NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

HEADPHONE-DELIVERED THREE DIMENSIONAL SOUND IN NPSNET

by

Lloyd J. Biggs

September 1996

Thesis Advisor:	Michael J. Zyda
Thesis Co-Advisors:	John S. Falby
	Russell L. Storms

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
September 1996

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
HEADPHONE-DELIVERED THREE DIMENSIONAL SOUND IN NPSNET

5. FUNDING NUMBERS

6. AUTHOR(S)
Biggs, Lloyd J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943--5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The current MIDI-based sound system for the distributed virtual environment of NPSNET can only generate aural cues via loudspeaker delivery in two dimensions. To further increase the sense of immersion experienced in NPSNET, a sound system is needed which can generate aural cues via headphone delivery in three dimensions.

The approach taken was to explore the different feasible methods of rendering and presenting headphone-delivered spatial sound. One alternative was to implement a sound server capable of the real-time rendering of three dimensional sounds. Another alternative was to create a library of pre-recorded positioned sound files. In software, new algorithms were developed to integrate the sound server into NPSNET and to provide a table lookup capability for NPSNET's new spatial sound file library.

The result of this research is a sound server capable of rendering up to twenty-four simultaneous sounds for a single participant in NPSNET using "off-the-shelf" sound equipment and computer software. This sound server was tested during numerous demonstrations of NPSNET. This research provided another method of increasing a participant's level of immersion in NPSNET through the use of aural cues.

14. SUBJECT TERMS
Sound Localization, Spatial Audio, 3D Sound, Psychoacoustics, HRTF, Binaural Recordings, MIDI, Free-Field, NPSNET, DIS, Virtual Environment, Auditory Channel, Virtual Audio.

15. NUMBER OF PAGES
141

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

HEADPHONE-DELIVERED THREE DIMENSIONAL SOUND IN NPSNET

Lloyd J. Biggs
Captain, United States Marine Corps
B.S. Computer Engineering, University of Florida, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

September 1996

Author: Lloyd J. Biggs

Approved by: Michael J. Zyda, Thesis Advisor

John S. Falby, Thesis Co-Advisor

Russell L. Storms, Thesis Co-Advisor

Ted Lewis, Chairman,
Department of Computer Science

ABSTRACT

The current MIDI-based sound system for the distributed virtual environment of NPSNET can only generate aural cues via loudspeaker delivery in two dimensions. To further increase the sense of immersion experienced in NPSNET, a sound system is needed which can generate aural cues via headphone delivery in three dimensions.

The approach taken was to explore the different feasible methods of rendering and presenting headphone-delivered spatial sound. One alternative was to implement a sound server capable of the real-time rendering of three dimensional sounds. Another alternative was to create a library of pre-recorded positioned sound files. In software, new algorithms were developed to integrate the sound server into NPSNET and to provide a table lookup capability for NPSNET's new spatial sound file library.

The result of this research is a sound server capable of rendering up to twenty-four simultaneous sounds for a single participant in NPSNET using "off-the-shelf" sound equipment and computer software. This sound server was tested during numerous demonstrations of NPSNET. This research provided another method of increasing a participant's level of immersion in NPSNET through the use of aural cues.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

When the dust settles on this thesis, I can say I accomplished my primary goal -- to implement headphone-delivered 3D sound in NPSNET. I wanted to accomplish this goal by developing a "home-baked" solution in order to save money but ran into dead-ends at every turn. The solution ended up costing $10,000 and turned out to be an implementation of a known commercial solution. For that I am disappointed. But the effort I expended in researching the different alternatives is documented and my hope is that anyone who may follow me in this research can pick up the guidon and take it from here without blundering down the same paths that I did.

I must acknowledge those who helped me complete this thesis and the research behind it. My first acknowledgment and thanks goes to Russell Storms. He was my mentor for this research giving me direction, ideas, assistance and support. Russell showed great patience and sympathy as I whined about how disappointing my thesis was turning out. Thank you Russell! Thanks to John Falby for his instruction and agreeing to be my sponsor for the directed studies classes I took to better get a handle on the different areas of research connected to 3D sound production. His lockstep approach that he imposes on his students for thesis milestones had much to do with me getting this thesis written. Procrastination is my enemy and John helped me fight it. Thanks John. Thanks goes to Paul Barham for his patience with my endless questions about NPSNET and also for his code that I was able to "re-use" for the implementation of NPS-ACOUST. Thanks Paul. Thanks especially goes to my thesis advisor Dr. Mike Zyda. He was very amenable to everything that I wanted to pursue during the conduct of this research. He allowed me the freedom to pursue the different alternatives and then forked over $10,000 of his research budget when it was apparent that I wasn't going to come up with a locally developed solution. Thank you Mike.

Lastly, I would like to acknowledge my wife, Andrea. Her patience and support as I camped out in front of the computer for hours on end allowed me to devote my full attention to all aspects of my academic pursuits here. Thanks sweetheart!

I. INTRODUCTION

There are many facets to a virtual world. For people to participate in a virtual world, they must have some sense of immersion and interaction with objects simulated in a three dimensional (3D) environment. To achieve the goal of total immersion, all of a person's senses must be stimulated. However, only the visual, hearing and to a lesser extent, tactile senses have been seriously addressed in virtual world research to date. The topic of this thesis addresses methods of introducing sound into virtual worlds using headphones in a way that leads a user further down the path of immersion.

A. MOTIVATION

The motivation of this thesis is to design and implement an appropriate headphone-delivered 3D sound system for use with the Naval Postgraduate School Networked Vehicle Simulator (NPSNET) [ZYDA93] [ZYDA94] [MACE94]. NPSNET is a distributed, interactive, real-time networked computer application that allows users to participate in virtual world simulations. The system was developed by the NPS Computer Science Department in their Graphics and Video Laboratory. The goal of NPSNET is to be a "low-cost" solution for virtual world applications. To accomplish this goal, the NPSNET Research Group (NRG) uses commercially available off-the-shelf software and hardware to implement the environment. Additionally, NRG Ph.D. and MS students make valuable contributions to NPSNET research projects.

One of the features of NPSNET is its use of the Distributed Interactive Simulation (DIS) networking protocol. DIS is a jointly sponsored networking format that standardizes information about virtual world entities. Developed at the University of Central Florida Institute for Simulation and Training, these simulation standards were an outgrowth of the Defense Advanced Research Projects Agency (DARPA) Simulation Networking (SIMNET) project. One of the key features of DIS is that separate DIS-compliant virtual world applications can interact with each other over a communications network, most notably, the internet. [MACE94][MACE95]

In order for these separate virtual world applications to interact with each other, they must share information about the entities that comprise the simulated environment. The information shared is communicated via DIS Protocol Data Units (PDUs). Suffice it to say, to support a robust virtual world, many DIS PDUs are needed to describe all manner of things related to the participating entities and their environment. Generally speaking, however, there are two types of PDUs -- simulation and control. Simulation PDUs describe an entity's state and actions while control PDUs focus on message passing between participants. Control PDUs primarily facilitate the passing of logistics coordination data. NPSNET currently employs only three simulation PDUs -- Entity State, Fire and Detonation. The Entity State PDU (ESPDU) describes an entity's identity (e.g. tank, helicopter, etc.), position, orientation, velocity and actions. As the data for the entity changes, the changes are broadcast to other simulation participants over the network using an ESPDU. As the participants receive the PDUs, they use the standardized information to make calls to their applications library and in turn present the simulation of the entity visually and aurally.[ZESW93]

The aural aspect of simulated entities can be presented in two ways -- loudspeakers (open-field) and headphones (closed-field). When the host computer for a participating NPSNET entity (herein referred to as a "player") receives a DIS PDU describing an external entity or event in the simulation, the host computer running NPSNET delivers the appropriate visual and aural cue to its player. For example, if a helicopter (a player from a different host) flies near the local player in the simulation, the sound of a helicopter engine should be delivered to the local player. If the helicopter fires a missile, the sound of the missile firing should be heard as well as the subsequent missile impact and detonation (if the local player is close enough to hear the detonation). In this example, not only did the host computer receive a "helicopter" PDU (entity state), but it also received a "missile firing" PDU (fire) and an "explosion" PDU (detonation). Upon receiving these PDUs, the host computer would play a helicopter sound, a missile firing sound and a missile detonation sound. However, it is not sufficient to simply play the appropriate sound cue for a given event. To continue progress towards the goal of total immersion, a more realistic presentation of the sound is needed. Namely, we must strive to present the sound spatially. If in our example the helicopter is to the left in reference to the local player's position and orientation in the virtual world, it would be appropriate to present the corresponding aural cue in such a way that it actually sounds as if the helicopter is on the left. This is the subject of much research in the field of virtual world simulations as well as the primary motivation for this thesis.

B. RESEARCH OBJECTIVES

Past NPS students working in the area of spatial sound developed several working models for delivering 3D sound in the NPSNET environment. However, these applications all concentrated on delivering spatial sound using loudspeakers [ROES94][STOR95]. The primary objective of this research is to implement a headphone-delivered sound system for integrating 3D sound cues into NPSNET.

The following are the objectives of this thesis:

- Identify, compare and contrast the different methods of rendering headphone-delivered spatial sound.

- Identify hardware and software applications capable of producing headphone-delivered spatial sound.

- Identify the capabilities and limitations of each hardware and software application alternative and their applicability to NPSNET.

- Investigate the possibility of generating sounds from the same workstation being used by a player participating in an NPSNET session.

- Design and implement an application capable of delivering pre-recorded, headphone-delivered spatial sounds into the NPSNET virtual world.

- Investigate the possibility of implementing a sound server that can service the audio needs of multiple clients participating in a single NPSNET session.

- Provide an appropriate direction for future NPSNET headphone-delivered sound systems.

C. SCOPE

The focus of this research is the development and application of a headphone-delivered spatialized sound system for use within NPSNET. The primary goal of this research is to increase the level of immersion for a virtual world participant by introducing realistic 3D audio cues. Secondary goals include:

- Low-cost solution - Ideally, every virtual world participant should be presented with robust spatial audio to enhance their participation and increase their level of immersion. The requirement that hundreds and in some cases thousands of players be allowed to simultaneously participate in the same virtual world dictates the need for a low-cost per player spatial audio solution.

- Ease of use - The solution should be easy to implement, use and maintain for participants and follow-on researchers. Implementations that are difficult to understand are rarely used and become "shelfware."

- Future work - Because this thesis is the first to introduce headphone-delivered sound in NPSNET, it should lay the groundwork and direction for future research in this area.

