affects the placement of the virtual camera, so he/she can navigate by changing positions within the room.

8.1.4 Movement IV

During the fourth movement, energy peaks are quantized into discrete impulses triggered by large acceleration spikes. This effectively creates small "drops" in the wave field whenever the user makes a quick movement. Because no energy is generated when the user remains still, this mode has the potential to be much quieter and subtle than the others. If the user allows the waves to die down, the entire space will fade to darkness and silence. Resonance is set to high, to emphasize the reverberant nature of the system. "Neptune," also from Holst's "The Planets" suite, is used as an audio source.

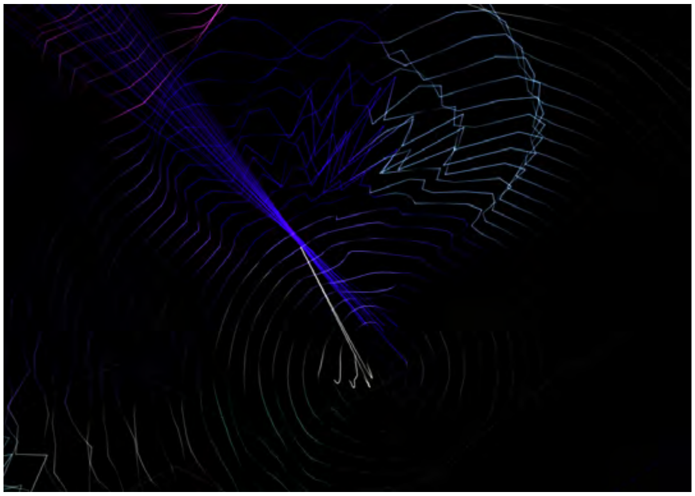


Figure 11: A blue impulse triggered in movement IV

8.1.5 Movement V

Visualization is a main focus of the fifth movement. The sphere is warped by the amplitudes of the spectral bins as in the second section, however the camera is not constrained to the sphere's center, so the audience is able to get a better view of the overall shape. The virtual camera is controlled directly and a constant energy peak is created at the location of each hand. The audio sample used is a clip from "Moonlight Becomes You" by Booker Little, 1958.

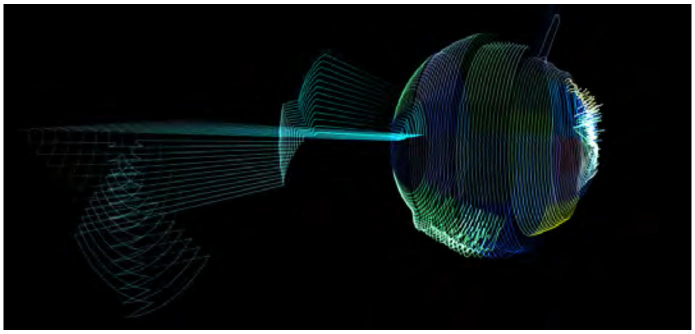


Figure 12: Movement V

8.1.6 Transitions

The transition between movements is triggered by keeping track of the running "amount" of user input, calculated by taking the distance traveled over the last few moments. When the amount of input drops significantly after a long period of action, a flag is triggered to allow for the cycle of movements. The amount of time since the last cycle is checked and if this falls below the minimum time set for the current movement, it ends and the next one is triggered.

8.2 Autonomy

While the piece is built to be interacted with by users, it is somewhat autonomous and can run as a self-propelled installation if no user input is detected. This was meant to allow for long pauses when gloves were being switched from one user to another, or if no one in the audience wanted the responsibility of taking control. In this case, the movements will still cycle through in succession normally. All energy is manually added to the system in "drops" and streams that are stochastically generated. While in fully autonomous mode, the amount of activity is increased and decreased according to rules determined by each movement. For instance, the fourth, “quiet” movement only features small periodic drops of energy, while the third movement uses regular pulses in order to keep time with the tempo.

The camera used to navigate the virtual world is always somewhat self-driven, although it is programmed to mimic and react to the users’ input if present. Its speed reflects the general level of recent activity input and it tries to “follow” acceleration peaks by choosing new targets or increasing spin during moments of heightened input. These animations are created by choosing various target locations around the center of the environment and then interpolating toward that location at each frame. Some of these locations are pre-determined, such as the center of the sphere and a few other “characteristic” viewpoints. Others are chosen at random based upon boundary constraints. The camera is programmed to perform various “tricks,” such as flying through the center of the sphere, placing it at the very top of the dome, or causing it to spin slowly around the AlloSphere’s “equator.”

