Experiencing Audio and Music in a Fully Immersive Environment

Xavier Amatriain, Jorge Castellanos, Tobias Höllerer, JoAnn Kuchera-Morin, Stephen T. Pope, Graham Wakefield, and Will Wolcott

UC Santa Barbara

Abstract. The UCSB Allosphere is a 3-story-high spherical instrument in which virtual environments and performances can be experienced in full immersion. The space is now being equipped with high-resolution active stereo projectors, a 3D sound system with several hundred speakers, and with tracking and interaction mechanisms.

The Allosphere is at the same time *multimodal, multimedia, multi-user, immersive*, and *interactive*. This novel and unique instrument will be used for research into scientific visualization/auralization and data exploration, and as a research environment for behavioral and cognitive scientists. It will also serve as a research and performance space for artists exploring new forms of art. In particular, the Allosphere has been carefully designed to allow for immersive music and aural applications.

In this paper, we give an overview of the instrument, focusing on the audio subsystem. We give the rationale behind some of the design decisions and explain the different techniques employed in making the Allosphere a truly general-purpose immersive audiovisual lab and stage. Finally, we present first results and our experiences in developing and using the Allosphere in several prototype projects.

1 Introduction

The Allosphere is a novel environment that will allow for synthesis, manipulation, exploration and analysis of large-scale data sets providing multi-user immersive interactive interfaces for research into immersive audio, scientific visualization, numerical simulations, visual and aural data mining, knowledge discovery, systems integration, human perception, and last but not least, artistic expression.

The space enables research in which art and science contribute equally. It serves as an advanced research instrument in two overlapping senses. Scientifically, it is an instrument for gaining insight and developing bodily intuition about environments into which the body cannot venture: abstract, higher-dimensional information spaces, the worlds of the very small or very large, the very fast or very slow, from nanotechnology to theoretical physics, from proteomics to cosmology, from new materials to new media. Artistically, the Allosphere is an instrument for the creation and performance of new avant-garde works and the development of new modes and genres of expression and forms of immersion-based entertainment, fusing future art, architecture, science, music, media, games, and cinema.

R. Kronland-Martinet, S. Ystad, and K. Jensen (Eds.): CMMR 2007, LNCS 4969, pp. 380–400, 2008.

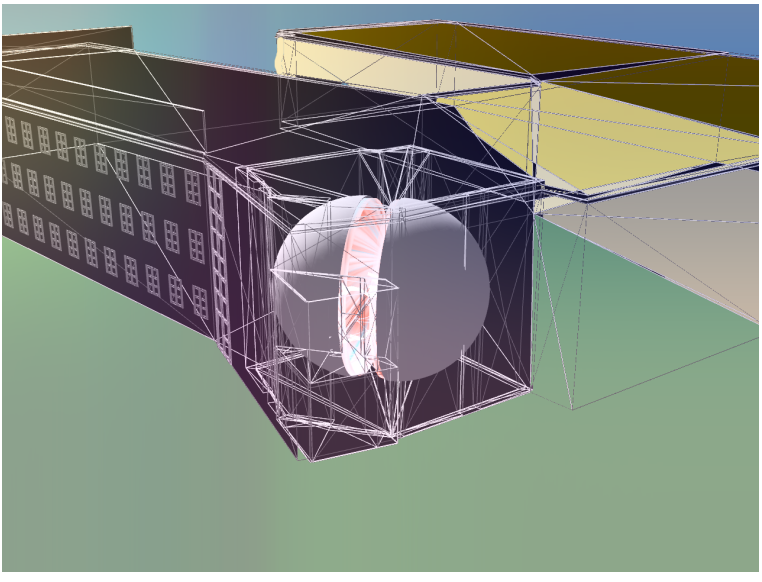

Fig. 1. A virtual rendering of the Allosphere

The Allosphere is situated at one corner of the California Nanosystems Institute building at the University of California Santa Barbara (see virtual model in Figure 1), surrounded by a number of associated labs for visual/audio computing, robotics and distributed systems, interactive visualization, world modeling, and media post-production. The main presentation space consists of a three-story near-to-anechoic room containing a custom-built close-to-spherical screen, ten meters in diameter (see Figure 3). The sphere environment integrates visual, sonic, sensory, and interactive components. Once fully equipped, the Allosphere will be one of the largest immersive instruments in the world. It provides a truly 3D 4 π steradians surround-projection space for visual and aural data and accommodates up to 30 people on a bridge suspended in the middle of the instrument.

The space surrounding the spherical screen is close to cubical, with an extra control/machine room in the outside corner, pointed to by the bridge structure. The whole outer space is treated with sound absorption material (4-foot wedges on almost all inner surfaces), forming a quasi-anechoic chamber of large proportions. Mounted inside this chamber are two 5-meter-radius hemispheres, constructed of perforated aluminum that are designed to be optically opaque (with low optical scatter) and acoustically transparent. Figure 4 is a detailed drawing showing a horizontal slice through the Allosphere at bridge height. The two hemispheres are connected above the bridge, forming a completely surround-view screen.

We are equipping the instrument with 14 high-resolution video projectors mounted around the seam between the two hemispheres, projecting onto the entire inner surface. A loudspeaker array is placed behind the aluminum screen, suspended from the steel infrastructure in rings of varying density (See speaker in the bottom left corner in Figure 2).

Fig. 2. Looking into the Allosphere from just outside the entrance

The Allosphere represents in many senses a step beyond already existing virtual environments such as the CAVE [9], even in their more recent "fully immersive" reincarnations [15], especially regarding its size, shape, the number of people it can accommodate, and its potential for multimedia immersion. In this paper, we focus on a particular aspect of the multimedia infrastructure, the audio subsystem.

Although the space is not fully equipped at this point, we have been experimenting and prototyping with a range of equipment, system configurations, and applications that pose varying requirements. We envision the instrument as an open framework that is in constant evolution, with major releases signaling major increments in functionality.

2 A Truly Multimedia/Multimodal System

An important aspect of the Allosphere is its focus on multimedia processing, as it combines state-of-the-art techniques both on virtual audio and visual data spatialization. There is extensive evidence of how combined audio-visual information can influence and support information understanding [19]. Nevertheless, most existing immersive environments focus on presenting visual data. The Allosphere is a completely interactive multimodal data mining environment with state-of-the-art audio and music capabilities [28].

Figure 5 illustrates the main subsystems and components in the Allosphere, as well as their interactions. The diagram is a simplified view of the integrated multi-modal/media system design. The exact interactions among the various media data (visual, aural, and interactive) are dependent on the particular individual applications to be hosted.