D. ASSUMPTIONS

There is no certain level of knowledge that the reader is assumed to possess in order to read and understand this thesis. Practically all the concepts discussed in this research are presented with the layman in mind. However, this research is better understood if the reader has a basic knowledge of computers, virtual worlds, audio systems, and acoustics.

E. LITERATURE REVIEW

In the preparation of this research, a thorough literature review was performed. The results of this review were instrumental in preparing this research and are presented as an annotated list of references which can be found in the bibliography. This list is a conglomeration of references which were gathered from various research efforts including: 1) Elizabeth Wenzel from NASA-Ames Research Center; 2) Richard Duda from San Jose State University; 3) Center for Computer Research in Music and Acoustics (CCRMA) from Stanford University; and 4) the NRG Auralization and Acoustics Laboratory at the Naval Postgraduate School. This consolidated list is quite exhaustive including numerous facets of sound as it pertains to various theories and applications. This list is a helpful resource for anyone interested in pursuing further research of sound not only as it pertains to its use in virtual environments, but also in practically any application.

F. THESIS ORGANIZATION

This thesis is organized into seven chapters and four appendices. Chapter II provides a background of the properties of 3D sound perception. Chapter III outlines previous work in headphone-delivered spatial sound as well as previous attempts at delivering spatial sound for use in NPSNET. Chapter IV describes the current environment in the NPS graphics lab. Chapter V discusses the research of three different in trying to solve the problem of spatial sound generation. Chapter VI discusses the Acoustetron II and its applicability to NPSNET. Chapter VII concludes the thesis with the work accomplished and future research defined.

Appendix A contains a list of definitions and abbreviations used throughout this thesis. Appendix B contains the user guide for setting up and running the Acoustetron II and NPS-ACOUST. Appendix C lists and describes the sounds available on the Acoustetron II. Appendix D outlines a proposal for a common NPSNET sound class interface.

G. DEFINITIONS AND ABBREVIATIONS

See APPENDIX A: DEFINITIONS AND ABBREVIATIONS on page 89 for a list of definitions and abbreviations relating to pertinent aspects of this research.

II. BACKGROUND

To present the topic of 3D sound in a distributed virtual environment, the theory of sound and its localization perceptions must be discussed. Once these theories are understood, the mechanics of sound localization can be modeled and implemented in a synthetic environment. This is not a task easily accomplished. There are many factors that contribute to our ability to locate sound, some of which are directly contributed to mental processes not easily modeled or reproduced in a virtual world. For the purpose of this thesis, the terms *localized sound, spatialized sound,* and *3D sound* all mean the same thing -- namely that a sound is presented at a specific azimuth, elevation and distance from a listener.

A. BINAURAL SOUND

Recorded sound can be divided into three categories: monaural, stereo and binaural. Monaural sounds are recorded using one microphone. When replayed, there are no sound localization cues. In other words, the monaural sound has no recorded positional information. When the sound is replayed, the sound is positioned in one place. Over headphones, a monaural sound is presented directly in the center and inside the listener's head. Stereo sound contains some positional information and is perhaps most familiar to people who listen to music. Recorded with two microphones, stereo sound has lateral positional information. It is presented laterally depending on the position of the microphones during the recording. When listening to the playback of stereo sound, the lateral position of the sound can be detected. However when listening with headphones, the sound is still inside the head of the listener because it does not contain any of the externalization sound cues normally present when we listen to actual sound. Binaural sound recording captures these externalization cues. Binaural sound recording is accomplished by inserting very small microphones into the ears of either a live person or a dummy. The small microphones should be of sufficient quality to capture not only the sound source but also sound localization cues that help us perceive direction and distance of sounds. Researchers interested in inserting 3D sound into virtual environments pursue binaural sound production methods.

There are many kinds of sound externalization cues captured in binaural recordings. These different sound cues influence the way we perceive spatialized sound. The two major components of spatialized sound research are psychoacoustics and sound localization theory. Additionally, a head centered coordinate system has been developed as a way of describing and applying directional vectors that represent the positional relationship between a sound source and a listener. Each of these topics are briefly discussed.

B. PSYCHOACOUSTICS

Psychoacoustics is the term applied to the contribution of the mental aspects of sound interpretation. Physical factors such as sound waves and the mechanics of how we *hear* sound play only a part in how we *perceive* sound. Vision, familiarization with the sound or its source, and other mental factors also play a crucial part in perceiving localized sound. While vision is a sense that we can model in a virtual world through the display of computer rendered 3D objects, real world visual cues can often fool our sense of hearing, making us believe we are hearing sound from a visual source what is not actually emitting sound. This is a mental "slight of hand" that is not well understood nor easily modeled. Additionally, familiarization with a sound or sound source is another mental ability that helps us quickly assimilate sound localization cues and make position and distance determinations. A virtual world simulation would require the ability for entities to remember aspects of its environment and instantly associate that data with presented aural cues. Today's computer memory and performance limitations make this an unrealistic goal. The familiarity factor is another facet of our mental abilities not easily modeled in a virtual world simulation.

C. SOUND LOCALIZATION

Sound localization theory is the culmination of scientific research and discovery about the physical factors of sound perception and interpretation. Although much is still unknown about how we localize sounds, it has been discovered that the following physical cues play a major role: interaural time difference, interaural intensity difference, pinna response, shoulder echo, head motion, early echo response, reverberation, and vision. Other cues include atmospheric absorption, bone conduction, and a listener's prior knowledge of the sound source. [BURG93]

1. Interaural Time Difference

Because sound travels at a finite speed, distances and delays can be detected by the human ear. Each ear hears sounds differently. For example, if a sound source produces sound from a person's front left, the left ear will hear the sound slightly before the right ear. This difference is called the interaural time difference (also known as interaural delay) and has much to do with the ability of estimating the direction of the sound source. Figure 1 shows a graphical representation of the interaural time difference.

2. Interaural Intensity Difference

The interaural intensity difference is the sound intensity that is received by each ear. In the same example above, the right ear will hear a slightly less intense sound than the left ear because of the position of the right ear relative to the sound (the ear faces away from the sound source). Other factors influencing sound intensity are the density of the cranium in which the sound travels through (also known as *head shadowing*) and the different echo angles in which the ear receives sound. Figure 1 shows a graphical representation of the interaural intensity difference.

3. Shoulder Echo

Shoulder echo also makes its contribution. Echoed sound waves reflect off a person's shoulders and strikes the ears at different angles/times than do the sound waves that traveled directly from the sound source to the ears. Other echoes are present as well.

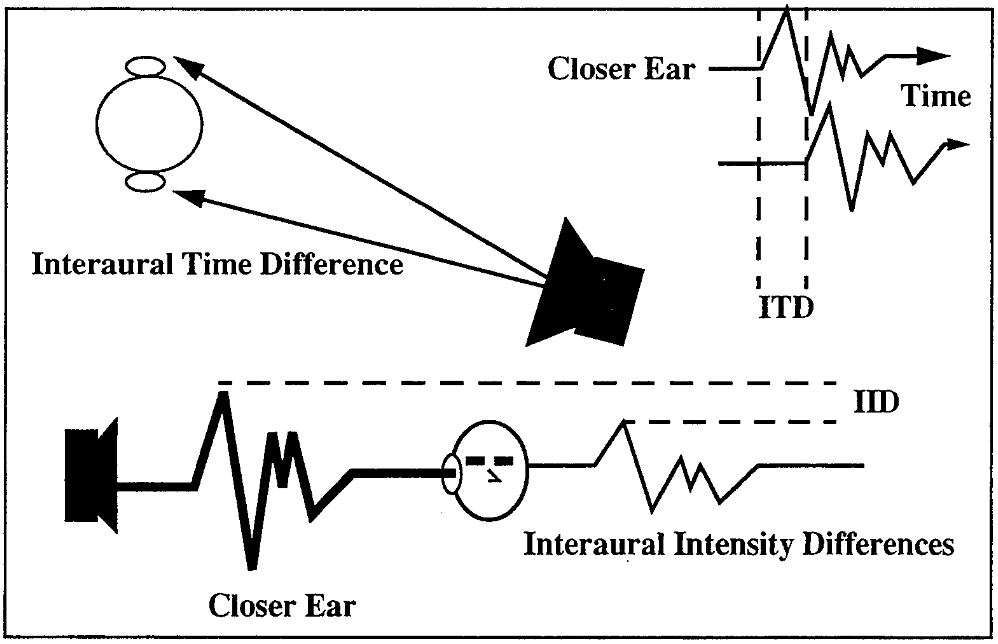

Figure 1. Two primary cues of sound localization. From [STOR95].

Any object that reflects sound produces an echo that is also received by both ears. The different arrival times and intensities of these echoes contribute to sound localization. Figure 2 shows examples of different echo sources.

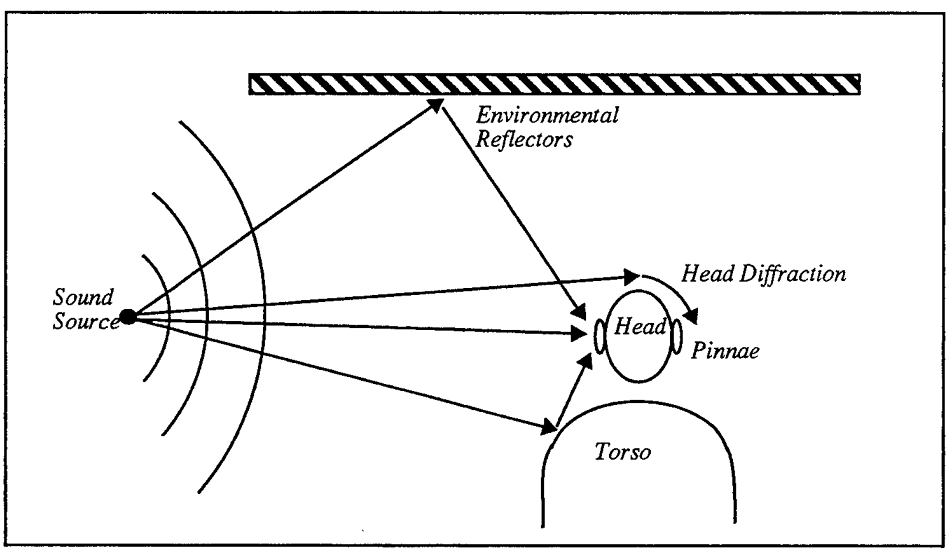

Figure 2. Acoustic Paths. From [STOR95].