9 Technical Aspects

9.1 State Synchronization

This piece was created almost entirely in C++ using the AlloCore multimedia framework[17]. The system which enabled the piece to be projected seamlessly around the top hemisphere of the AlloSphere requires the use of 5 separate computers running identical programs in parallel. In order to ensure proper synchronization, large amounts of data must be distributed to each computer at the frame rate of the piece: specifically, height and color values for each node of the wave field, which need to be shared as they are calculated. Normally, this problem is solved through the use of OSC messages sent from one machine designated as the “master” to the 4 other systems. OSC messages over UDP, however, are constrained to a theoretical limit of 64kB and a practical limit much lower than this, therefore requiring the use of multiple messages to send larger amounts of data such as were necessary in Standing Waves.

The ØMQ framework was used to overcome this limitation, enabling a single, roughly 200kB message to be sent and received during each frame. This packet delivers a large C++ struct containing all of the necessary values for maintaining synchronization of the piece between each machine. The method by which these messages can be sent and received were wrapped into a set of convenience functions and classes in hope that they will be useful for others while creating similar large-scale multimodal projects in the AlloSphere and elsewhere. As content becomes increasingly complex, it will require the transmission of larger magnitudes of data.

10 Related Work

A number of previous motion-capture-based studies informed the creation of “Standing Waves.” These were mostly carried out in the transLAB, a 6-by-5 meter room neighboring the AlloSphere, equipped with 4 projectors and a 16-channel sound system. The Optitrack-brand motion capture interface installed within the lab is similar to the one in the AlloSphere, except that during these studies passive markers configured to create a single rigid-body point were used instead of active-marker gloves. This reduces the data down to a single Pose for each controller, which can be used much like the Phasespace Glove’s centroid.

10.1 Controller Design

After much experimentation, a set of three custom controllers were created from small, thin pieces of balsa-wood, glued into simple geometric configurations. These are roughly the size of an outstretched hand and weigh a couple of ounces. These controllers were developed while trying to improve on the design of the plastic ones included with the default Optitrack system, which proved to be very uncomfortable during extended use. The Optitrack controller forces the user to keep a small, tight grip which seems to cause fatigue more quickly than the large, loose grip encouraged by the custom layout. It is also extremely light, weighing less than an ounce, which seems to actually reduce the feeling of control and “presence.”

The final set of three were colored red, blue, and yellow respectively in order to be identified, as can be seen in figure 13. This configuration greatly aided the processes of calibration and parameter scaling because the same colors were used in all of the software, providing an instant link between physical and virtual worlds. They were also helpful for defining and explaining separate functionality to participants, since unlike Gloves, they are not easily identified as “right” and “left.”

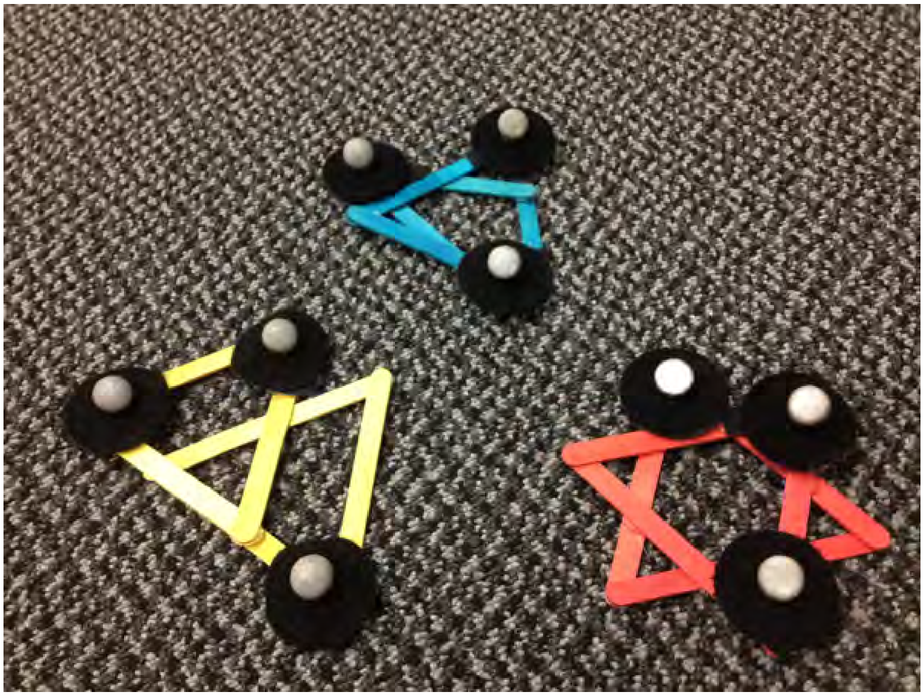


Figure 13: Three controllers.