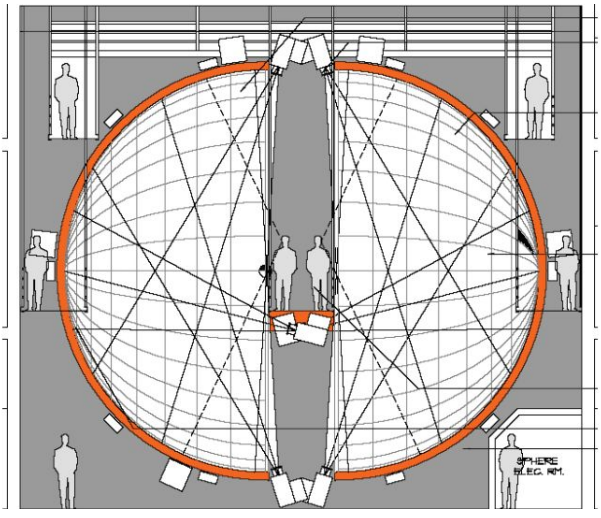

Fig. 3. The Allosphere

The remainder of this section will briefly introduce each of those components and subsystems, as well as the way they interact. In Section 3 we will then discuss the audio subsystem (*Allo.A*) in detail.

The main requirements for the Allosphere visual subsystem (*Allo.V*) are fixed both by the building and screen characteristics and the final image quality targeted [12]. The sphere screen area is 320.0 m^2 and its reflective gain, FOV averaged, is 0.12. The Allosphere projection system (*Allo.V.D.P*) requires image warping and blending to create the illusion of a seamless image from multiple projectors. We have designed a projection system consisting of 14 3-chip DLP active stereo projectors with 3000 lumens output and SXGA+ resolution (1400x1050) each. The projectors are being installed with an effective projector overlap/blending loss coefficient of 1.7.

A typical multi-modal application in the Allosphere will integrate several distributed components, sharing a LAN:

- back-end processing (data/content accessing)
- output media mapping (visualization and/or sonification)
- A/V rendering and projection management.
- input sensing, including real-time vision and camera tracking (related to *Allo.V.V*), real-time audio capture and tracking (related to *Allo.A.C*), a sensor network including different kind of regular wireless sensors as well as other presence and activity detectors (related to *Allo.SN*).
- gesture recognition/control mapping
- interface to a remote (scientific, numerical, simulation, data mining) application

It follows from our specification requirements – and our experiments have confirmed this view – that off-the-shelf computing and interface solutions are insufficient to power the sphere. Allosphere applications not only require a server cluster dedicated to video

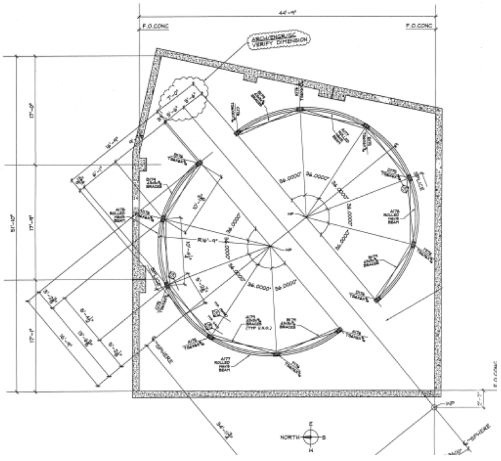

Fig. 4. Horizontal section of the Allosphere

and audio rendering and processing, but also a low-latency interconnection fabric so that data can be processed on multiple computers (in a variety of topologies) in real time, an integration middleware, and an application server that can control the system in a flexible and efficient way.

The computation infrastructure will consist of a network of distributed computational nodes. Communication between processes will be accomplished using standards such as MPI. The Allosphere Network (*Allo.NW*) will have to host not only this kind of standard/low-bandwidth message passing but also multichannel multimedia streaming. The suitability of Gigabit Ethernet or Myrinet regarding bandwidth and latency is still under discussion. In our first prototypes, Gigabit has proved sufficient, but our projections show that it will become a bottleneck for the complete system, especially when using a distributed rendering solution to stream highly dynamic visual applications. We are considering custom hardware technologies as a possible necessity in the future.

3 The Audio Subsystem

The Allosphere is designed to provide "sense-limited" resolution in both the audio and visual domains. This means that the spatial resolution for the audio output must allow us to place virtual sound sources at arbitrary points in space with convincing synthesis of the spatial audio cues used in psychoacoustical localization. Complementary to this, the system must allow us to simulate the acoustics of measured or simulated spaces with a high degree of accuracy.

In a later stage we also plan to complement the audio subsystem with a microphone array in order to arrive at fully immersive audio [25]. However, this component is still at the very early stages of design and will therefore not be discussed in this section.

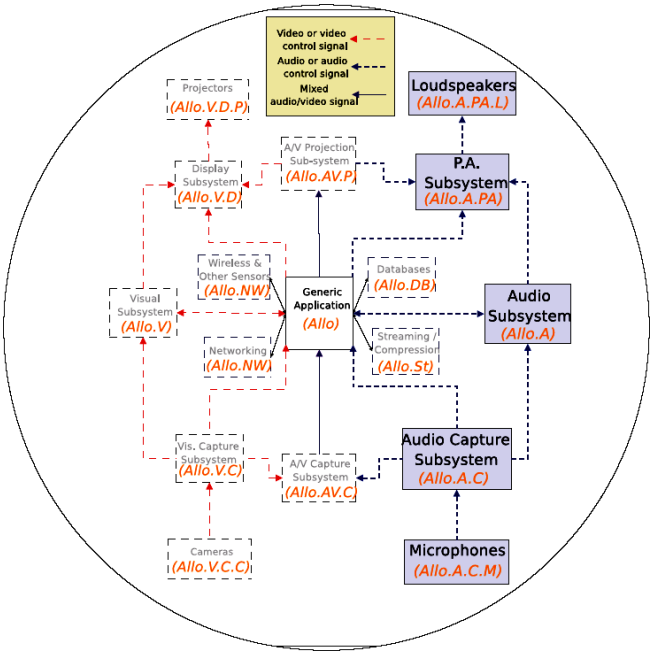

Fig. 5. The Allosphere Components with a highlighted Audio Subsystem

3.1 Acoustical Requirements

In order to provide "ear-limited" dynamic, frequency, and spatial extent and resolution, we require the system to be able to reproduce in excess of 100 dB sound pressure level near the center of the sphere, to have acceptable low- and high-frequency extension (-3 dB points below 80 Hz and above 15 kHz). We designed the spatial resolution to be on the order of 3 degrees in the horizontal plane (i.e., 120 channels), and 10 degrees in elevation. To provide high-fidelity playback, we require audiophile-grade audio distribution formats and amplification, so that the effective signal-to-noise ratio exceeds 80 dB, with a useful dynamic range of more than 90 dB.

To be useful for data sonification [4] and as a music performance space, the decay time (the "T60 time") of the Allosphere was specified to be less than 0.75 seconds from 100 Hz to 10 kHz [6]. This is primarily an architectural feature related to the properties of the sound absorbing treatment in the quasi-anechoic chamber, which was designed to minimize the effect of the aluminum projection screen. The perforations on the screen have also been designed to minimize its effect across most of the audible spectrum. Initial experiments confirm that the absorption requirements have indeed been met.