4. Early Echo Response

Early echo response are the echoes perceived shortly after (50 -100ms) the original sound source. These early echoes combined with the follow-on reverberations provide additional directional and distance cues. Echoes received outside this time threshold are usually not associated with the original sound source but with the location of the surface that reflected the sound. If the echoed sound is received before the actual sound, our sense of locating the sound may be fooled. This is known as the *precedence effect* and is treated with some detail in [STOR95].

5. Pinna Response

The pinna response is a term used to describe the shape of the ears and their role in externalizing sounds. It has been discovered that the ear shape plays a much larger role than previously thought in how individuals localize sound.

6. Head Motion

Head motion describes the natural tendency for humans to orient their head towards the perceived direction of the sound. As the head moves, the localization cues shift as well. The shifting of the cues provides yet another clue as to the direction of a sound source.

7. Vision

Finally, vision plays an important role in the psychoacoustical aspects of sound localization. We combine the aural cues presented with a visual lock of the source to locate its position and distance. Sight plays such an important role that it is entirely possible that while the sound cues perceived indicate sound from one direction and distance, a different visual cue might override the sound cues and cause us to misperceive the location of a sound source. [TONN94]

D. SUMMARY

The main problem in applying spatialized sound in a virtual world is producing sound that is correctly peppered with localization cues so the listener hears a realistically positioned sound. The word *producing* implies that we want to create a sound and place it in 3D space without the benefit of an actual sound source emanating from that position. That is the crux of this research. Fortunately, the physical aspects of this procedure are well understood. Much research has been accomplished and results obtained. We are no longer in the position of having to rely on pre-recorded binaural sound samples to recreate a positional sound. We now have the ability to synthesize spatial sound from single monaural recorded sound samples through the use of Head Related Transfer Functions (HRTF). HRTF's measure a person's ability to hear spatial sound and are created in the following manner. Tiny microphones are inserted into a person's ears who is then exposed to numerous pre-recorded sound samples at different positions relative to the person's head. These sounds are re-recorded using the tiny inserted microphones and the resulting recording is compared to the original sound sample data. The comparison yields a set of linear functions (HRTFs) that describe the unique externalization cues for the individual. The HRTFs are then used to create a set (one for each ear) of finite impulse response (FIR) filters. Each FIR filter is used to manipulate a monaural sound sample and present two slightly different sound samples, one for each ear. The difference in these two sound samples are the differences that make up the externalization and localization cues associated with spatialized hearing. These two filtered monaural sound samples are combined into one 2-channel sound sample. When presented to a listener, the simultaneous replay of the two filtered sounds to each ear gives the effect of spatial hearing.

Once these FIR filters are obtained, the next step of inserting spatialized sound into a virtual world seems relatively straight forward. Populate a virtual environment with as many monaural sound samples as are needed for each sound event. Only one sound file is needed for each sound event because we can than take that one file, manipulate it using a listener's FIR filters and place the sound into the virtual world. Ideally, we would want to have this filtering technique implemented in a real-time environment so the instant a sound event occurs, the representative sound sample is filtered and replayed to the listener.

There are several problems in accomplishing this goal. The actual filtering of a monaural sound file using FIR filters is computationally expensive. Processor resources are precious in a real-time graphics environment. While sound is certainly important in a virtual world, graphics usually receives more emphasis. Due to the processor intensive nature of calculating real-time 3D sound, processes that render graphics and 3D sound cannot co-exist on the main processors of today's graphics workstations. Moreover, as in the real world, a virtual world would contain many simultaneous sounds. The ability to filter four monaural sound files simultaneously would tax even the most powerful processors of today. However, even four simultaneous sounds in a virtual world is an unrealistic restriction. As an example, NPSNET can easily handle ten players at one time. Ten players would each have a vehicle that at a minimum is capable of motion (engine noise) and weapons firing (firing and detonation of explosive munitions). Three sounds for each player would make thirty sounds possible at a minimum. If all ten players are located in the same vicinity in the virtual world, it is possible that there would be thirty simultaneous sound events each requiring filtering and placement within the virtual world. The real-time production of 3D sound would have to be sequentially very fast or accomplished concurrently (one process for each sound event) so that little or no latency occurs between the sound event and the delivery of the actual 3D sound. There are no commercially available, low-cost computer platforms that exist today that could handle the graphics and networking responsibilities of a virtual simulation as well as the burden of real-time production of multiple, simultaneous, spatialized sounds. This leaves two alternatives for 3D sound rendering -- separate sound hardware that would constitute a sound server or non real-time recording of several pre-positioned sound files having a lookup table to play the appropriate sound when a near match sound event occurs. A discussion of previous work on these two ideas is presented in the next chapter.

III. PREVIOUS WORK

Much research has been conducted in creating and delivering spatialized sound. As mentioned earlier, the two areas of research in 3D sound delivery are open-field (loudspeakers) and closed-field (headphones). Because the focus of this thesis is headphone-delivered spatial sound in NPSNET, the work specializing in this type of sound delivery will be reviewed along with the work accomplished by researchers connected to the NPSNET series of 3D sound research. The relevance of previous 3D sound research in NPSNET, albeit in an open-field format, makes it necessary to recount previous experiences and accomplishments.

A. NPS SOUND

NPSNET researchers first attempted to insert sound into the NPSNET environment in 1991. Two NPS students (Major Joseph Bonsignore and Elizabeth McGinn) created a system that was the basis for today's NPSNET sound environment (see Figure 3). They used a Macintosh IIci connected to a SGI workstation via an RS-232 serial cable interface. The SGI workstation would send the name of a sound file to play to the Macintosh. Macintosh software would decipher the filename and then in turn play the appropriate sound file through the use of a soundcard. Although this was a significant advance for the NPSNET environment, the solution had several problems. The sounds were not spatialized, there was a noticeable latency in NPSNET sound events and the actual sound played to represent that event (i.e., sounds could not be replayed in real-time) and only discrete sounds such as explosions could be replayed. Continuous sounds such as a running helicopter engine could not be replayed. In spite of these problems, this first attempt at inserting sound into NPSNET served to validate the idea that sound cues were feasible in a real-time virtual world simulation and served as the basis for further work at NPS in this area.[STOR95]

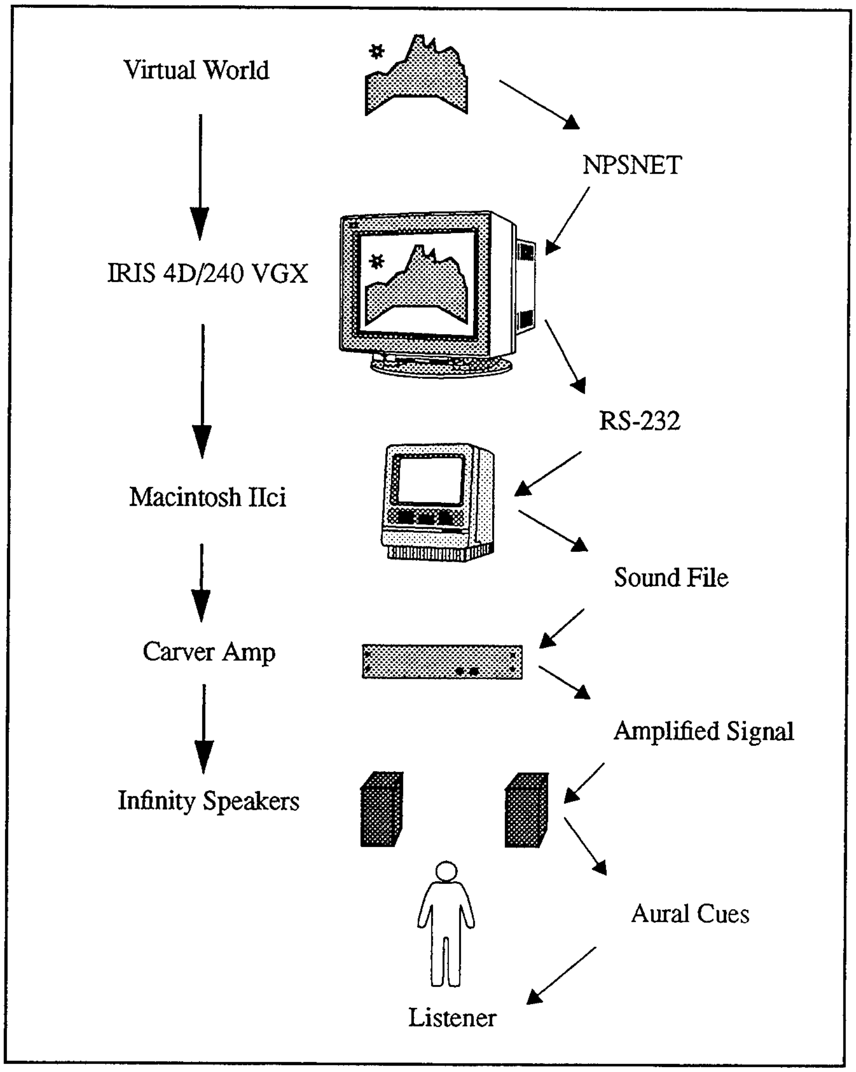

Figure 3. Overview of NPS-Sound. From [STOR95].