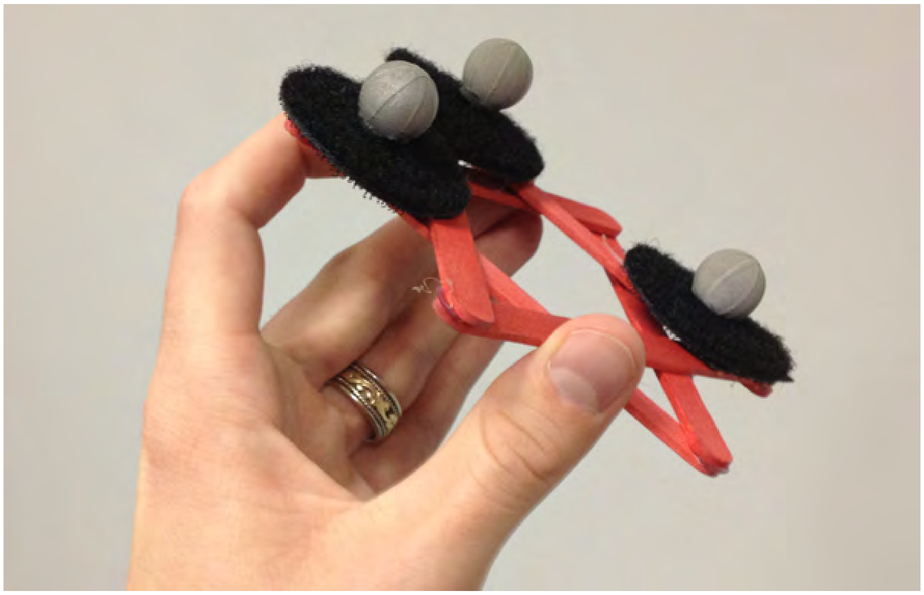
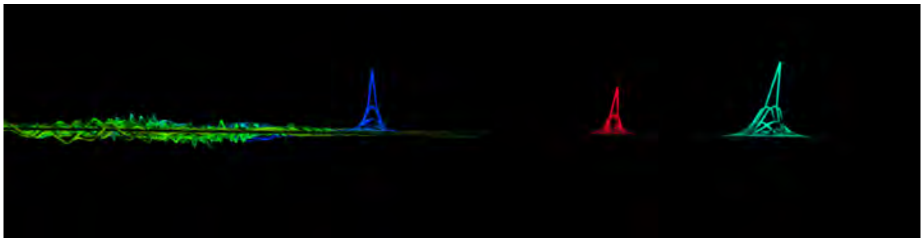


Figure 14: Red controller held in hand.

10.2 Standing Waves Prototype

The original version of "Standing Waves" was a multi-user installation in the transLAB built in *Processing* using the *glGraphics, PeasyCam,* and *minim* libraries. Although it uses the same principle of visualizing and sonifying energy spikes along a 2D wave field as its AlloSphere counterpart, the mapping and visualization model are much simpler. The three red, yellow, and blue controllers are placed on the floor of the space, and up to three users can control the X and Z location of the energy peaks along the system. These are projected as spikes along a flat plane as opposed to wrapped around the surface of a 3D sphere. The piece is also static in the fact that it does not cycle through movements, making it more of an installation than a composition.

The control interface utilizes the most simple mapping scheme mentioned in this document, however it remains one of the most intuitive and effective because of its direct and easily understood spatial connection. Three controllers were placed on the ground of the tracking portion of the transLAB, each corresponding to one energy peak. When a viewer first enters the installation, assuming no one else has been interacting with it, he or she observes only 3 randomly spaced spikes of color, which look like frozen or "standing" waves. As soon as one of the controllers is moved, energy is propagated through the system and entire field becomes visible, allowing its overall form to be discerned.