3.2 Speaker System

It has been a major project to derive the optimal speaker placements and speaker density function for use with mixed-technology many-channel spatialization software

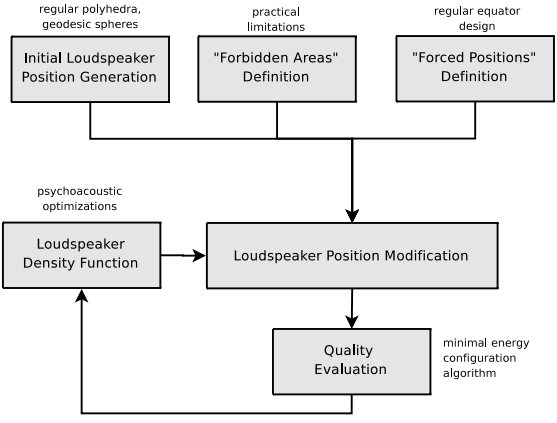

Fig. 6. Allosphere speaker placement iterative design method and variables

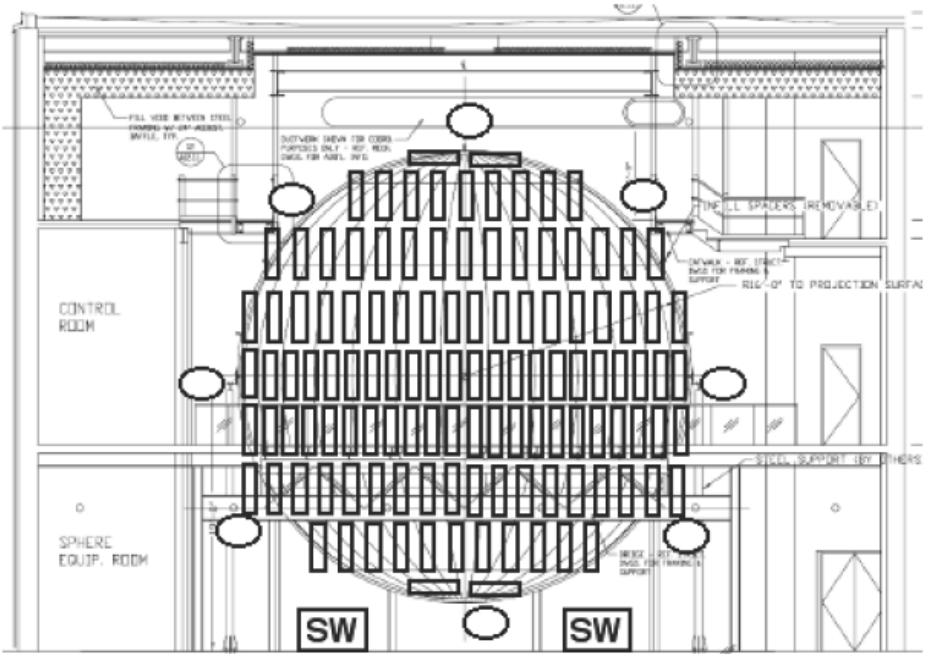

Fig. 7. Allosphere speaker placement design, initial results after first iterations taking into account VBAP requirements

(see discussion and calculations in [13], summarized in figure 6). Our driver placement design comprises between 425 and 500 speakers arranged in several rings around the upper and lower hemispheres, with accommodations at the "seams" between the

desired equal and symmetrical spacing and the requirements of the support structure. The loudspeakers will be mounted behind the screen.

We have projected densely packed circular rings of speaker drivers running just above and below the equator (on the order of 100-150 channels side-by-side), and 2-3 smaller and lower-density rings concentrically above and below the equator. The main loudspeakers have limited low-frequency extension, in the range of (down to) 200-300 Hz. To project frequencies below this, four large sub-woofer(s) are mounted on the underside of the bridge.

At this moment, because of timing and construction constraints, we have installed a prototype system with only 16 full range speakers installed along the three different rings mentioned above and two subwoofers under the bridge. Those speakers are connected to the computer via Firewire audio interfaces that support 32 channels.

For the imminent growth of the prototype into the full system, we plan to switch to passive speaker elements wired to a set of 8-16 networked digital-to-analog converter (DAC) amplifier boxes, each of which supports in the range of 32-128 channels and has a Firewire interface. As an alternative, we are also considering building custom interface boxes consisting of a Gigabit Ethernet interface, digital/analog converter, power amplifier, and step-up transformer (this would be based on a design developed at CNMAT for their 120-channel loudspeaker array [10]).

4 Spatial Sound System

Since the Allosphere is to foster the development of integrated software for scientific data sonification and auditory display, as well as artistic applications, it is essential that the software and hardware used for audio synthesis, processing, control, and spatial projection be as flexible and scalable as possible. We require that the audio software libraries support all popular synthesis and processing techniques, that they be easily combined with off-the-shelf audio software written using third-party platforms such as Csound, Max/MSP, and SuperCollider, and that they support flexible control via (at least) the MIDI and Open Sound Control (OSC) protocols. Due to the sophistication of the audio synthesis and processing techniques used in Allosphere applications, and the expected very large number of final output channels, we require that the core audio libraries support easy inter-host streaming of large numbers of channels of high-resolution (24- bit, 96 kHz) audio, probably using both the CSL/RFS and SDIF networked audio protocols.

This section discusses the design of the spatial audio software library developed for the Allosphere. The CREATE Signal Library (CSL) [20] is intended to function as the core library, handling all audio needs of the Allosphere. We have developed a flexible software framework based on the CSL, in which different techniques, sets of psychoacoustical cues, and speaker layouts can be combined and swapped at run time (see Castellanos thesis [7]).

The first step towards this goal was to design a spatial audio library that integrated seamlessly with CSL. To this end, the spatial audio software developed for the Allosphere consists of the implementation of a library written in C++ as part of CSL. By designing this framework, we provided an immediate solution for spatial sound

reproduction in the Allosphere, but also, most importantly, opened the path towards the development of a universal spatial-sound reproduction system. The system aims to be intuitive and easy to operate by those that need a ready-to-use surround sound system, but at the same time sufficiently complex and flexible for the initiated user who may desire to fine-tune the system and or add new configurations and techniques. Such system would ideally include, in one package, all major currently existing spatialization techniques. The next paragraphs give an overview of the framework that was designed to that effect.

4.1 Spatial Audio Framework

The design of the spatial audio framework was driven by the goal of using the Allosphere as a multipurpose environment, equally suitable for scientists and artists for a variety of applications. This goal required flexibility in the systems configuration and interfaces manipulation. Its most current version [7] was designed using the ”Metamodel for Multimedia Processing Systems” (also known as 4mps) proposed by Xavier Amatriain [1]. This metamodel provides a solid ground for a flexible, dynamic and extensible library.

The flexibility vs. simplicity trade-off was solved by using a layered interface, where each layer provides different levels of flexibility, with the trade-off of complexity. Essentially, the system provides different interface layers, where higher-hierarchy layers conceal the complexity of lower (and more flexible layers), while providing more default / standardized options. Thus, a higher degree of complexity and flexibility is available on to those who need it.