B. NPSNET SOUND SERVER

Following NPSNET Sound, more work was accomplished by a follow-on MS student, Lieutenant Leif Dahl and a NPS summer hire employee, Ms. Susannah Bloch. The next generation of NPSNET Sound came in the form of a sound server (see Figure 4). It replaced the Macintosh with an EMAX-II digital sound sampler as the sound server. The EMAX-II was loaded with digital sound samples such as explosions and firing weapons. Because the EMAX-II was a MIDI driven device, a C program was written to send MIDI

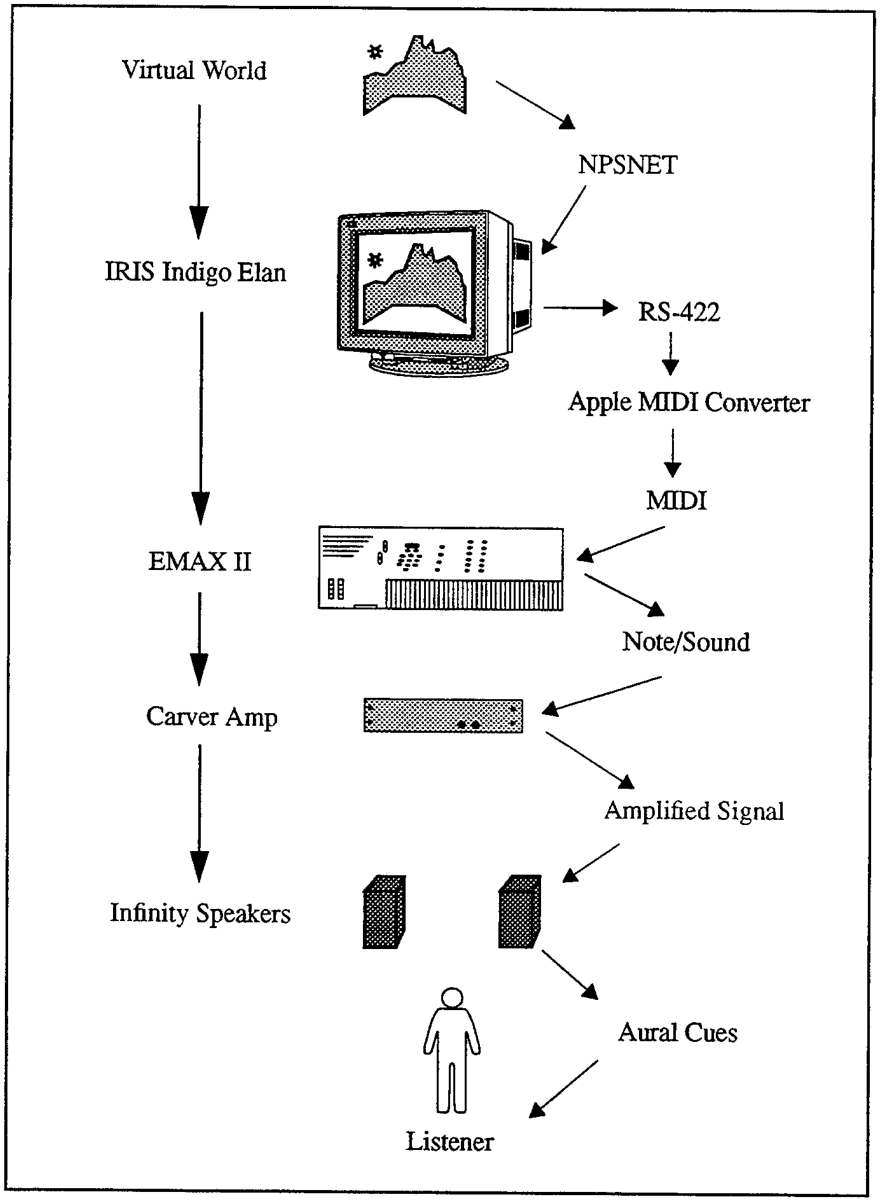

Figure 4. Overview of NPSNET Sound Server. From [STOR95].

commands from an SGI workstation to the EMAX-II telling the EMAX-II which sounds to play. The program also monitored the NPSNET network and captured DIS packets that indicated events that needed sound attached. The continuing work on NPSNET Sound Server decreased latency and increased the flexibility of NPSNET Sound through the use of MIDI commands. However, the lack of continous and spatialized sounds continued.

Further, sound events coming from moving vehicles began to be considered as a desirable addition.

C. NPSNET-PAS

Further work by Lieutenant John Roseli extended the NPSNET Sound Server to include spatialized and continuous sounds. A new program, NPSNET-Polyphonic Audio Spatializer (NPSNET-PAS), was written to enhance the sound cues presented in NPSNET (see Figure 5). Two dimensional spatialized audio cues were presented over four speakers in addition to low-level frequency sounds delivered over two subwoofers to give the "rumbling" effect present when operating heavy machinery. Still, continuous sounds were fixed in one place - there were no provisions for implementing moving sounds. However, pitch bending was added to the continuous sounds to give the effect of raised and lowered engine RPMs. NPSNET-PAS was a significant step forward towards the goal of immersion.[STOR95]

D. NPSNET-3DSS

Continuing work in NPSNET Sound was accomplished by Captain Russell Storms, USA in 1995. He developed the NPSNET-3D Sound Server (NPSNET-3DSS) (see Figure 6). NPSNET-3DSS improved on NPSNET-PAS in that it provided open-field sound cues in three dimensions. NPSNET-PAS was extended from four speakers to eight speakers in a "sound cube" configuration. Additionally, synthetic reverberation was used to give the effect of distance perception. This synthetic reverberation was accomplished using Ensoniq DP/4 Digital Signal Processors (discussed in the next chapter). Additionally, Captain Storms implemented a model for the Precedence Effect (PE). The PE is another cue that helps humans localize sound. Simply stated, if a sound wave arrives at the ear and corresponding echoes arrive an instant later, the first sound source heard is the direction in which we perceive the sound coming from. If we hear the echoed sound first, then we perceive the sound coming from the source of the echo. The PE is an important cue in helping to localize sound and was implemented in his sound cube configuration. However,

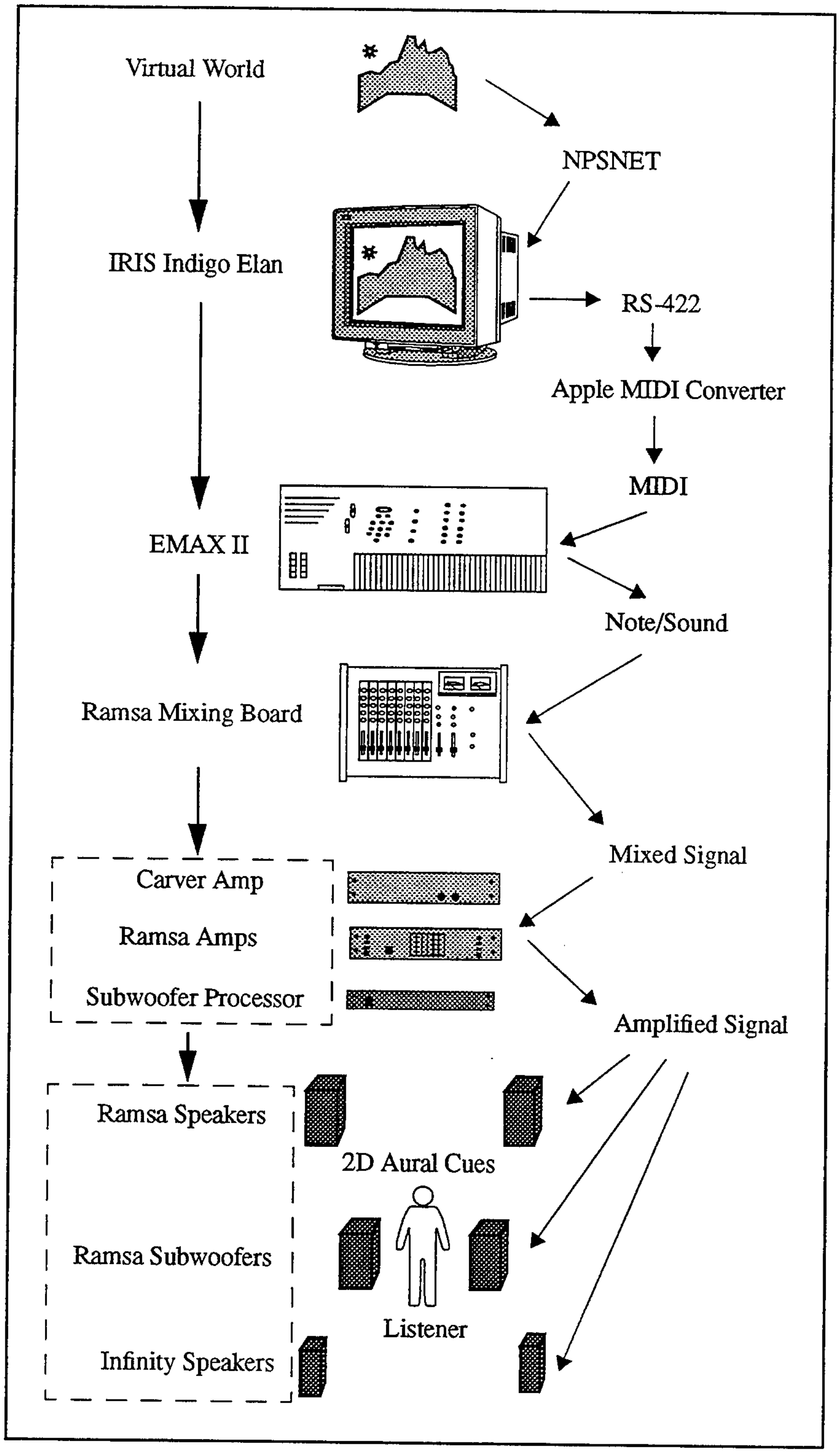

Figure 5. Overview of NPSNET-PAS. From [STOR95].

