A 16-SPEAKER 3D AUDIO-VISUAL DISPLAY INTERFACE AND CONTROL SYSTEM

Mark F. O'Dwyer, Guillaume Potard & Ian Burnett

School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Northfields Avenue
Wollongong, NSW 2500, Australia
mo15@uow.edu.au, gp03@uow.edu.au, i.burnett@uow.edu.au

ABSTRACT

This paper details the CHESS system developed at the University of Wollongong. CHESS aims to provide a hardware and software platform for the creation, manipulation and playback of complex three-dimensional sound scenes. Ambisonic techniques are used to render a virtual sound scene on sixteen speakers arranged hemispherically around a user. A 3D visual representation of the scene is provided, which may be viewed on one or more display devices, including a virtual reality headset. User input may be provided via a 3D glove. These elements are combined to produce highly configurable immersive audio-visual applications with true three-dimensional audio. A control system has been developed for this system, which allows users to easily create complicated applications. We conclude by considering an example application of the system: cockpit and air traffic control systems.

1. INTRODUCTION

Spatial audio systems implemented using loudspeaker based methods have been investigated by many parties. More than a century ago, the first multi-channel techniques were devised to provide spatial sound. Clement Ader used multiple telephone transmitters and receivers to relay sounds from a remote location, the idea being to mimic the received sound field at the remote location [1]. In the 1930s Fletcher, Steinberg and Snow at Bell Laboratories proposed a system to produce a wall of sound using loud speakers [2]. In 1982, again at Bell Laboratories, Fox used a curtain of hanging microphones that were placed in front of the sound source. The signals from the microphones were used to drive a similarly laid out curtain of loud speakers, in an attempt to recreate the original sound wave front [1].

Over the last two decades a number of multi-speaker sound systems have gained mainstream acceptance, such as Dolby Surround Sound, which offer playback which is not restricted to one dimension. Generally, existing systems are not truly three-dimensional or they are not capable of reproducing a general three-dimensional sound scene. The system described in this paper represents a progression beyond these earlier methods.

A hardware and software system has been developed at the University of Wollongong that is capable of rendering a general synthesized three-dimensional wavefront. Sixteen speakers are arranged in a hemispherical format around a centrally located user who hears a number of sound sources move around them in a realistic and immersive manner. The 16-speaker dome is shown in Figure 1. This system is known as the Configurable Hemispherical Environment for Surround Sound, or CHESS.

A three-dimensional visualization of the sound scene is provided on one or more display devices, produced by a program developed using Java3D [3]. Java3D is an Application Programming Interface (API) for the Java programming language. It equips the programmer with a set of classes that provide a means of generating complex three-dimensional visuals, without needing to be intimately involved with low-level rendering functions. In Java3D, scenes are represented in a hierarchical structure where scenes are formed by linking appropriate classes to one another.

A control system has also been developed which stores all data associated with a three-dimensional sound scene and based on user inputs or assigned object behaviors, updates the scene periodically. It coordinates all data between itself, the audio system and visualization program. The different system programs are typically run on separate computers and communicate with one another over a network connection.

Sounds scenes may be static or manipulated in real-time and are defined using XML (Extensible Markup Language) according to a prescribed scheme. XML encapsulates data inside tags that may be named to describe the data [4].

The software developed to control the sound scene and corresponding visuals has been designed so as to allow future users of the system to create their own applications with as little programming as possible. This will allow many applications of the system to be explored without users needing to be intimately involved in its low level functions.

To the authors' knowledge no such three-dimensional audio-visual interface for a true 3D loudspeaker audio system has been reported.

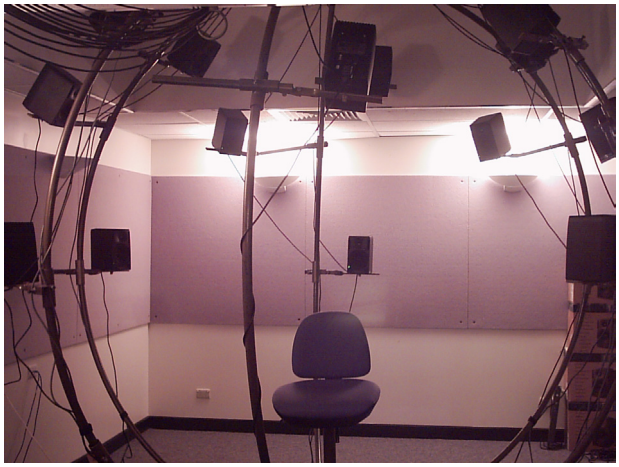

Figure 1. *The 16-speaker dome.*

The current system should be regarded as a prototype for a larger system which may be used with a larger audience.