The simplest interface conceals from the user all spatialization mechanisms, not offering the option of choosing any distance cues or the spatialization technique for sound reproduction. The user is responsible only with determining the desired location of the sound source in a 3D coordinate space, and providing the audio material. The framework will handle everything else, including the encoding/decoding technique and the loudspeaker configuration. At the other end, by using the lowest layer, the user can determine the distance cues, filters and spatialization algorithm. It is also possible to perform changes dynamically at run-time.

The most complex configuration of the framework is created around the concept of a Spatializer. A spatializer constitutes a processor capable of manipulating a stream of audio with its output appearing to originate at a particular location in a virtual/simulated space. A spatializer is composed of various processing units, such as distance filters, panners and a layout of the loudspeaker setup. Ideally, the Spatializer would simplify the spatial audio reproduction by loading the most appropriate ”panner” (vbap, ambisonic, etc.) based on the audio setup description (loudspeaker layout). This technique would eventually appear to the user as a single spatialization engine that performs satisfactorily under any circumstances. When more flexibility is needed, the various components of a spatializer can be used individually creating custom or more complex audio graphs.

The current design does not place any restrictions on the number of loudspeakers to be used and their placement. The limit to the number of loudspeakers to be used in a particular configuration is primarily imposed by the computing resources available.

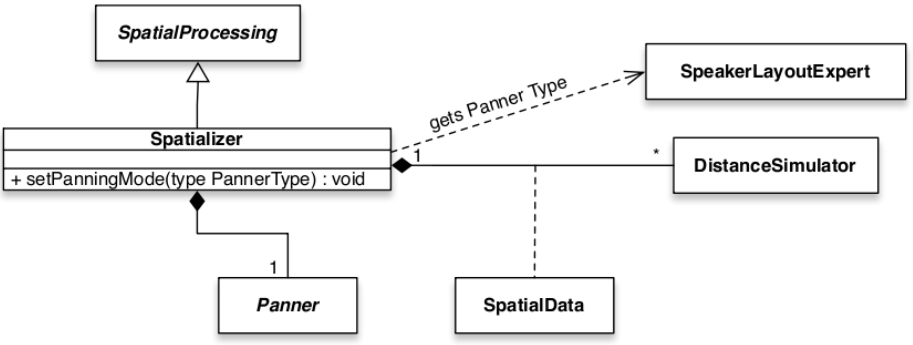

Fig. 8. Spatializer Class Diagram

The loudspeaker layout has to be specified as needed for audio spatialization processing. The framework is designed to load the loudspeaker layout from a text file containing the position of each individual component (loudspeaker). For the more user-friendly configurations, there are several default loudspeaker layouts that can be chosen without the need of manually entering the coordinates of the loudspeakers For instance, a Stereo setup will automatically add two loudspeakers -30 and 30 degrees.

5 Spatial Audio Techniques

There are three main techniques for spatial sound reproduction used in current state-of-the-art systems: (1) vector-base amplitude panning [21], (2) ambisonic representations and processing [17], and (3) wave field synthesis (see [23] and [5]). Each of these techniques provides a different set of advantages and presents unique challenges when scaling up to a large number of speakers and of virtual sources.

In the following paragraphs we outline how we have approached the challenges and opportunities of each of these techniques in the context of the Allosphere project.

5.1 Vector-Base Amplitude Panning

With the Vector base Amplitude Panning technique, a sound can be located in a virtual space by manipulating the balance of the audio signal sent to each speaker. It is assumed that the speakers are equidistant from the listener, and that panning only allows moving the source position along the arc between speakers (i.e., source distance must be simulated independently). The first step is to determine which three speakers define the smallest triangle that includes p (the desired position), and what the contribution of energy from each of these will be to simulate a source at position p. Given a desired source position, one can apply an arbitrary weighting function to derive the factors for the output channels given the position of the vector for the loudspeaker triplet L (see equation 1).

$$gains = p^T L_{mnk}^{-1} = [\, p_1 \; p_2 \; p_3 \,] \begin{bmatrix} l_{k1} & l_{k2} & l_{k3} \\ l_{m1} & l_{m2} & l_{m3} \\ l_{n1} & l_{n2} & l_{n3} \end{bmatrix}^{-1} \tag{1}$$

Practical VBAP systems allow interactive performance with multiple moving sound sources, which are mapped and played back over medium-scale projection systems. VBAP has been mainly promulgated by groups in Finland and France and is used effectively in 8-32-channel CAVE virtual environments.

The drawbacks of VBAP are that it does not directly answer the question of how to handle distance cues (relatively easy to solve for distant sources and low Doppler shift), and that it provides no spatialization model for simulating sound sources inside the sphere of loudspeakers. This is a grave problem for our applications, but also a worthy topic for our research. The question boils down to how to spread a source over more than 3 speakers without limiting the source position to the edges of the surface described by the chosen set of speakers.

The VBAP algorithm involves a search among the geometrical representations of the speakers defining the playback configuration, and then some simple matrix math to calculate the relative gains of each of the three chosen speakers. There are several open-source implementations of VBAP that support multiple sources (with some interactive control over their positions), and flexible speaker configurations involving up to 32 channels.

Members of our research group implemented a system in which the user can move and direct a number of independent sound sources using a data glove input device, and play back sound files or streaming sound sources through VBAP, using a variable number of loudspeakers specified in a dynamic configuration file (see McCoy thesis [18]). VBAP can be integrated with a spatial reverberator, allowing early reflections from a reverberator to be individually panned, though this gets computationally expensive with many sources, complex room simulations, or rapid source (or listener) motion.

Because VBAP is so simple, most implementations are monolithic, 1-piece packages. This is obviously unacceptable for our purposes, so we needed to consider both (1) how the VBAP system scales to large numbers of sources, rapid source motion, and many output channels, and (2) how such a scaled-up application can best be distributed to a peer-to-peer server topology streaming data over a high-speed LAN. The scalability of VBAP encoding software is excellent, since the block-by-block processing is very simple, and the computation of new output weights for new or moving sources can be accelerated using well-understood geometrical search techniques. For the case of many sources or rapid source or listener motion, VBAP scales linearly, because each source is encoded into 3 channels, meaning that many mappers each write 3 channels into a many-channel output buffer. Alternatively, if the servers are distributed, each mapper sends 3 channels over the LAN to its output server. If the output servers are themselves distributed (each taking over a subset of the sphere's surface), then most encoding servers will stream to a single output server.

Computational distribution of a VBAP-based spatial reverberator is more difficult, since by definition the individual reflections are not localized to a small number of channels; indeed, if you calculate a reasonable number of reflections (e.g., 64 or more) for a complex room model, you can assume that the reflections will approximate an

even distribution among all channels, leading us back to a monolithic output server topology. We look forward to attacking this scalability and partitioning issue in the full system. For the time being, we run the reverberator on a single server.