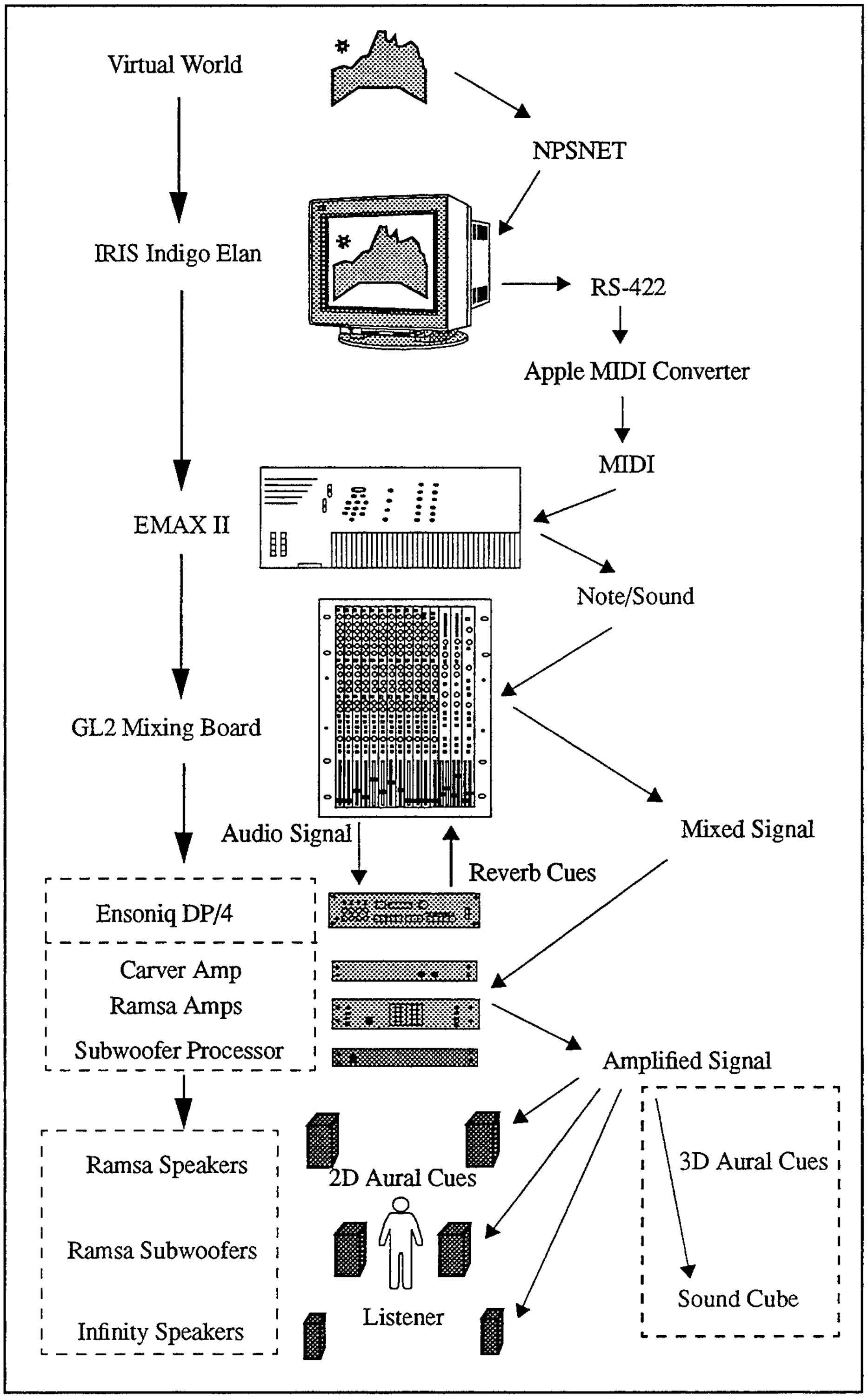

Figure 6. Overview of NPSNET-3DSS. From [STOR95].

due to hardware limitations, he was not able to create echoes fast enough (a maximum 30ms time delay was necessary to effectively imitate an echo) rendering the PE sound model ineffective.[STOR95]

E. MERCATOR PROJECT

In 1992, E.D. Mynatt and W.K. Edwards of the Georgia Tech Graphics, Visualization, and Usability Center (GVU Center) worked on the Mercator project. The Mercator project attempted to provide blind users with a 3D sound interface to X-windows applications. The components of the X-windows display were mapped to spatialized auditory cues to help the blind user navigate through X-windows graphical user interfaces (GUIs). HRTFs and FIR filters were used to map the sounds. Because a comprehensive spatial audio system can easily overwhelm system processor resources, the spatial audio system was developed in a client/server fashion. The system was implemented using an Ariel S-56x DSP controller board for the spatial audio filtering, a SPARCstation IPX host machine and an Ariel ProPort 656 for digital to analog conversion. Although a SGI Indigo workstation has its own built in DSP engine, the researchers decided not to try the difficult method of porting the DSP microcode and associated host-side driver software to the SGI Indigo. As for the client/server relationship, they used simple UDP-based routines to communicate messages between the audio clients and the 3D sound server. They connected an SGI Indigo Elan via an ethernet LAN to the SPARCstation sound server. Position information was sent from the Indigo to the SPARCstation and the sound server in turn used the appropriate FIR filters to spatialize a given sound source. The spatialized sound was then sent back to an amplifier via coax cables and played over headphones back to the blind user. See Figure 7 for details on the connections [BURG92].

The Mercator Project research was especially important. It validated the idea that spatialized sound processes cannot be co-located on the same processor as graphics intensive processes where even a reasonable frame rate is desired. It also provided the idea of a client/server alternative to co-locating processes on the same workstation.

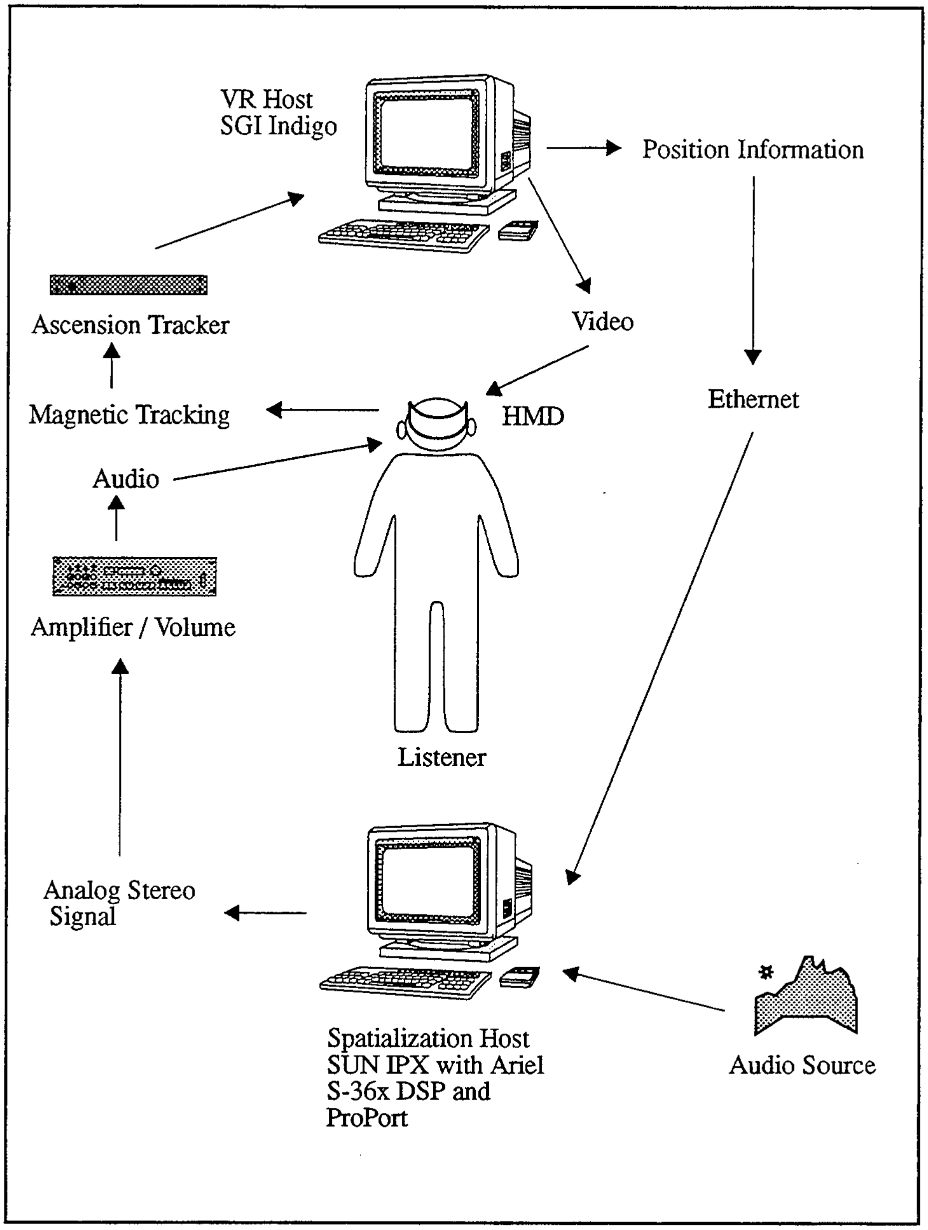

Figure 7. Overview of NA3 Audio Server.

F. EXPERIMENTAL VIRTUAL ACOUSTIC DISPLAY

In 1993, an experimental 3D acoustical display was developed by Mr. Andrew Wheeler and Mr. Joshua Ellinger at the Applied Research Laboratories, University of Texas. Their goal was to create a low-cost virtual acoustic display in which users could encode spatial cues onto monaural sound data. They used the same filtering process discussed above with HRTFs and resultant FIR filters. There was no budget for their project so they had to borrow the parts for their experiment. They obtained a Motorola 56001 digital signal processor (DSP) based wire-wrapped controller board. The board consisted of the DSP chip which ran at 20Mhz, 32k by 24 bit static RAM, 32K by 8 bit ROM, decode logic, and an RS-422 driver chip. They also borrowed a Crystal 4215 codec-based evaluation board, which supported 2-channel CD quality A/D and D/A throughput, and an IBM PC clone. To listen to the resulting spatial sound, they borrowed a Rotel RC980BX preamplifier and Sennheiser HD530 headphones. Once all this gear was put together, they made modifications to the software on the DSP controller board. The result of their experiment was the ability to spatialize one sound in one of 144 locations within the 3D space of the listener. The participants in the experiment were able to locate the spatialized sound within15 degrees azimuth. As the spatialized sound approached the median plane, front-back reversal problems occurred in which the participant was confused as to whether a sound was in front or behind. They noted that this might have largely been overcome if visual cues had been provided. They also observed that the participants often moved their heads when they heard a spatial sound. This seemed to give credence to the idea that people use head movement to help them perceive the location of the sound source. Another result they observed was the amount of processor resource required to spatialize one sound. They reported the processor was 90 percent utilized in computing the spatial sound. Wheeler and Ellinger suggested that processing multiple sounds would require more processing power and even multiple processors dedicated to computing sound spatialization.[WHEE93]

G. NASA AMES

Dr. Durand Begault and Dr. Beth Wenzel of NASA have done much work in the area of spatial sound. In 1993, NASA Ames developed the Ames Spatial Auditory Display (ASAD). This was the first 3D sound processor that could process multiple sounds at once. The ASAD was capable of placing up to five different sounds at fixed spatialized positions about a listener's head. Chief among the uses for the ASAD was its implementation in an emergency command, control and communications center. A single operator in such a center would have a difficult time distinguishing between multiple voices talking at the same time if all those voices were presented over the headphone in a monaural or stereo fashion. The ASAD could spatialize each one of those voices into different locations making each more intelligible. Also, because each of the voices were more intelligible, the operator was less fatigued in trying to interpret each of the voices. This technology has obvious advantages in emergency command, control and communications centers such as 911 operators and security personnel at large facilities that require constant communication. Also, air traffic controllers could find this useful in managing multiple aircraft and pilots. The ASAD was implemented using five separate communication channels, each connected to its own Motorola 56001 DSP. Each of the DSPs filtered the incoming sound using HRTFs and adapted FIR filters. All five resulting spatialized sounds were then sent to a common output headphone jack.[SALU93]