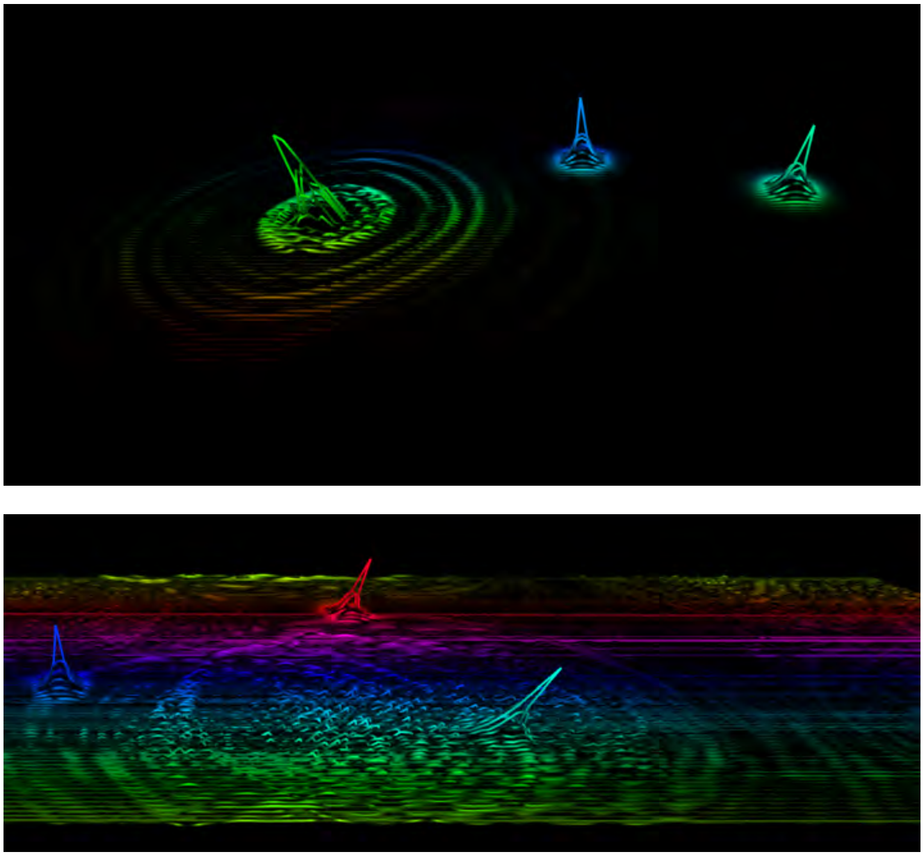


Figure 15: "Standing Waves" installation screenshots.

The sonification was generated through a field of 20 sine wave oscillators, spaced evenly along the wave field in a grid in a similar fashion as the AlloSphere version of the piece. The oscillators were tuned stochastically, however, as opposed to using frequencies selected from a sound file. They are tuned in multiples of 30 Hz, randomly selected up to the 30th harmonic (900 Hz) when the program is initialized. Using harmonic multiples in this fashion, guarantees a tonal center despite randomization. Capping values to a lower maximum harmonic caused the output to sound very consonant and boring, while allowing much higher ratios made it sound overly dissonant and unstructured. The 30th harmonic was chosen as an empirically pleasing compromise.

This success of the installation seems to have been the result of clear and simple metaphors used for the audio, visuals, and control mappings. The visual waves themselves for instance, actually mimic what appears while observing a spectrogram analysis of the installation's audio. When all three controllers are still, 3 tall visual waves correspond to 3 tall spikes seen in the spectrogram, corresponding to amplitude peaks in each sine wave. When energy is added to the system and it starts to oscillate with more visual complexly, similar patterns appear on the spectrogram because of the more chaotic distribution of energy between each sine wave. An example of this can be seen in figure 16.

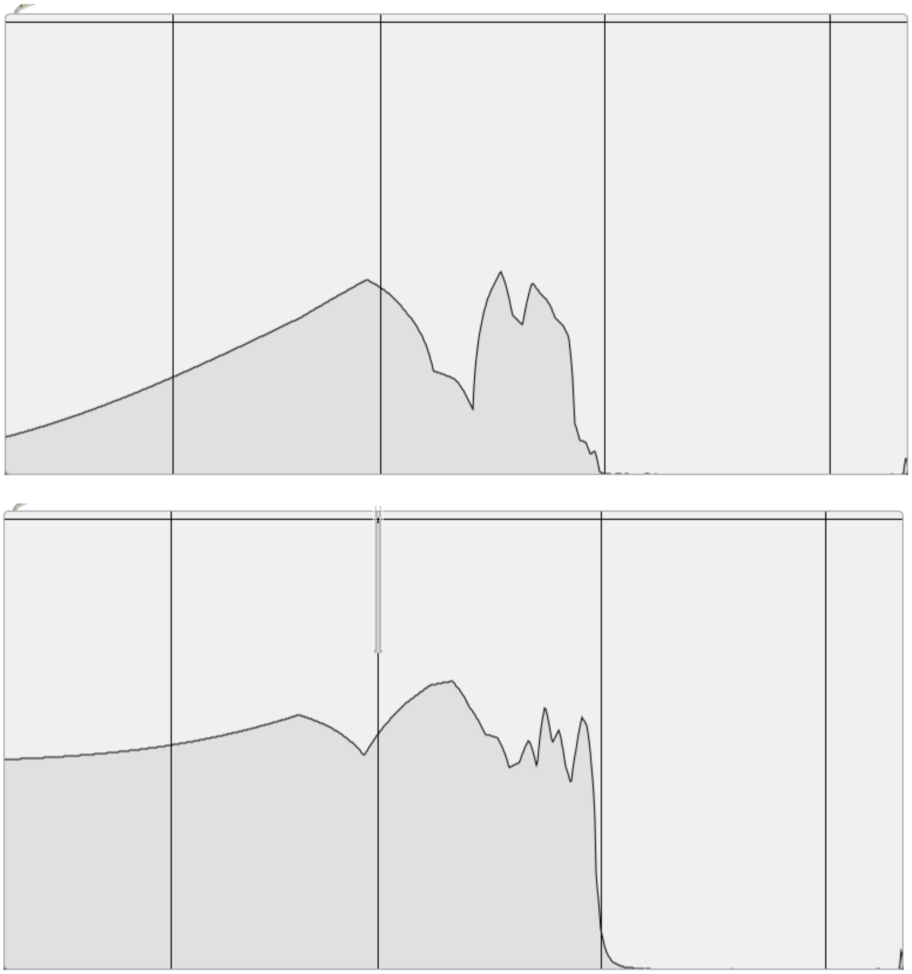


Figure 16: spectral analysis of the audio, first with no interaction showing three distinct spikes, second with interaction showing energy spread throughout the spectrum.