The assumptions of the speaker elements and system configuration for playing VBAP are that elements be identical full-range speakers, and that they be placed in triangles of more-or-less equal size in all directions. The speaker density can be made a function of height, however, leading to somewhat poorer spatialization accuracy above (and possibly below) the listener. All that being said, since VBAP makes so few assumptions about the constructed wave, it supports non-uniform speaker distributions quite well. Directional weighting functions to compensate for an uneven distribution of speakers can be built into the VBAP amplitude matrix calculations, and the fidelity of spatial impression is a directional function of both the speaker density and regularity of spacing. In our earliest designs for the sphere, we ran a set of programs to tessellate spherical surfaces, leading to the 80-channel configuration shown in Figure 7. Note the two regular rings above and below the equator; one can rotate the upper hemisphere by 1/2 the side length to form a zigzag pattern here (which handles VBAP better) Continuing this process, we can design and evaluate further regular subdivisions of a sphere.

5.2 Ambisonics

Ambisonics [11] is a technique to re-create the impression of (or synthesize) a spatial sound-field via a two-part process of encoding recorded or virtual spatial sources into an Ambisonic domain representation, and then decoding this representation onto an array of spatially located loudspeakers. The Ambisonic domain is a multi-channel representation of spatial sound fields based upon cylindrical (2-D spatialization) or spherical (3-D spatialization) harmonics. First-order Ambisonics, also known as the B-Format, encode sound-fields as an omni-directional signal (named W) plus three additional difference signals for each of the axes X, Y and Z. Higher Order Ambisonics (HOA) increases the detail of directional information and expanding the acceptable listening area of the decoded spatial sound field by using higher ordered cylindrical/spherical harmonic orders, and thus increasing the number of encoded signals [17]. The number of Ambisonic domain channels depends only on the order and dimensionality of the representation chosen, and is somewhat independent of the number of sources and of the number of loudspeakers (it is required that the speakers outnumber the domain channels).

To implement an Ambisonic encoder, one generates an encoding matrix based upon the virtual source orientation (azimuth, elevation) relative to the center-spot and uses this matrix to mix a scaled copy of the input signal into each of the Ambisonic domain channels. An Ambisonic encoder does not need any information about the speaker layout. The encoding matrix must be recalculated whenever the center to source orientation changes. A decoder pre-calculates a decoding matrix of weights per Ambisonic domain channel for each of the loudspeakers in the array, again using cylindrical or spherical coordinates and harmonics. The decoder uses these weights to mix each received Ambisonic domain channel to each speaker, and thus is essentially a static NxM matrix mixer.

One of the main benefits of the Ambisonic representation is that it scales very well for large numbers of moving sources. Since the encoding and decoding operations are

linear and time-invariant, as many sources as needed can be encoded into the same Ambisonic domain channels, and encoder matrix recalculation can occur at less than sample-rate resolution (and be interpolated).

Ambisonic encoders and decoders can also therefore be decoupled from one another. For a simple scaled-up system, multiple 3rd-order encoders would run on machines in our server farm, each of them streaming a 16-channel signal to the output driver(s). These signal buses can be summed and then distributed to one or more output decoders. The decoding scales well to large numbers of speakers because decoders are independent of one another, each receiving the same set of inputs. CPU limits can therefore be circumvented by adding more encoding and/or decoding nodes. The scalability to higher orders is well understood, and scales with the number of channels required by the representation, bounded by LAN bandwidth.

Ambisonic decoders work best with a regular and symmetrical loudspeaker configuration. There is no way in the standard algorithms to compensate for irregular speaker placement, though this is an area for future research, along with spatial radiation patterns and near-field encoding. What is interesting is the fact that very large speaker arrays can especially benefit from higher-order ambisonic processing, using ever-higher orders of spherical harmonics to encode the sound field, and then decoding it using these factors to play out over a (regular and symmetrical) many-channel speaker array with a very large acceptable listening area.

As with VBAP, graduate researchers from our group (see [13])) have implemented higher- (up to 11th-) order ambisonic processing and decoding in C++ using the CSL framework. The encoder and decoder are separate classes, and utility classes exist for processing (e.g., rotating the axes of) Ambisonic-encoded sound. We also implemented the algorithm for Max/MSP [27], and there are also open-source implementations in both SuperCollider and PD.

Using Ambisonics for Navigable Immersive Environments. Adapting Ambisonics for navigable virtual environments presents a number of challenges. Ambisonics models spatial orientation well [17], but does not inherently model distance. We have extended our implementation to incorporate multiple distance cues for point sources using standard techniques (amplitude attenuation, medium absorption/near-field filtering, Doppler shift and global reverberation mix [8]). Additionally, we implemented a rudimentary radiation pattern simulation by filtering spatial sources according to the orientation of the source relative to the listener. A more realistic approach to radiation pattern simulation can be found in [16]. This system was used in the AlloBrain project described in section 6.

Though an Ambisonic sound-field in its entirety can be efficiently rotated around three axes using equations based upon spherical harmonics [17], this efficient feature is unfortunately inapplicable to navigable virtual worlds, since any navigation movement changes the spatial orientations on a per-source basis rather than as a group.

Sound source direction in Ambisonics is expressed in terms not immediately appropriate to virtual environments. C++ code was written to efficiently translate absolute positions and quaternion orientations of sound sources and the mobile viewpoint into the appropriate Euler angles of azimuth and elevation for Ambisonic encoding, and the relative distance and angle needed for distance/radiation simulation.

5.3 Wave Field Synthesis

Wave field synthesis (WFS) is an acoustic spatialization technique for creating virtual environments. Taking advantage of the Huygens' principle, wave fronts are simulated with a large array of speakers. Inside a defined listening space, the WFS speaker array reproduces incoming wave fronts emanating from an audio source at a virtual location. Current WFS implementations require off line computation which limits the real-time capabilities for spatialization. Further, no allowances for speaker configurations extending into the third dimension are given in traditional wave field synthesis.

A wave field synthesis system suitable for real-time applications and capable of placing sources at the time of rendering is presented. The rendering process is broken into logical components for a fast and extensible spatializer. Additionally, the WFS renderer conforms to the spatial audio framework designed by Jorge Castellanos [7] and thus fits well in CSL. A broad range of users and setup configurations are considered in the design. The result is a model-based wave field synthesis engine for real-time immersion applications.

WFS Theory and Supporting Work. Wave field synthesis is derived from the Kichroff-Helmholtz integral,

$$P(w,z) = \iint_{dA} G(w,z|z') \frac{\partial}{\partial n} P(w,z') - P(w,z') \frac{\partial}{\partial n} G(w,z|z') dz' \tag{2}$$

which states that the pressure $P(w,z)$ inside an arbitrary volume and due to an incoming wave can be determined if the pressure at the surface of the volume $P(w,z')$ and wave transmission properties $G(w,z|z')$ (free field Green's function) are known. Applying this, the wave field synthesis principle states if the volume surface is lined with speakers, exact acoustic scene reproduction is possible for listeners inside the volume. It should be noted from the integral the volume shape is not defined and the speaker configuration can be irregular. However, there is a disconnect between the outlining theory and practical WFS system.