2. SYSTEM IMPLEMENTATION

2.1. Audio Hardware and Software

The sixteen speakers used are identical and mounted on a mobile spherical scaffold which allows them to be freely positioned around the user [5]. This permits the configuration to be changed with relative ease, for example, changing between horizontal and hemispherical sound fields. Alternate configurations, for example cubic, have also been implemented. A multi-channel sound card is installed in a Macintosh G4 computer, which generates the speaker outputs.

The speaker outputs are produced by the 'DSP Layer' which is implemented using the MAX/MSP software package. Based on the positions of a number of sources in space, ambisonic techniques [1] are used to synthesize a three-dimensional wavefront.

Ambisonics were chosen over alternatives such as Vector Based Amplitude Panning and Wave Field Synthesis for reasons of its suitability to desired applications. Abisonics allows scenes to be recorded and played back later with different speaker arrangements. Scenes defined using B-format, for example sound recorded using a directional microphone, may be directly imported and easily added to existing sound scenes. Further, different order ambisonics may be used to achieve the desired precision of localization with acceptable processor load.

The loudspeaker implementation was chosen over options such as the Head Related Transfer Function method to avoid the complexities of head tracking so users may move in the scene, allow multiple users to be easily accommodated by the system and avoid the need to know an individuals HRTF for high quality results.

At present the 'sweet spot' at the center of the 16-speaker dome may accommodate one or two listeners at once.

2.2. Visualisation Software

A Java3D program has been developed to render a three-dimensional visual representation of the sound scene. The output of this program is highly configurable and may be displayed simultaneously on multiple display devices and from different viewpoints. This flexibility makes the system suitable to a variety of applications. Typically, sound sources will have associated and meaningfully related visual objects. The objects in the sound scene are then seen to move on the display/s and heard to move around in three dimensions, in an immersive manner. Additional visual objects, not associated with sounds, may be incorporated into the visual scene to create complex audio-visual environments. Visual objects may be either predefined system objects or input into the scene from an external file created with a software package or downloaded. At present the Object File (.obj) format is used.

The visualization system is a series of programs which may be used independently of the rest of the system as a simple 3D graphics engine. An instance of the visualization program is declared by whatever Java program wishes to use it and simple text based commands are issued to it to manipulate the objects in the scene. Based on these commands the visualization program manipulates the various classes in the Java3D hierarchy to alter the scene. Thus, no knowledge of Java3D is required by the user. The instance of the visualization program will usually be declared by the CHESS control program, which is discussed in Section 2.3.

If more than one display is required a visualization slave program is used. These programs are entirely separate to the master visualization program. They were chosen to be independent programs so as to allow them to be run on separate computers/platforms. The visualization slave programs receive their commands from the visualization master program rather than the control program itself. This scheme allows multiple views of the scene to be displayed on different devices, each which may be configured according to its individual needs. For example, the primary display may show a view of the entire scene viewed from an external location in the virtual scene by the controller and a slave display may provide a virtual reality like display of the scene for the user sitting inside the 16-speaker dome from their viewpoint in the virtual scene.

The visualization program is capable of producing highly configurable light sourcing effects that add to the realism of displayed scene. Effects include ambient lighting, point lights, spotlights and transparency.

Preliminary subjective testing has shown that users are generally able to locate the spatial location of sound sources based on audio cues only with this system. However, the tests have shown that the accuracy and speed with which sources can be localized increases with the presence of the 3D display.

2.3. Control Software

As was previously discussed, the DSP Layer resides on a Macintosh computer. The control program, which has been implemented in Java so as to function on multiple platforms, resides on a separate computer. The two computers linked to one another via a network connection. Communication is achieved through the Open Sound Control (OSC) protocol over the User Datagram Protocol (UDP). UDP does not feature rigorous error checking and packets may be lost, however, it does have the advantages of being relatively simple and fast. It is also possible to have more than one computer communicating with the DSP Layer, making it possible to have multiple remote control systems. Communication may occur from any remote location provided the network supports the required protocols.

Sources may have associated behaviors that specify how they are to move in the scene with time. Using these, it is possible to have objects move in a complex manner. Sources may also be manipulated in the scene by issuing text based commands to the control program or issuing it with an XML scene description document.

The control program updates the stored parameters associated with each source periodically or when a user input is received. If any data in the scene changes during this period, a series of data packets are sent over the network connection to the DSP Layer with the updated information. The visualization program also receives the updated information, which it relays to any slave visualization programs running.