In terms of control metaphors, because each physical dimension is directly mapped to its corresponding virtual dimension, participants observe an instant one-to-one correlation between their actions and the response on screen and over the speakers. The participants physically “drag” waves through the virtual pool, a connection which seems to have created a very intuitive and unified experience.

10.3 Pantograph

“Pantograph” is a tape piece composed entirely through motion-capture-controlled granular synthesis. Inspired through the study of microsound techniques, the interface used to create the piece was developed in order to control sample playback speed through in real-time through gesture. These studies inspired the idea of “exploring” the frequencies of a sound frozen in time, and therefore played a large role in the creation of “Standing Waves.” “Pantograph” was created in winter 2012, and was played on April 29th on the (fortuitously named) *Kinetics Radio Broadcast.*

The piece is named “Pantograph” after the mechanical construct used to translate and transform direct motion, often in the context of shrinking or enlarging drawn forms. I find this to be an apt metaphor for the way in which motion capture synthesis was used to translate movement into sound.

> *A pantograph (from Greek roots πaντ- ’all, every’ and γρaφ- ’to write’, from their original use for copying writing) is a mechanical linkage connected in a manner based on parallelograms so that the movement of one pen, in tracing an image, produces identical movements in a second pen. If a line drawing is traced by the first point, an identical, enlarged, or miniaturized copy will be drawn by a pen fixed to the other.*
>
> *Because of their effectiveness at translating motion in a controlled fashion, pantographs have come to be used as a type of motion guide for objects large and small.*

It was composed in a two-part process: sounds were first synthesized in a series of real-time “performances” through a custom motion-capture interface and then organized into a final composition offline. A small amount of post-synthesis effects were added to the final piece, but by-and-large the composition is nothing more than an organized collection of sounds generated in real-time through this instrument.

Figure 17: A pantograph.

The motion-capture interface was designed inside of Max/MSP using the Granular Toolkit (GTK) by Nathan Wolek. Grain streams were created through cascaded sets of the grain.pulse~ object, a unit which produces a sequence of grains sampled from a user specified input buffer and filtered through a customizable window. After loading sample and window buffers, 3 parameters can be specified dynamically: "sample start position", which sets the grain playback location within the sampled buffer, "grain length", and "pitch multiplier", which controls the playback speed of the sampled waveform.

It was determined that the "richest" configuration was formed using three cascades, each comprised of a chain of four grain.pulse~ generators. All three chains were configured to read from the same buffer using the same grain window (Gaussian), however their dynamic parameters (start position, length, playback speed) would be determined by movement in only one of the three axes. Separating them by axis seemed logical and enabled the performer to focus on a single stream by limiting movement to a single dimension or (more often) to create complex inter-stream parameter relationships through more complex multi-dimensional gestures.

10.3.1 Interaction

Two rigid-body controllers were used at once (one in each hand), meaning that parameters could not only be mapped to the position vectors of each controller individually, but also their distances relative to one another. Despite using only two points instead of an entire glove, the mapping system used in Pantograph is more complicated and less intuitive than the one used in the Standing Waves Prototype because of the extended simultaneous use of room-stationary and inter-controller position vectors. Nonetheless, such a potentially complex interface facilitated many interesting experiments and pleasing results once the learning curve had been overcome.

10.3.2 Synthesis Modes

Because of the many mapping, scaling, and configuration choices available to a motion-capture granular synthesizer, a seemingly limitless number of combinations can be made, each of which will have distinct sonic characteristics when performed. “Pantograph” makes use of three such configuration possibilities, named here for their sonic affect:

10.3.3 Drone-like Mode

This employs the most simple mapping strategy and outputs only a single grain stream. The sample start location of this output stream is determined by the (room stationary) x-dimensional position of the first controller, the grain length is determined by the y-distance between the two controllers, and the frequency of the train~ object triggering the stream is inversely proportional to their z-distance.

In practice, the first controller was held by the non-dominant hand and was moved forward and backward along the x-axis while otherwise remaining stationary. The performer generally aligned himself directly along the x-axis so that large movements in time could be created by walking forward or backward respectively, with fine control tuned by hand. The second controller was held by the dominant hand and would move over the y and z axis relative to its partner, changing the frequency and length of the grain stream. Interestingly, this somewhat mimics the separation of role and axis of movement for each hand experienced while playing the Theremin.