Berkhout [5] describes three of the assumptions needed for a WFS driving signal at each speaker source. First, the two terms inside the integral of the Kirchoff-Helmholtz integral represent both monopole and dipole sound reproduction sources along the volume surface. Dipoles are an unrealistic expectation in a numerous channel environment. Fortunately, dipole sources can be omitted at the expense of an incorrect sound field outside the speaker-enclosed volume. Given the reproduction room in anechoic, this is a reasonable assumption. In place of the dipoles, a windowing function is applied to the speaker signal allowing sound only when a virtual source is behind the speaker.

Next, the Kirchoff-Helmholtz integral requires a continuous sound reproduction surface which must be discretized for practical speakers. Discretization to monopole point sources results in a 3dB per octave boost to the original signal. The WFS driving signal corrects for this effect with a high-pass filter. Additionally, the spatial sampling of speakers creates spatial aliasing. Unlike the more familiar temporal aliasing, spatial aliasing does not produce as pronounced artifacts, but instead confuses spatialization above the aliasing frequency. The aliasing frequency is proportional to the distance between sources in linear arrays. Aliasing for circular speaker arrays are described by Rabenstein et al. in [24].

Finally, Due to hardware and computational limitations, most WFS designs range contain less than 200 speakers, not enough to surround a listening space. Instead, the dimensionality is reduced from three to two. Ideally, a speaker array is in the plane of the listener's ear.

Based on assumptions and limitations listed above, a driving signal is derived from the Kirchoff-Helmholtz integral by Rabenstein [22]. The driving signal defines the filter computed per each virtual source at each speaker in the array.

$$D_\theta(w,x|x') = 2w(x',\theta)A(\left|x-x'\right|)K(w)e^{j\frac{w}{c}|x',n_\theta|}F(x,\theta) \tag{3}$$

$w(x',\theta)$ is a window function as a result of eliminated dipole speakers and the normal dot product of the incoming wave. $A(|x-x'|)$ is the amplitude attenuation due to distance and the reduction to 2 dimensions. $K(w)$ is a square root of the wave number spectral shaping also due to the dimension reduction. $e^{j\frac{w}{c}|x',n_\theta|}$ applies the appropriate delay to the incoming wave. Finally, $F(x,\theta)$ is the signal emitted from the source.

Model-based wave field synthesis simulates acoustic sources by modeling the properties of its incoming wave to the WFS speaker array. Any arbitrary acoustic shape or radiation pattern is viable for wave field synthesis. However, due to computational complexity, point and plane wave sources are used as audio emitters. Baalman [3] has demonstrated how an arbitrarily shaped WFS system can work.

Interactive Environments. Additional requirements are placed on a wave filed synthesis renderer for use in an interactive environment. Most importantly, rendering must happen at as close to real time as possible. Psyco-accoustical experiments from Wenzel [29], find that audio latency of 250ms presents a perceivable lag when paired with visual or user feedback. Existing WFS implementations such as WONDER [2] or CARROUSO [26] offer real-time processing, but restrict virtual sources to pre-computed positions, panning between points to simulate source movement. Using this method, not only is WFS rendering incorrect between points, but the perceptual cue of Doppler effect inherent in WFS is omitted.

A different method is presented in which WFS filters are calculated in real time per each sample with a small computational overhead. For each buffer, audio is processed by a filter calculated from the sources current position. Stationary virtual source cost no additional cycles and are treated in the traditional way from [22]. For moving sources, an arbitrary source position determines the new filter for the current audio buffer and corresponding source metadata. The source's relative speed is then used to find the Doppler rate corresponding to the buffer's sample rate due to its speed. The result is a sample accurate WFS rendering with buffer rate position updates.

Outer and Focused Point Sources. Another necessity for effective immersion is continuity of the audio scene. WFS allows for virtual sources outside the speaker array and inside (often called 'focused sources'). Rendering virtual sources inside the speaker array is non-causal and different filters must be used. Special consideration is given to the transition from outside to inside filters in order to prevent dead spots or discontinuous audio. The windowing function $w(x',\theta)$, which determines if the source is in front or behind the speaker, is reversed for focused sources. For this reason, knowing a speaker's

and source's positions alone is not sufficient to determine the source's location in the speaker array and which filter should be applied. To overcome this confusion, an expert container class is implemented which models the speaker array shape and informs the rendering chain the correct filter for the virtual source. A third state for virtual sources is near field. When a source is placed at a speaker, the 3db per octave approximation from discretized speakers no longer applies. To accommodate the near-field effect, The spectral filtered audio is mixed with unfiltered audio proportional to is distance to the speaker at small distances.

Separation of Processing. For large scale systems, the WFS rendering process may need to be distributed to keep up with real-time computational requirements. Separating the WFS rendering chain into smaller components allows the distribution of work to multiple machines. Calculation of the auditory scene using the driving signal can be split into two groups, source and speaker calculations. Separation of these processes removes the need for an m (virtual sources) times n (speakers) number of applied filters as suggested in [22]. Additionally, splitting the rendering in this way allows the entire process to be distributed in a way that suits the particular WFS system. If a large number of virtual sources are given, all source-related DSP could take place on multiple machines. Likewise, for a large speaker array, multiple machines could each be synced to handle their own smaller number of speakers.

Finally, an WFS interface is designed to accommodate a range of audio applications and operating systems. The interface must not only allow for the connection of block-rate source audio, but also asynchronous control messages describing the virtual audio source.

CSL and Spatial Audio Interface. The WFS rendering engine is integrated as a spatial audio technique in the spatial audio framework outlined above. Virtual audio sources and positions serve as input to a scene rendered for a certain number of speaker outputs. However, wave field synthesis, while similar, requires additional information compared to VBAP, Ambisonics and other spatial audio techniques.

Due to the physical model of WFS, virtual sources can be easily represented as point or plane sources, a concept unique to WFS. This source attribute would normally accompany its position data. Secondly, a wave field scene rendering requires not only each speaker's position but its normal vector. As a result, the WFS module extends the spatial audio framework to allow for source shapes and extra speaker data. When other spatialization modules, such as VBAP or ambisonic, are used inside CSL, the additional components brought on by WFS are ignored allowing integration between all spatial audio techniques.

6 Testbed Applications

In parallel to the development of the core libraries described above, several tests, prototypes and demonstrations of the Allosphere capabilities have been performed. This section describes the approaches adopted for some of these prototypes.

In the first iteration over the prototype we have set up an environment consisting of the following elements:

* 4 active stereo projectors (Christie Digital Mirage S+2K), 3000 ANSI lumens, DLP
  2 rendering workstations (HP 9400), AMD Opteron 64@2.8Ghz, NVidia Quadro FX-5500
* 1 application manager + Audio Renderer (Mac Pro), Intel Xeon Quad Core @3Ghz
* 2 10-channel firewire audio cards.
* 16 full-range speakers + 2 subwoofers
* Several custom-developed wireless interfaces.

The research projects described below make use of this prototype system to test the functionality and prove the validity of the instrument design.