Initial testing with subjects has revealed that the auditory and visual displays do not need to be updated with the same periodicity in order to achieve perceptually smooth motion. The update periodicity depends upon the speed of the moving objects, however, generally the visual output requires more frequent updating compared to the audio output. To this end the periodicity of both may be configured and adjusted real-time for changes in processor load and object speed. Typically however, update periodicities of 40 Hz and

25 Hz for visuals and audio respectively have been found to be acceptable.

The system has been comprehensively tested using a PC connected to the Macintosh DSP Layer using the existing switched network. The number of objects that can be manipulated in real-time with reasonable fluidity depends greatly on the nature of the scene itself. With reverberation effects associated with all sources the DSP Layer is capable of rendering up to five moving sound sources simultaneously before noticeable distortion occurs. This is due to the high processing requirements of reverberation effects. Without reverberation effects, the system has been tested with up to ten sources moving in a scene with packets being sent to the DSP Layer with a frequency of 25 Hz. With more objects than this or higher update frequency the network tends to become saturated and irregularities in source movement become clearly audible.

Figure 2 shows how the described components may be arranged.

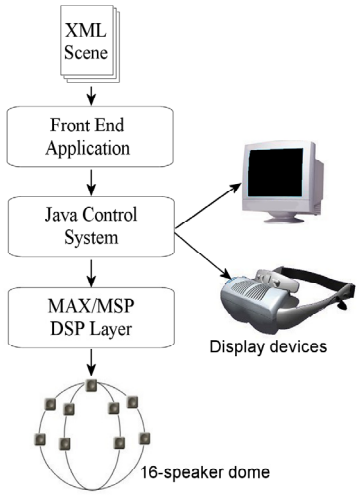

Figure 2. *Typical functional arrangement of the system components.*

2.4. Defining Three-Dimensional Sound Scenes

A sound scene description scheme has reported been by Potard and Burnett in [6] and is exploited in this system. Sound scenes are defined using the Extensible Markup Language (XML) format and use a centralized scene score containing temporal information such as timing events and object trajectories. Sound sources may have variable parameters that permit complex auditory effects. Among others, these include:

- Position: in three-dimensional space of the object in Cartesian or spherical co-ordinates.
- Orientation: a three dimensional vector or two rotation angles specify the direction of the source.
- Shape: it is possible to specify the wideness of the sound source which is formed using uncorrelated point sources.
- Directivity: describes the directivity pattern for different frequency bands. Correctly setting the directivity properties allows highly realistic virtual sound scenes to be created. For example, the back of a violin tends to emit more high frequency content than the front.
- Visual object: specifies the predefined shape or Object File to be associated with the sound source on the visual display.
- Behaviors: specify how the control program is to update the source's position with time.

Surfaces that obstruct and or reflect incident wavefronts may also be included in a scene. These are defined through a list of vertices and a set of transfer functions defining the incident angle dependent reflection/obstruction filtering.

The sound transmission medium can be defined which influences the propagation delay and attenuation of the wavefront. Medium parameters include the speed of sound, temperature, humidity and pressure.

The XML scene is parsed by the Java control system using the Document Object Model (DOM). JDOM [7], a Java implementation of DOM, is used to parse the XML document before it is translated into a series of commands native to the system. An XML document may specify anything from a small change to a single source, to an entire complex scene with many moving sources. An XML schema based standard for describing interactive 3D audio virtual reality scenes is used [6]. Alternatively, sound scenes may be defined though a series of text based commands.

2.5. Interactivity

The aforementioned control software has been developed to allow interactive three-dimensional audio-visual applications. The interactivity is provided through three primary systems:

2.5.1. Graphical User Interface

A graphical user interface allows a user to manipulate the three-dimensional sound scene in real-time. A series of simple buttons and sliders are provided to create sources, translate them, control audio effects such as reverberation, attribute behaviors and manipulate the visual objects among other parameters. The scene is viewed through one or more visualizations, typically one being used to show the scene from a remote location and another the view from the users point of view in the center of the 16-speaker dome.

2.5.2. Head Tracker

A head tracker has been incorporated to provide a user with the ability to effectively look around the scene with the display being updated accordingly. Though the head tracker is capable of tracking the users head movements over six degrees of freedom, since with the current system the user is at a fixed position, only two are required. Thus, the angle of elevation and the angle of rotation of their head are tracked. For simplicity and compatibility with a variety of devices, these movements are input into the system as mouse movements using Java's mouse classes. The head tracker used was found to be too sensitive to head movements and had a noticeable lag. Jitter filtering was used in an attempt to eliminate this, however, it results in the head tracker being too insensitive to pick up desired movements.

2.5.3. 3D Glove

A 3D glove tracks the users hand through six degrees of freedom. Like the head tracker, it was determined that only two

degrees of freedom were needed for initial applications. Again, the data from the glove is input into the system through Java's mouse classes. Translations in the horizontal and vertical directions are tracked and movements of the user's fingers picked up. This allows them to move their hand inside the scene and interact with the sound sources in real time. At present objects may be pushed away from the user using their hand, which results in the sound source and associated visual object to move away with an appropriate velocity. In the future it may be possible to have complex interactions with objects such as picking them up and moving them or even interacting with other users in a scene.