H. NETAUDIO3

In 1993, Mr. David Burgess of the Georgia Tech GVU Center began working on the Netaudio3 (NA3). NA3 is a networked audio server that allows multiple clients to control multiple independent audio sources in a shared auditory environment. The NA3 is a third generation outgrowth from the Mercator project discussed above. NA3's architecture allows audio processing tasks to be distributed in a shared memory or message passing MIMD parallel computer. The NA3 features sound effects such as pitch-bending, muffling/thinning, and non-linear distortion. The internal structure of the NA3 is based on the thread concept, allowing processing tasks to be distributed in parallel computers. The software architecture consists of three layers. The top layer is the programmer's interface. This layer allows the creating, controlling, querying and deleting of sounds. These sounds can be referred to by standard unit measures of sounds, namely hertz and decibels. Layer one uses a Remote Procedure Call (RPC) to allow the server to be controlled over a LAN. The second layer of the software architecture converts the programmer-provided sound units into raw signal processing parameters which are used to control the third layer. The third layer actually computes and processes the sound signals. Layer two has only a single thread whose job is to process RPC requests from layer one in a synchronous manner (first come-first serve). Layer three has as many threads as there are sounds in the environment. Layers two and three are loosely coupled in that communication between the two only occurs when the environment changes. In layer three, audio samples flow through pipelines of threads in high-bandwidth, synchronous channels.[BURG93]

Although there were problems noted by Mr. Burgess with this implementation of NA3 (most notable, a latency of several seconds before the sound would play), the server is a significant improvement over its predecessors in that it was able to play multiple sounds near simultaneously by distributing the workload over several processors using RPCs and thread concepts associated with modern distributed operating system principles.

I. SOUNDHACK

Soundhack is a program written by Mr. Tom Erbe at the California Institute of the Arts. Written for the Macintosh, Soundhack takes pre-existing sound files and, among other things, binaurally filters them and saves the output to a file. It was this program that gave NPSNET Sound researchers the idea of populating a virtual world environment with a number of discretely positioned sound files and then have a lookup table that would play the closest file to a sound event's position. Although this is less desirable as far as the accuracy of the placed sound goes, it does relieve the processor from the burden of real-time computation of spatialized sound so it can be devoted to graphics rendering. However, limitations were discovered with Mr. Erbe's program. Because it was written for the Macintosh, a Macintosh would have to be added into the NPSNET environment. Although this is not necessarily a limitation, the desire and goal of this research is to stay within the SGI environment present in NPSNET. Additionally, the goal of this research is to provide a spatial sound environment that is as realistic as possible. The experimental results from the 3D Acoustical Display at the University of Texas (described earlier) demonstrated that listeners were able to distinguish sounds at 15 degree intervals. To achieve realistic 3D sound in a non real-time, lookup table solution, sound files would have to be filtered for intervals of 15 degrees or less. Using even 10 degree intervals requires that thirty-six positioned sound files be generated to achieve 360 degree coverage. Moreover, a minimum of three elevation levels would be needed (below, even and above). Using these minimum standards, the total number of files for each sound in a virtual world would be 108. Using ten sound samples in a virtual world (although a more realistic number might be upwards of thirty), 1080 sound files would be required. Not only would this require a substantial amount of disk space to store the filtered sound files, it would require a substantial amount of time and effort to generate these files unless some type of background, automated process could be implemented. Soundhack took approximately ten minutes to filter one file for one position. We could not find a way to use Soundhack in the automated manner desired. However, Soundhack provides the ability to retrieve pre-recorded sound samples, filter them with HRTFs/FIR filters and then store them in a filtered format. We felt sure there were other programs available that would have the same functionality implemented in a UNIX environment to facilitate the background, automated processing requirements. In writing Mr. Erbe about this subject, he directed us to a program written specifically for SGI workstations called VSS.[ERBE94]

J. VSS

Virtual Sonic Space (VSS) was written by Mr. Rick Bidlack. It can take a sound source and compute its 3D image in a dynamic, real-time manner. The program also calculates and presents Doppler shift and distance perception filtering. It also has the ability to interpolate smoothly between FIR filter points so that moving sound sources sounded more realistic (as opposed to a choppy repositioning of the sound as it moved between FIR filter points). Written specifically for the SGI Indy and Indigo computers, it uses publicly available HRTFs and filters the sound through a pair of FIR filters. We were able to obtain a version of this program and test it with a great deal of success. However, in our testing, we noted the same limitations as were noted and documented in other research pursuits in this area. The real time filtering of a sound source required a majority of the processor's resources. Additionally, the program was only able to handle one sound source at a time. This was not sufficient for a robust, real-time virtual world simulation such as NPSNET that easily should accommodate as many as ten sounds simultaneously. It also did not have the capability to binaurally filter the sound source and save the output to a file. It did provide hope, however, that publicly available products do exist for the SGI environment in the area of sound spatialization.[BIDL94]

K. ACOUSTETRON II

Crystal River Engineering (CRE) has developed a hardware solution to headphone-delivered spatialized sound in the Acoustetron II. The system is a stand-alone audio server. The main workhorses of the Acoustetron II are four Motorola 56001 DSP 80 MIPS chips capable of spatializing up to twelve concurrent sound sources at a sampling rate of 44,100 Hz or twenty-four concurrent sound sources at a sampling rate of 22,050 Hz In its basic configuration, sound files are stored on the Acoustetron II sound server. The Acoustetron II is connected to a SGI workstation via an RS-232 serial interface. The SGI workstation sends specific parameters (which sound file to play, the sound event's location and the listener's location and orientation) to the Acoustetron II. The Acoustetron II filters the sound using HRTFs and corresponding FIR filters. The resulting sound is sent to an audio port which can be connected to headphones, nearphones or speakers. Although the Acoustetron II is state of the art in true, real-time 3D sound spatialization, it is not without its limitations. The two biggest limitations are that it is expensive and can only serve one workstation at a time. At approximately $10,000 per system, it is financially not feasible to purchase a system for each player in a multi-player virtual world simulation. However, with some experimentation and further research, it may be possible to use a single Acoustetron II to service several same location virtual world participants.

L. AUDIOWORKS2

Paradigm Simulation, Inc. has developed a commercial product called AudioWorks2. This program supports both open-field and headphone-delivered spatialized sound. Written for SGI workstations, it supports stand-alone audio processing on SGI workstations as well as an interface to CRE's Acoustetron II. In its stand-alone configuration, AudioWorks2 filters sound for stereo or quad-delivered spatialized sound. This means there are no elevation cues presented -- only XY-plane positioned sound cues. Connected to the Acoustetron II, AudioWorks2 takes advantage of the Acoustetron II's DSP-based filtering and lets the Acoustetron II handle the filtering and spatializing which delivers true 3D sound. As part of the application's package, Paradigm includes a powerful, high level C language application programming interface (API). This API allows a programmer to develop realistic spatialized sound including modeling Doppler shifts, propagation delays, and range attenuations. AudioWorks2 automatically recomputes coordinate and vector information when the listener re-orients himself and dynamically matches new spatialized sounds to the listener's new position. The application also takes advantage of multi-processor computers by allowing the programmer to assign specific sound rendering processes to specific processors or allow the application to automatically manage the computer's processor resources. Because AudioWorks2 only spatializes sound on one plane and relies on the Accoustetron II to deliver true 3D sound, it does not meet the goals of this thesis.

M. AUDIO IMAGE SOUND CUBE

Visual Synthesis Inc. (VSI) has developed a product called Audio Image SoundCube. The basis for this system is its digital Sampling Acquisition/Control System (SACS). SACS is an external module that is connected to an SGI workstation via a SCSI interface. As with the Acoustetron II, the SGI workstation sends specific parameters (which sound file to play, the sound event's location and the listener's location and orientation) to the SACS. The SACS does not use HRTFs to filter the sounds. Rather they use sophisticated sound sonification techniques. In trying to gain more information about their sonification techniques, VSI was reticent to give any specific information about their methods. They specifically did not want to discuss how their sonification techniques are different from traditional HRTF filtering.

Another product from VSI is the Audio Architect. Audio Architect is an advanced toolkit that uses existing SGI audio hardware to provide real-time audio development. Based on the same localization techniques as the Audio Image SoundCube, Audio Architect provides an alternative product to spatialize sound. However, because Audio Architect uses existing SGI audio hardware, only mono/stereo sound files are spatialized and presented in an XY-plane, much like Paradigm's AudioWorks2 product. A related VSI product is the Sonic Architect. Sonic Architect is a new product that is not yet marketed. However, preliminary reports say that Sonic Architect will be an application that takes advantage of existing hardware resources and use them to filter sound files to include elevation cues. It is not clear whether VSI will use HRTFs or is using a more sophisticated version of their sonification techniques used in Audio Architect. Additionally, VSI sells Vigra MMI-110 audio cards. These cards provide Indigo audio to SGI Onyx workstations. This subject will be included in the conclusions and recommendations chapter as a topic worthy of further research.

IV. CURRENT ENVIRONMENT

A. GENERAL

NPSNET is categorized as a multiple instruction stream, multiple data stream (MIMD) computer system. It is a collection of interconnected, independent workstations that do not share a common memory space. The NPSNET software can be generally described as a loosely coupled software system. Independent versions run on separate computers but interact with each other via DIS PDUs. If a participating workstation suffers a significant degree of failure, only the entity provided by that workstation to the interactive simulation is effected, not the entire system.

B. HARDWARE ENVIRONMENT

NPSNET runs in the NPS Graphics and Video Laboratory on SGI IRIS workstations. Different workstations have varying capabilities but all share a robust capability to compute and display graphics. Table 1 lists examples of the different kinds of workstations in the NPS graphics lab as well as capabilities for each. The SGI workstations

Workstation	Model	Processor	Graphics Card	Memory	Disk Space
Elvis	Onyx	4- R4400 150MHz	Reality Engine II	192 MB	4 GB
Meatloaf	Onyx	4- R4400 100MHz	Reality Engine II	256 MB	6 GB
Totally	Indigo 2 Extreme	1-R4000 100MHz	GU1 Extreme	64 MB	1 GB
Norman	Indigo Elan	1-R4000 100MHz	GR2 Elan	96 MB	1 GB
Rambo	Indy	1-R4000 100MHz	Indy 24 Bit	64 MB	1 GB

Table 1. NPS Graphics Lab Workstation Capabilities

are connected within the lab by an ethernet LAN. The networking architecture will be discussed in a later section. Complementing the graphical hardware is a suite of sound equipment designed to support open-field spatialized sound over six deliberately positioned loudspeakers in the laboratory. The sound support includes one EMAX II Digital Audio Sampler/Sequencer, one Apple MIDI Interface converter, one GL2 Allen and Heath Mixing board, two Ensoniq DP/4 Digital Signal Processors, one Ramsa Subwoofer Processor, two Ramsa Power Amplifiers, two Ramsa Subwoofers, two Ramsa Studio Monitors, one Carver Amplifier and two Infinity Speakers.