This mode is an extreme example of an "opt-out" style interface, because grain amplitude is always set to full. The only way that silence can be output is if the performer navigates over a silent portion of the waveform. Stillness by the performer merely results in a parameter freeze, which causes the grain streams to create a looped "drone-like" output, hence its name.

Sample Location	x1
Grain Length	abs(y1 - y2)
Train Frequency	abs(z1 - z2)

10.3.4 Phrased Mode

This style is much more subtly expressive because all three streams are utilized, each being controlled by one of the three axes (x, y, z). Velocity along each axis is directly coupled to the amplitude of its corresponding stream, so the performer can "mute" each of them individually by freezing movement in one direction. This is an "opt-in" (fading) interface because stillness will always result in amplitude values of zero and therefore complete silence.

Grain waveforms for all three streams were always taken from the same larger sample, however they were each sampled at different instances and tuned to different playback rates, so they are essentially made up of different waveforms. Nevertheless, since they were drawing from samples containing the same pitches and their fundamental frequencies were tuned in harmonic octave ratios (0.5, 1.0, 4.0), they maintain a certain pitch-space unity. They sound separate but related to one another. These are best described as *polychrome clouds*, or granular synthesis clouds containing two or more waveforms[12].

The specific mappings in this mode are as follows: X-axis controls stream 1, Y-axis controls stream 2, and Z-axis controls stream 3. The absolute velocity in each of these axes is mapped to the amplitude of its respective stream. The distance between the two controllers in each of these axes sets the length of the grains.

Stream 1 Amplitude	abs(x'1)
Stream 2 Amplitude	abs(y'1)
Stream 3 Amplitude	abs(z'1)
Stream 1 Sample Location	x2
Stream 2 Sample Location	y2
Stream 3 Sample Location	z2
Stream 1 Grain Length	abs(x1-x2)
Stream 2 Grain Length	abs(y1-y2)
Stream 3 Grain Length	abs(z1-z2)

The subtle interplay of the two hands is very important in this mode and produces quite interesting results. Exact control over the three streams individually is generally too difficult to keep track of, however gestural "guidelines" are quickly learned and used to perform series of phrases: for example, extending one's arms makes the samples longer and less overlapped, whereas placing hands closer together makes small, very quick grains. In any situation you can cause an "acceleration" to occur by moving your hands toward one another or vice-versa for deceleration. Touching your hands together and then moving them apart in all but one axis will cause a sort of rhythmic counterpoint where two streams expand while one remains fixed. The use of inter-hand expansion and contraction to seems to be a valuable parameter in any two-handed interface such as this, because out hands seem to have an intuitive ability to find one another in this way whether or not our concentration is elsewhere.

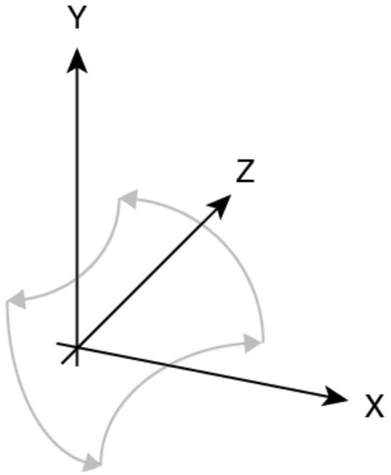


Figure 18: Complex motion along multiple axes, creates complicated volume envelopes and grain-length shifting.

10.3.5 Hissing Mode

The "hissing" mode is essentially an extension of the phrased mode where the playback speed of each stream is multiplied by a factor of the absolute distance between each controller in each axis, giving it a strong vocal quality, possibly the result of intentionally extreme aliasing. Absolute distance on the x-axis set the playback speed of the first stream, distance on the y-axis set the playback speed of the second stream, and distance on the z-axis set the playback speed of the third stream. Values are scaled so that a distance of 0.0 to 1.0 meters would correspond to playback speeds of 0 to 100 for stream 1, 0 to 200 for stream two, and 0 to 400 for stream three. These speeds are set in multiples of the original playback rate, so a playback speed of 100 would be 100 times as fast as the original.

The sonic effect of modulating the playback speed in this way is further complicated by the fact that playback length is tethered to the same parameters, so as distance between the controllers increases, the length of the sampled waveform expands almost as a way of compensating for the increased speed. These values are not scaled as to be perfectly complementary however, and therefore this relationship creates non-intuitive and interesting musical results. Changing the distance between each controller resulted in a strong shift in the perceived formant of the streams, almost resembling a "hissing" or breathing sound, that could be controlled with great accuracy.