3. APPLICATIONS

The potential for creative use of this system is almost without limit. The hardware and software have been developed to such a stage that only minor modifications, if any at all, would be required when applying the system to a particular application. A front-end application is designed for the given task, which interfaces with the Java control system.

A number of applications have been investigated in order to illustrate the potential usefulness of the system and are at varying stages of development. One is discussed in some detail here.

3.1. Cockpit and Air Traffic Control Systems

The use of spatial auditory displays in pilot cockpits has been suggested in the past [8]. Such systems typically produce the auditory signals using the Head Related Transfer Function (HRTF) and are heard on fixed headsets. The possibility of a loudspeaker alternative has been investigated using CHESS.

The speakers are distributed around the pilot in three dimensions. A visual display is generated for the pilot using the Java3D CHESS program developed, which may be viewed through a head-up display in an actual aircraft situation. The control system is then stimulated by inputs from the aircraft's sensory equipment. External objects, which the pilot needs to be aware of, will prompt an input to the system, producing an audio-visual cue. The simulated spatial location of these in the synthesized audio-visual environment corresponds to the location of the actual object outside the plane. Since it has been shown that people respond more quickly to a combination of auditory and visual cues [9], the pilot's attention is more quickly drawn to this object than would be otherwise. This decrease in reaction time is clearly advantageous. Radio communications between pilots could also be spatialized.

A similar application of spatialized audio, which has been investigated using CHESS, is in an air traffic control situation. Loudspeakers arranged in hemispherically three dimensions and visual displays placed at regular angular intervals in the horizontal plane around the monitoring station. The planes being monitored are then seen on the display devices and heard through the speakers in a location of spatial relevance to its location in the outside environment. In this instance particularly, the loudspeaker method has clear advantages over a HRTF fixed headset method, as users positions and head movements are not likely to be fixed and would be difficult or bothersome to track.

The nature of the system developed allows it to be easily configured to work with any speaker arrangement, requiring only a small number of parameters to be changed.

3.2. Additional Applications

Loudspeaker based three dimensional audio systems such as CHESS may be applied to next generation multimedia, gaming and home entertainment systems. A larger version may be appropriate for theatres and presentation halls.

An Electroencephalogram (EEG) is a recording of activity in various regions of the brain. A three dimensional audio-visual system such as CHESS may provide an alternative and novel method of presenting the results of an EEG.

Generally teleconferencing applications feature zero, one or two-dimensional audio. An enhanced experience may be achieved using a three dimensional audio system.

Other applications of CHESS that have been suggested and are being investigated include virtual reality and three-dimensional music and acoustic art.

4. CONCLUSIONS

This paper has presented a functional background on a hardware and software platform developed to render a virtual three-dimensional wavefront on sixteen loudspeakers. Sound scenes are defined in XML or using a command based system. A control and 3D visual feedback system have been developed which allow complex immersive audio-visual applications to be developed. In fact, when developing many applications or scenes it may possible for the user to be oblivious to XML and/or programming.

The areas where CHESS may find application are limited almost only by the resourcefulness and creativity of the user. Musical, artistic, virtual reality, teleconferencing, air traffic control, cockpit and psychoacoustic evaluation applications have been developed with initial success. These applications highlight the possible usefulness of the system when fully developed.

The system will be continually expanded upon and additional and existing applications further investigated.

5. REFERENCES

[1] D. G. Malham and A. Myatt, "3D Sound Spatialisation using Ambisonic Techniques," *Computer Music Journal*, vol. 37(2) pp 157-188, 1995.

[2] C. Kyriakakis, P. Tsakalides and T. Holman, "Surrounded by Sound," *IEEE Signal Processing Magazine*, vol. 16(1) pp 55-66, January 1999.

[3] Java3D, http://java.sun.com/products/java-media/3D/

[4] I. S. Graham and L. Quin, *XML Specification Guide*, John Wiley and Sons, New York, 1999.

[5] G. Potard and S. Ingham, "Encoding 3D sound scenes and music in XML," in *Proc. ICMC*, Singapore, September 2003.

[6] G. Potard and I. Burnett, "Using XML Schemas to Create and Encode Interactive 3D Audio Scenes for Multimedia and Virtual Reality Applications," in *Proc. Fourth Int. Conf. on Fistributed Communities on the Web*, 2002.

[7] JDOM, www.jdom.org

[8] R. D. Shilling, T. Letowski and R. Storms, "Spatial Auditory Displays For User Within Attack Rotary Wing Aircraft," *Proc. Int. Conf. for Auditory Display*, 2000.

[9] M. Grohn, T. Lokki and T. Takala, "Comparison of Auditory, Visual and Audio-Visual Navigation in 3D Space," *Proc. Int. Conf. for Auditory Display*, 2003.