C. SOFTWARE ENVIRONMENT

NPSNET is implemented using C/C++ along with graphical design tools and libraries such as Performer and MultiGen. The current version, NPSNET-IV, was rewritten from its earlier version using an object-oriented paradigm. Although there is still a good amount of "legacy" code, vehicles and weapons are implemented as hierarchical classes to take advantage of the object-oriented feature of inheritance. For example, helicopters and jets are both vehicles in NPSNET that can fly. Both of these vehicles inherit characteristics of flying vehicles from its superclass such as taking off, flying, landing, and other attributes common to flying vehicles. However, they are specialized in their respective subclasses to give them their vehicle-specific attributes.

D. NETWORKING ARCHITECTURE

The physical network medium in which NPSNET is implemented in the graphics laboratory is ethernet. Because ethernet is capable of data transmission speeds up to 10 Mbps, it is a sufficient medium within the lab to support a relatively small number of participants. However, NPSNET is capable of wide area use, typically over the internet where T1 (1.5 Mbps) connections are common. With an increase in the user base over a wide area network (WAN) and a corresponding decrease in the available bandwidth (T1 connections), efficient data distribution schemes become increasingly important. A balance must be struck between data communications reliability and speed so that a real-time environment such as NPSNET meets its real-time requirements. A transport protocol such as Transmission Control Protocol (TCP) uses congestion control and is not well suited for a distributed, real-time application. While IP broadcast could be used on a LAN such as the ethernet LAN in the graphics lab, NPSNET could not use IP broadcast over a WAN because IP broadcast would distribute unnecessary data to every host on the WAN. This would be an expensive and unnecessary burden. Point-to-point communication on the other hand would require each NPSNET participant to maintain N*(N-1) virtual connections to every other player on the network. Every DIS packet sent would have to be sent to each virtual connection and would degrade network throughput performance to unacceptable levels, introducing too much latency into the real-time aspects of NPSNET. Researchers at NPS decided to use IP multicast which provides a one-to-many broadcast path. The idea behind IP multicast is that many users can belong to a group and only the data broadcasted between them will go to members in the group. This method of "selective broadcasting" provides for a happy medium between IP broadcasting and point-to-point communications. IP multicast uses the User Datagram Protocol (UDP). UDP is considered to be an unreliable, best effort delivery scheme for PDUs. In order to guarantee reliable delivery of data (as does TCP), each host would have to acknowledge each PDU received. This too would cause serious degradation in network performance, ultimately effecting the real-time nature of NPSNET. However, with NPSNET, guaranteed delivery of PDUs is not required. NPSNET uses a dead reckoning algorithm that updates a vehicle's position based on heading and speed data from the last ESPDU. This algorithm allows an entity within the virtual world to continue on its course of action without the benefit of constant updates from DIS packets. The algorithm uses the vehicle's heading and velocity information to "guess" where the vehicle's position will be and let it continue on its path. As new DIS PDUs are received, this heading and velocity information is updated and corrections are made to the vehicle's course and state. Because this significantly reduces the number of DIS PDUs required to maintain the real-time nature of an entity, network PDU traffic is significantly reduced. Moreover, if a DIS packet is lost due to a failure of UDP best effort services, the next DIS PDU received will be sufficient to update the vehicle's state. Therefore, an unreliable scheme such as UDP is sufficient. [MACE94][MACE95]

V. LOCALLY-DEVELOPED PURSUITS

A. GENERAL

As outlined in previous chapters, inserting realistic 3D audio in a virtual world is not an easy task. The main obstacle is the processor intensive requirement to synthesize spatial sound from monaural sound samples in a real-time manner. The original goal of this thesis was to identify a low-cost, locally developed implementation of headphone-delivered 3D sound. Three different approaches were studied -- rendering sound on the same workstation that is rendering the graphical representation of virtual entities, setting up a pre-positioned sound file library, and setting up a multiple client sound server. Research into each of these methods showed that none of them were viable. This chapter outlines those attempts and their shortcomings.

B. SAME WORKSTATION SOUND RENDERING

At first thought, rendering sound on the same workstation as is the virtual world player seemed to be the best and most obvious solution. Each workstation hosting a particular player would be responsible for generating the rendered spatial sound for that player. However, it was quickly discovered that the combined computational requirements to render 3D sound and real-time graphics made it impossible to accomplish both simultaneously on any of the workstations in the NPS graphics lab.

In order to have an effective 3D sound capability, any sound event within close proximity to a player's "hearing" must be rendered spatially and presented in less than 100 msecs to the player. One hundred msecs is the widely-published maximum latency threshold after which humans begin to disassociate instantaneous interactive control[DURL95]. Additionally, a workstation responsible for rendering graphics in a realistic, real-time manner must be capable of generating a frame rate of at least eight to ten frames per second (also a widely recognized minimum required threshold to present the illusion of continuous motion)[DURL95]. As an example, consider the following scenario.

A participant is flying a helicopter in NPSNET. He fires a rocket at a nearby vehicle and hits the vehicle. Examine each component of this scene and the demands on a workstation to present the sights and sounds for this event. First consider the graphical aspect. The workstation must receive and interpret the user input device (in this case, a joystick), receive and update other player's entity state, fire and explosion PDUs from the network, conduct the application processing required to implement the user input and PDUs, and render the graphical representation. Each of these stages plus the time it takes to synchronize each stage introduces lag into the virtual world simulation. VR lag is the sum of all of the various time delays and can be loosely defined as the total time between when a user performs an action and when the application presents the result of that action. The CPU requirements for each different model workstation capable of running NPSNET in the NPS graphics lab are presented in Table 2 (see Table 1 in the previous chapter for each workstation's capabilities and specifications).

Workstation	Frames per second	CPU Usage
Elvis	30	46.96%
Meatloaf	30	62.46%
Totally	20	94.36%

Table 2. NPS CPU Requirements

Now consider the sound aspects for the above scenario. For 3D sound rendering, the workstation takes the positional and orientation information from the received PDUs, loads the appropriate sound file for the given event (i.e., the helicopter engine sound, the missile firing sound and the subsequent detonation sound) and then must render the sound file to position it where reported. All of this sound processing must be accomplished within 100 msecs. The exception is in the case of the explosion sound if the source of the explosion is at a distance so that the speed of sound travel temporally places the sound outside the 100 msec threshold. In other words, since sound travels at a speed of 1100 feet per second, any sound outside a 110 foot radius would not have to meet the 100 msec threshold. But in the case where the player fires his weapon or changes the state of his vehicle that causes a change in the vehicle's sound, the 100 msec threshold applies.

A test was conducted to determine the CPU usage for real-time rendering of one sound and then again for two simultaneous sounds. Rick Bildlack's VSS program (described fully in Chapter III) was used to benchmark these tests. VSS was chosen because of its ability to render spatial sound in a real-time manner on SGI workstations. The CPU requirements for each test conducted on each of the workstations in the graphics lab are presented in Table 3.

Workstation	CPU Usage (one sound)	CPU Usage (two sounds)
Elvis	66.83%	81.85%
Totally	100%	100%

Table 3. VSS CPU Requirements

As demonstrated, the separate computational requirements on a workstation for graphics and sound rendering are heavy. Ideally, however, the goal is to perform graphics and sound rendering simultaneously. It follows that a workstation required to render both graphics and sound at the same time must meet the performance standards outlined above for each requirement. Several tests were run for each different workstation in the graphics lab capable of performing both requirements levied on the workstation at the same time. For each workstation tested, NPSNET and VSS was executed as seperate processes. VSS first rendered one then two sounds simultaneously. These tests proved to be overwhelming for each of the workstations. Graphics output suffered an average degradation in performance of 65% (frame rate). Spatial sound suffered on an average a 850 msec lag time, far exceeding the 100 msec threshold.

Unfortunately, it was not possible to conduct a test in which VSS and NPSNET were implemented as seperate threads in the same process. The source code was not available for VSS and an extensive rewrite of NPSNET code would have been necessary to include VSS in the process loop.

One alternative for the same workstation rendering approach is to install specialized audio hardware in a graphics workstation. Different companies are developing sound cards that support some measure of 3D sound production. Most of these cards are based on digital signal processing (DSP) chips and would relieve the workstation's main system resources by assuming the computational burden for 3D sound rendering. These sound cards will give some measure of spatialized sound but do not yet solve the problems of elevation and back/front reversal. However, they are a good low-cost solution if a high level of 3D sound fidelity is not required. But for the purposes of NPSNET's research goals, a high level of 3D sound fidelity is desired. Moreover, these cards were not available for testing. This is an area that could be explored further and will be outlined in the recommendations and conclusions chapter as an area recommended for further research.

It was obvious early on that rendering sound on the same workstation that is rendering graphics was not feasible given the current capabilities of workstations in the NPS Graphics Lab. A different approach was needed.

C. PRE-POSITIONED SPATIAL SOUND LIBRARY

Another alternative was to develop a library of pre-recorded spatial sounds. Providing a virtual world with a library of pre-positioned 3D sound cues was considered an overall inexpensive solution. If the virtual world were populated with enough discretely positioned sound cues, the replay of the closest sound file that matched the position of a sound event would be sufficient. A level of accuracy in 3D sound placement would be lost because only a discrete number of sound files could be recorded. Moreover, an average spatially positioned sound file is 100 KBytes in size. Depending on the variety of sounds that must be presented, it was thought that hard disk space would quickly became the limiting factor. However, a listener can determine the direction from which a sound comes to only within a fifteen degree range of accuracy[WHEE93]. Thus, a specific sound event can be spatialized and captured at fifteen degree intervals and provide 360 degree coverage. Additionally, a minimum of three different elevation levels would be needed to give the third positional dimension for spatial sound. One drawback to this approach is that it did not have the ability to interpolate smoothly between sound file points, a requirement if moving sound sources are to sound more realistic. This results in a choppy repositioning of the sound as it moves between sound event positions. However, this approach does lend itself to static sound events (no movement) such as weapons firing and detonations. Also, it was decided to use five degree intervals vice fifteen degree intervals in an attempt to increase the level of accuracy in 3D sound placement. NPSNET currently uses 30 different sounds as part of its environment. The following equation calculates the space requirement for the set of spatial sounds required for an NPSNET spatial sound library:

$$3levels \times \frac{360°}{level} \times \frac{1interval}{5°} \times \frac{1soundfile}{interval} \times \frac{100KBytes}{soundfile} \times \frac{1MByte}{1024KBytes} \times \frac{21.09MBytes}{SpatialSoundSet} \times 30SoundSets = 632.81MBytes$$

Eq 1

Although 632 MBytes is a good deal of space, the current price of hard drives does not make this requirement prohibitive.