Stream 1 Playback Speed	abs(x1-x2)
Stream 2 Playback Speed	abs(y1-y2)
Stream 3 Playback Speed	abs(z1-z2)

10.4 Spatialization Studies

Motion capture seem to be an ideal method through which to control the spatialization of audio since it is inherently spatial. Experiments with such interfaces dates back at least as far as 1951 when Pierre Henry and Pierre Schaeffer used a device called "relief desk" in the piece "Symphonie Pour un Homme Seul":

> *...a spatial control system called 'relief desk' was tested. This system was used to control dynamics during the performance (music*

> *was played from several shellac players) and also to create what was called a 'stereophonic' effect, which actually was a left–right control on the position of a monophonic sound. The organisation of the loudspeakers in the hall was quite original too: two loudspeakers were placed at the front right and left sides of the audience; two other loudspeakers completed the distribution – one was place at the rear, in the middle of the hall and another also placed at the rear, but over the audience. The system was controlled from the stage, with the 'relief desk', which consisted of two circular electro-magnets placed perpendicularly – the two hands of the performer moving in and out the circles, or towards left and right and thus controlling the spatial intensity and the localisation of the sounds[2].*

While composing "Pantograph," inspired by the work of Henry and Schaeffer, various attempts were made to create a motion-capture spatialization tool. These studies were conducted through *Processing* and *Sound Element Spatializer (SES)*, a cross platform C++ application written by Ryan McGee, capable of spatializing multichannel audio in real-time[3]. It was hoped to create a spatialization interface capable of controlling multiple sound sources both as groups and independently when desired. With more than one source per controller used, this is no trivial task.

Figure 19: Pierre Henry controlling the "relief desk."

The first model was a gravity-based particle system, where 8 particles were set up corresponding to 8 spatialized sound sources. These would rotate around a central particle, the position of which was determined by the motion-capture controller's 2-dimensional position within the room. This system produced effective spatial trajectories, but did not allow for very direct control. In order to improve this, a second controller was added in order to dynamically alter the amount of gravity within the system. This was scaled to the distance between the two controllers, so when the user would outstretch his arms, gravity would increase and the particle's orbits would speed up. If his hands were put together, gravity would drop to zero and the particles would stop moving. This provided a much greater level of control, but it was still not a satisfactory solution.

The next iteration spatializer mapping used a 3D representation of the room, inside of which the position of 8 sound sources were represented as groups of 4 particles connected to the outside of a disk by what resemble spokes on a wheel. These disks could be moved around the room and rotated directly in a 1-to-1 mapping with each motion capture controller. In one iteration, the controller's rotation speed would determine the size of each disk, causing sounds to spread out from one another as if from centrifugal motion. This interface seems more promising than the particle-system approach, but it still does not enable the user to spatialize sound sources individually. Perhaps a more elegant solution is yet to be discovered.

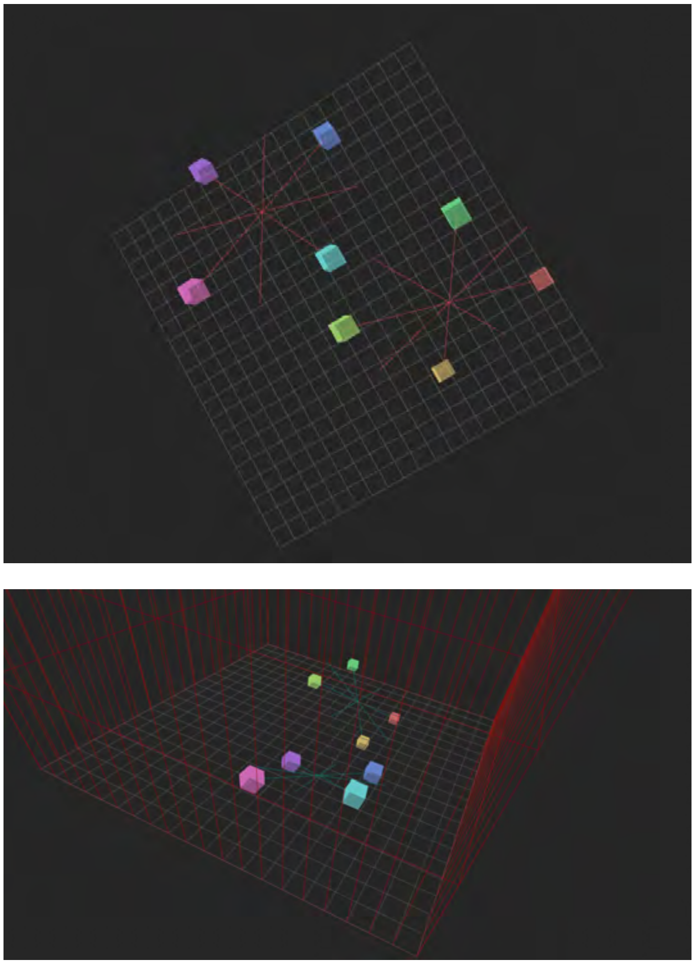


Figure 20: Spatialization interface.