In the first project, we are developing an immersive and interactive software simulation of nano-scaled devices and structures, with atom-level visualization of those structures implemented on the projection dome of the Allosphere (see Figure 9). When completed, this will allow the user to stand in the middle of a simulation of a nano-scaled device and interact with the atoms and physical variables of that device.

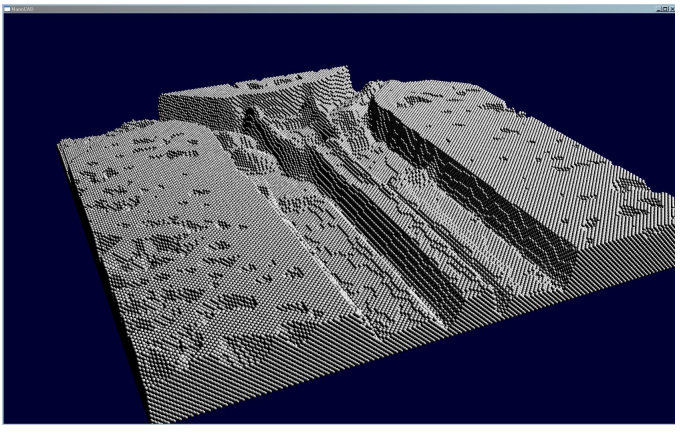

Fig. 9. Rendering of a 1M atom silicon nanostructure in real-time on a single CPU/GPU (Allosphere rendering occurs in stereo projection)

Our science partners are implementing algorithms for nano-material simulations involving molecular dynamics and density functional theory using GPUs, transforming a single PC workstation into a 4 Teraflop supercomputer. This allows us to run nanoscale simulations that are 2-3 orders of magnitude faster than current implementations. We will also be able to use this extra computational power to solve for the physical properties of much larger structures and devices than were previously possible, allowing nano-system engineers to design and simulate devices composed of millions of atoms. Sound design will play an important role in such simulations and visualizations. The sound system will be used to bring important temporal phenomena to user's attention and to pinpoint it precisely with 3D sound. For instance, to alleviate the difficulty of

finding specific molecules in the vast visual space of the Allosphere, subtle auditory clues can alert the user to the emergence or presence of a specific molecular event in a particular direction (which is especially relevant when the object is behind the user's back!).

In another project, we focus on molecular dynamics. We are extending the VMD [14] package through the use of Chromium in order to have seamless visualization of complex protein molecules and their interactions, immersively supported with direct manipulation and spatial sonification by the Allosphere.

6.1 AlloBrain

The last ongoing research project, called AlloBrain, explores brain imaging data as an immersive environment, following a desire to meld data from the sciences with the artistic pursuits of new media art. Our goal is not to interpret the data in a scientific way, but rather to indicate and provoke inspiration regarding immersive three-dimensional media offers in terms of new insights and interaction with data sets from other disciplines. Digital artist and transvergent architect Marcos Novak undertook the fMRI brain scanning and even before the Allosphere building was completed. The AlloBrain project became our driving prototype and experimentation platform.

While the brain data provides an intricate architecture for navigation (see figures 6.1 and 6.1), it does not by itself create a compelling interactive experience. Dynamic elements are added to the world through mobile, agents that indicate their presence spatially, visually, and sonically 11. Their distinct behaviors within the system provide a narrative for the installation based around exploration for features in data-space, and clustering activities around features of interest. The immersant can navigate the space and call specific agents to report the status of their findings using two wireless (Bluetooth) input devices that feature custom electronics, integrating several MEMs sensor technologies.

Several synthesis techniques were used to inform the immersant about the agents' current actions in the environment. Short noise bursts were used as spatial cues since wideband signals provide more precise elevation cues. In addition, we created a bed of ambient sound serving to draw the immersant into the environment. We found that in this sonic atmosphere immersants felt more inclined to spend longer stretches of time within the world.

Rather than building a software solution specific to the AlloBrain project, graduate students at MAT designed a generalized system, the Cosm toolkit, to support the rapid development of many different kinds of projects within the Allosphere and similar spaces, incorporating audio spatialization, stereographic distributed rendering within a real-time fully navigable scene graph. The toolkit is currently implemented as a C/C++ library, and has been embedded within the Max/MSP/Jitter environment [30] to support real-time project design and testing. To make the best use of the Allosphere's current audio capabilities, the Cosm toolkit currently employs third-order 3D Ambisonics and distance-coding software developed at MAT (described earlier in this document), coupled with 3D motion and navigation algorithms for virtual environments.

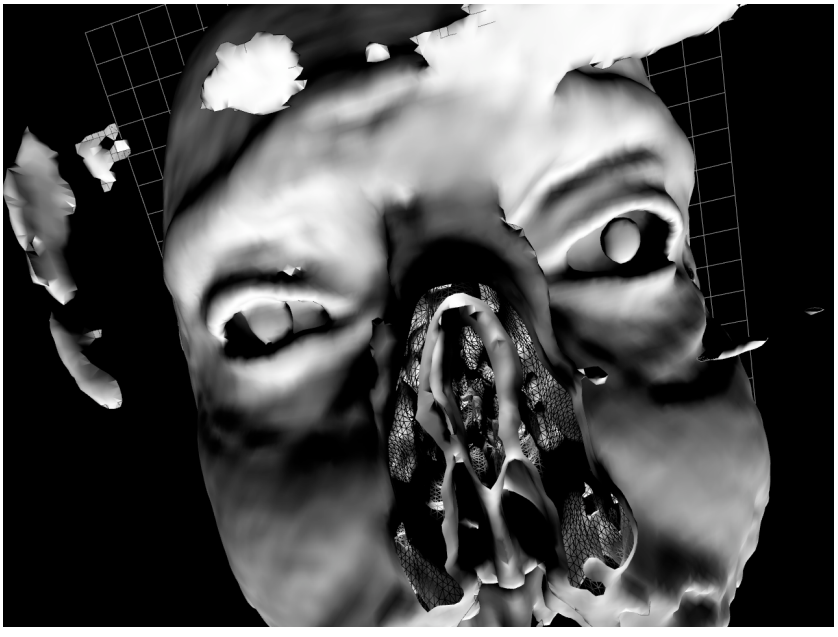

Fig. 10. Screen capture of the AlloBrain interactive recreation of the human brain from fMRI data. External frontal view in which even facial expressions are visible.