The spatialization method implemented for the final version of *Pantograph* employs a much simpler solution. Using the transLAB's 8-channel setup, each grain stream was configured to output to a different pair of speakers corresponding to one of the walls. For instance the first stream was output on speakers 1&2 along the left wall, stream 2 was output through speakers 3&4 along the front wall, stream 3 was output through speakers 5&6 along the right wall, and all three streams were mixed and output through speakers 7&8 along the back wall. With this setup, as the performer would shift movement from axis to axis, sound would be synthesized in different sections of the room corresponding to the streams with most amplitude. While simplistic, this mapping produced a strong and aesthetically pleasing spatial effect. It was this model of non-literal-position spatialization that informed the methods later used in "Standing Waves."

"Pantograph" was originally performed and recorded in 8-channels using this method. During the composition process it was reduced to 4 channels and a stereo version is also available[13].

11 Conclusion

"Standing Waves" is the first composition in the AlloSphere designed specifically for use with its glove-based motion-capture interface, and is therefore an important contribution to the project of creating "fully immersive" audio-visual environments. It is one of a small but growing number of pieces created for the spherical constraints and possibilities of this unique instrument. Such a novel multimodal performance space seems to necessitate the development of appropriate tools with which to control it. Motion-capture seems uniquely well-equipped for the task because of its ability to move arbitrarily in relation to any point in the sphere, and because of the possibility of anchoring motion in the virtual world to motion in the physical one. It is hoped that demonstrating the techniques though which the interfaces described here were developed will aid in the creation of a standardized set of methods and tools for creating similar systems in the future.

Solutions to various technical issues specific to the AlloSphere were devised throughout the course of this project. Most notably, the use of ØMQ to pass large amounts of state information around the network is the first simple and generic solution to this problem available in the context of AlloCore. Hopefully these will prove invaluable as more pieces utilize the its full projection capabilities, placing increased importance on inter-computer state synchronization. Similar methods were created for accessing and synchronizing the Phasespace tracking system among each machine, which is especially critical when trying to manage low-latency, high-bandwidth controllers.

In realizing the piece, new techniques were developed for sonifying the wave equation. The result is a unique blend of additive and vector synthesis, used here to enable the component frequencies of a single moment of a sound file. A complementary method of visualizing spectral freeze and modified additive synthesis was created using the wave equation as conceptual and aesthetic inspiration. The underlying principle of distributing sound sources over an N-dimensional grid and controlling their amplitudes through various energy distribution equations, could be extended to a wide range of material, and therefore seems to be an area of great potential for future research. I propose the term “density plot synthesis” for these techniques. Since it is inherently audiovisual, density plot synthesis is an ideal candidate for the task of the “perceptualization” of abstract equations into sight and sound.

References

[1] N. Wolek, “Granular Toolkit,” http://www.lowkeydigitalstudio.com/2007/03/granular-toolkit-v1-49/.

[2] D. Teruggi, “Technology and musique concrete: the technical developments of the Groupe de Recherches Musicales and their implication in musical composition,” *Organised Sound*, 2007.

[3] R. McGee and M. Wright, “Sound Element Spatializer,” *International Computer Music Conference*, 2011.

[4] Optitrack. http://www.naturalpoint.com/optitrack/products/v100-r2/.

[5] G. B. Guerra-Filhol, “Optical Motion Capture: Theory and Implementation,” *Revista de Informática Teórica e Aplicada*, 2005.

[6] Kinetics Radio, http://www.kineticsradio.com/.

[7] D. Wessel and M. Wright, “Problems and Prospects for Intimate Musical Control of Computers,” Comput. Music J. 26, 3 (September 2002), 11-22.

[8] D. Wessel, M. Wright, and J. Schott, “Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors,” International Conference on New Interfaces for Musical Expression (NIME).

[9] J. C. Schacher and A. Stoecklin, “Traces – Body, Motion and Sound,” International Conference on New Interfaces for Musical Expression (NIME).

[10] The AlloSphere Research Facility, http://www.AlloSphere.ucsb.edu/.

[11] “True RMS Meter,” *music-dsp mailing list*, http://music.columbia.edu/pipermail/music-dsp/2007-June/066273.html, June 2007.

[12] C. Roads, “Microsound,” MIT Press, 2004.

[13] “Pantograph,” http://www.amusesmile.com/?p=140, 2012.

[14] http://mediatechnology.leiden.edu/programme/curriculum/perceptualization/.

[15] R. Lala, “Quintilian: A Framework for Intuitive Interaction in Immersive Environments,” 2012.

[16] H. Cohen, “Colour, Simply,” 2006.

[17] AlloCore C++ Multimedia Framework, https://github.com/AlloSphere-Research-Group.

[18] ZeroMQ, http://www.zeromq.org/.

[19] L. Sevgi, “A Review of Discrete Solutions of Poisson, Laplace, and Wave Equations,” *IEEE Antennas and Propagation Magazine, Vol. 50, No. 1*, 2008.