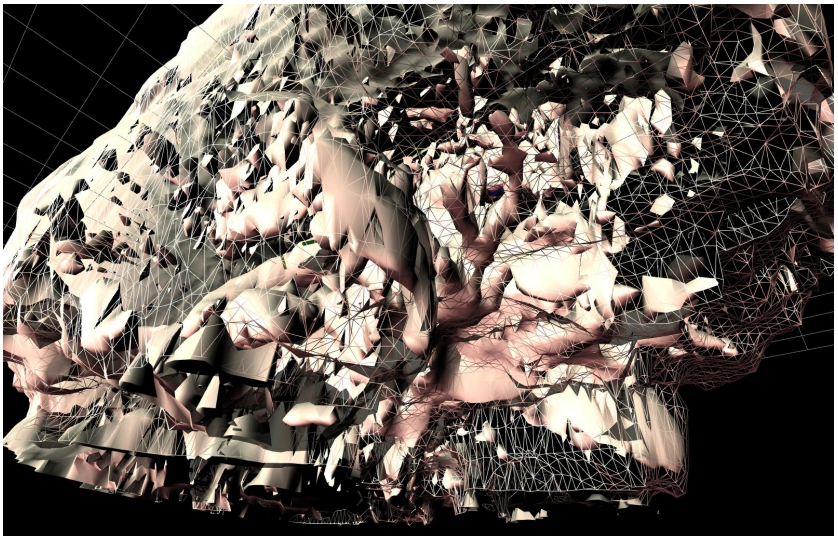

Fig. 11. Screen capture of the AlloBrain offering a side view with only some active layers. Users are able to navigate through the different layers of tissues and analyze the data in a collaborative immersive environment.

7 Conclusions

Once fully equipped and operational, the Allosphere will be one of the largest immersive instruments in existence. But aside from its size, it also offers a number of features that make it unique in many respects. In particular, it features immersive spherical projection, multimodal processing including stereoscopic vision, 3D audio, and interaction control, and multi-user support for up to 30 people. In this paper, we have focused on the audio infrastructure in the Allosphere, discussing the requirements, approaches, and initial results.

We envision the Allosphere as a vital instrument in the future advancement of fields such as nanotechnology or bio-imaging and it will stress the importance of multimedia in the support of science, engineering, and the arts. We have demonstrated first results in the form of projects of highly diverse requirements. These initial results feed back into the prototyping process but also clearly support the validity of our approach.

Although the Allosphere is clearly still in its infancy, we believe that the presented results are already meaningful and important, and will inform other integrative endeavors in the computer music research communities. The development of our prototype test-bed applications is geared towards an open generic software infrastructure capable of handling multi-disciplinary multi-modal applications.

References

1. Amatriain, X.: A domain-specific metamodel for multimedia processing systems. IEEE Transactions on Multimedia 9(6), 1284–1298 (2007)
2. Baalman, M.A.J.: Updates of the WONDER software interface for using Wave Field Synthesis. In: Proc. of the 3rd International Linux Audio Conference, Karlsruhe, Germany (2005)
3. Baalman, M.A.J.: Reproduction of arbitrarily shaped sound sources with wave field synthesis - physical and perceptual effects. In: Proc. of the 122nd AES Conference, Vienna, Austria (2007)
4. Ballas, J.: Delivery of information through sound. In: Kramer, G. (ed.) Auditory Display: Sonification, Audification and Auditory Interfaces, vol. XVIII, pp. 79–94. Addison Wesley, Reading (1994)
5. Berkhout, A.J.: A holographic approach to acoustic control. Journal of the Audio Engineering Society 36, 977–995 (1988)
6. Blauert, J.: Spatial Hearing. MIT Press, Cambridge (2001)
7. Castellanos, J.: Design of a framework for adaptive spatial audio rendering. Master's thesis, University of California, Santa Barbara (2006)
8. Chowning, J.: The simulation of moving sound sources. Journal of the Audio Engineering Society 19(11) (1971)
9. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.A., Hart, J.C.: The CAVE: Audio visual experience automatic virtual environment. Communications of the ACM (35), 64–72 (1992)
10. Freed, A.: Design of a 120-channel loudspeaker array. Technical report, CNMAT, University of California Berkeley (2005)
11. Gerzon, M.A.: Periphony: With-height sound reproduction. Journal of the Audio Engineering Society 21(1), 2–10 (1973)

12. Höllerer, T., Kuchera-Morin, J., Amatriain, X.: The allosphere: a large-scale immersive surround-view instrument. In: EDT 2007: Proceedings of the 2007 workshop on Emerging displays technologies, p. 3. ACM Press, New York (2007)
13. Hollerweger, F.: Periphonic sound spatialization in multi-user virtual environments. Master's thesis, Austrian Institute of Electronic Music and Acoustics (IEM) (2006)
14. Humphrey, W., Dalke, A., Schulten, K.: Vmd - visual molecular dynamics. Journal of Molecular Graphics (14), 33–38 (1996)
15. Ihren, J., Frisch, K.J.: The fully immersive CAVE. In: Proc. 3 rd International Immersive Projection Technology Workshop, pp. 59–63 (1999)
16. Malham, D.G.: Spherical harmonic coding of sound objects - the ambisonic 'o' format. In: Proceedings of the AES 19th International Conference, pp. 54–57 (2001)
17. Malham, D.G., Myatt, A.: 3-d sound spatialization using ambisonic techniques. Computer Music Journal (CMJ) 19(4), 58–70 (1995)
18. McCoy, D.: Ventriloquist: A performance interface for real-time gesture-controlled music spatialization. Master's thesis, University of California Santa Barbara (2005)
19. McGurk, H., McDonald, T.: Hearing lips and seeing voices. Nature (264), 746–748 (1976)
20. Pope, S.T., Ramakrishnan, C.: The Create Signal Library ("Sizzle"): Design, Issues and Applications. In: Proceedings of the 2003 International Computer Music Conference (ICMC 2003) (2003)
21. Pulkki, V., Hirvonen, T.: Localization of virtual sources in multi-channel audio reproduction. IEEE Transactions on Speech and Audio Processing 13(1), 105–119 (2005)
22. Rabenstein, R., Spors, S., Steffen, P.: Wave Field Synthesis Techniques for Spatial Sound Reproduction. In: Selected methods of Acoustic Echo and Noise Control, Springer, Heidelberg (2005)
23. Spors, S., Teutsch, H., Rabenstein, R.: High-quality acoustic rendering with wave field synthesis. In: Proc. Vision, Modeling, and Visualization Workshop, pp. 101–108 (2002)
24. Spors, R., Rabenstein, S.: Spatial aliasing artifacts produced by linear and circular loudspeaker arrays used for wave field synthesis. In: Proc. of The AES 120th Convention (2006)
25. Teutsch, H., Spors, S., Herbordt, W., Kellermann, W., Rabenstein, R.: An integrated real-time system for immersive audio applications. In: Proc. 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY (2003)
26. Theile, G.: Wave field synthesis - a promising spatial audio rendering concept. In: Proc. of the 7th Int. Conference on Digial Audio Effects (DAFx 2004) (2004)
27. Wakefield, G.: Third-order ambisonic extensions for max/msp with musical applications. In: Proceedings of the 2006 ICMC (2006)
28. Wegman, E.J., Symanzik, J.: Immersive projection technology for visual data mining. Journal of Computational and Graphical Statistics (March 2002)
29. Wenzel, E.M.: Effect of increasing system latency on localization of virtual sounds. In: Proc. of the AES 16th International Conference: Spatial Sound Reproduction (1999)
30. Zicarelli, D.: How I Learned to Love a Program that Does Nothing. Computer Music Journal 26(4), 44–51 (2002)