In order to create the library of 3D sound files, a software application was needed that could take a monaural sound file sample and positional/orientation data, filter the sound file creating a positioned sample and save it to disk. According to the equation above, 216 separate pre-positioned sound files are needed to fully represent a sound event positioned in all the different specified locations. It would be tedious (although not impossible) to create each positioned sound file interactively (i.e., the user actively involved in creating each of the 216 sound files). Using Tom Erbe's Soundhack program (fully described in Chapter III), one monaural sound sample took approximately 10 minutes to filter and save as a positioned sound file. Soundhack did not have a way of scripting the process thereby creating the 216 sound files automatically. User interaction was required for every positioned file created. Creating 216 sound files using Soundhack would take 36 hours to complete for each representative sound event. NPSNET has 30 sounds which would require 1080 hours (45 days) to fully create a library of pre-positioned sound cues. Clearly, this was not a satisfactory approach. A way of creating these positioned sound files in a background, batch process was needed. It was discovered that of the commercially and publicly available applications capable of binaurally filtering monaural sound files, no product had the required ability to save a series of filtered sounds to separate output files. Further exploration of available applications in this area must continue and will be outlined in the recommendations and conclusions chapter as an area recommended for further research.

Another obstacle to overcome was the latency issue. The idea of the sound file library was that positional and orientation information would be presented in which a calculation would be performed to determine which sound file to present. The time it took to determine which sound file to present, retrieve the particular sound file from the disk and play it could not introduce too much latency between the virtual world sound event and the subsequent playing of the appropriate sound file. A small application was written which reported the time it took to lookup a positioned sound's filename, then load and play the file. The results of several time trials for each of the different workstations are presented in Table 4 .

Workstation	Time Trial #1 (msecs)	Time Trial #2 (msecs)	Time Trial #3 (msecs)	Time Trial #4 (msecs)
Annabelle	870	850	760	1030
Bond	830	890	830	790
Totally	760	790	810	770
Elvis	730	760	760	740

Table 4. Sound File Loading Times (in msecs).

From inspecting the results in Table 4, it is clear that the time it takes to lookup and load the sound file by far exceeds the 100 msec established as a maximum latency threshold. An overwhelming part of the time is devoted to the access and loading of the actual sound file from disk. One idea to overcome this load time obstacle was to pre-load sounds in memory at application start-up time. However, this idea was quickly discounted when it was realized that even to pre-load three sound events (engine sound, munitions firing sound and a detonation sound) would necessitate the loading of all 216 sound files for each sound event. At 100 KBytes per sound file, this would require approximately 21 megabytes of workstation memory per sound event -- 63 megabytes for the three basic sounds. Because the workstation uses a majority of its memory for graphics processing, requiring 63 megabytes of workstation memory for the sole purpose of implementing a sound file library was not desirable nor feasible. This latency issue became the limiting factor that made the pre-positioned sound file library alternative not feasible.

D. MULTIPLE CLIENT SOUND SERVER

Research into this alternative concentrated on implementing a RPC algorithm that would take advantage of a client-server relationship. The issues explored were the load on the network and the development of a suitable algorithm to efficiently render multiple real-time 3D sounds for multiple virtual world players. In its basic form, a client sends a RPC PDU to the sound server containing the player's identity, the sound to be played and positional/orientation information to accommodate the 3D sound rendering process. The sound server takes that information, renders the spatialized sound in real-time (no table lookup this time) and returns the resulting sound data to the client. The results obtained while researching this alternative were not promising. It turned out that several clients would each request different sound file renderings for the same sound event. For example, if four players in NPSNET are in close proximity to each other and an explosion occurs in the vicinity, each player would request a different rendering for the same sound event based on their position and orientation relative to the position of the explosion. The server would be asked to render four different spatialized sounds for the same sound event. The actual sound data (approximately 100 KBytes) for each rendered version of the same explosion would be sent back over the network to the requesting clients. If this scenario is taken one step further and each player generates two sounds -- an engine noise and a weapon firing noise, each client workstation would send 8 sound requests to the sound server (2 requests for its own sounds plus 2 requests for each of the other three player's sound events). The sound server would be required to process 32 different spatial sound requests near simultaneously. Although the scenario described would not be an unreasonable occurrence in NPSNET, (in fact, it would be a very likely occurrence), it is unreasonable to expect a single sound server to service that many requests at the same time. This assumption is based on the sound rendering tests described previously. Moreover, the network could not accommodate the load requirement to pass 32 sound files at 100 KBytes per file over the network to the requesting client and meet the 100 msec threshold limitations described previously. The following equation calculates the network bandwidth required for this scenario:

$$32SoundFiles \times \frac{100kb}{file} \times \frac{1000bytes}{kb} \times \frac{8bits}{byte} = \frac{25,600,000bits}{100m\sec s} \times \frac{1000m\sec s}{\sec s} = \frac{256mbits}{\sec s}$$

Eq 2

The network bandwidth requirement for the above scenario more than twice exceeds the capacity of fast ethernet (rated at 100 MBits/sec). Although the 256 MBits/sec network load would not be considered taxing for larger capacity fiber optic networks, the graphics lab is installed with fast ethernet and as such, this research was directed towards existing hardware capabilities. Moreover, the described scenario was a simple one. It is likely that even more simultaneous sound events would be presented. Table 5 outlines the network requirements for different combinations of number of players and sound events.

Number of Players	1 sound	2 sounds	3 sounds	4 sounds	5 sounds	6 sounds	7 sounds	8 sounds
1	8	16	24	32	40	48	56	64
2	32	128	96	128	160	192	224	256
3	72	144	216	288	360	432	504	576
4	128	256	384	512	640	768	896	1024
5	200	400	600	800	1000	1200	1400	1600
6	288	576	864	1152	1440	1728	2016	2304
7	392	784	1176	1568	1960	2352	2744	3136
8	512	1024	1536	2048	2560	3072	3584	4096

Table 5. Network Bandwidth Requirements (in MBits/sec).

The demands placed on the network and sound server increase exponentially as more players and simultaneous sound events are added.

Additionally, an algorithm is required that can efficiently prioritize client requests and render the appropriate 3D sound. Applications such as Soundhack and VSS (discussed earlier) show promise but have limitations. Attempts to use the source code for each of these has been unsuccessful. With Soundhack, Mr. Erbe said that the issue was not obtaining the source code but rather it would have to be rewritten for the SGI environment.

Because Soundhack was written for the Macintosh environment, Mr. Erbe used Macintosh-specific development libraries. In a recent e-mail, Mr. Erbe stated "porting Soundhack to UNIX would be a monumental task as I do not use any ANSI calls but only Macintosh specific calls (not a single malloc or printf in 30,000 lines of code!)[ERBE96]." Even if it were feasible to adapt the code for Soundhack or VSS, the file loading latency issues described above would still need to be addressed. Moreover, no other publicly available software applications have yet been found that accomplish the kind of 3D sound rendering needed for this research.

In general, required thresholds for network load and performance levels must be met for a multiple client sound server to be a viable option. Investigation into this alternative showed that this approach was not feasible.

E. SUMMARY

Specific thesis research into the above and other alternatives is ongoing. There are many academic, government and commercial organizations that are pursuing virtual environment technologies. Because 3D sound is a recognized and achievable goal in virtual world applications, much effort is being expended in this area. The three alternatives investigated as part of this thesis research clearly illustrates that technology is not yet robust enough to support the real-time rendering of multiple sound events in a virtual world application. Rendering sound on the same workstation that is rendering the graphical representation of virtual entities overwhelms system resources. Setting up a pre-positioned sound file library shows promise but introduces too much latency into the replay of acoustic sound cues. Multiple client sound servers overwhelm network and processor capabilities. However, it is only a matter of time before advances in processing performance are to a level that will satisfy sound rendering requirements. Generally speaking though, as more alternatives are investigated, it is clear that locally developed solutions are computationally expensive and do not easily lend themselves to efficient real-time rendering of multiple audio sources in a dynamic virtual environment.

VI. SINGLE CLIENT SOUND SERVER

A. GENERAL

The original goal of this thesis was to find a method of locally developing a headphone-delivered 3D sound solution for NPSNET. As discussed in the previous chapter, sound servers provide an attractive alternative for 3D sound rendering because they relieve the client of the computational expense of binaurally filtering sound samples. A sound server that services multiple clients is not yet available while single client sound servers exist and work well. Crystal River Engineering's Acoustetron II is a commercially available sound server which is particularly well suited for NPSNET. The unfavorable aspect of this approach is that the Acoustetron II is a known commercial solution for rendering 3D sound. This strays from one of the original goals of this thesis -- a locally developed, low cost solution. However, any "in-house" 3D sound solutions would have to have been robust enough to meet the auditory expectations of a virtual world user. While conducting this research, a low cost alternative could not be found given the current inventory of equipment and capabilities in the NPS graphics lab. Further, the integration of the Acoustetron II into the NPSNET environment was not trivial and worthy of some discussion. Ultimately, the integration of the Acoustetron II met the primary goal of this thesis -- to provide a headphone-delivered 3D sound capability to NPSNET.

B. BACKGROUND

The Acoustetron II is an *AudioReality*™ sound server. *AudioReality*™ is a term created and trademarked by CRE to describe their audio spatialization techniques. The Acoustetron II adds a full spectrum of 3D sound, including Doppler shifts, spatialization, and acoustic raytracing of rooms and environments to high-end graphics workstations, such as the ones used in the NPS graphics lab. CRE was founded in 1987 by Scott Foster and its initial work was funded by NASA. An early innovator in the field of virtual reality, the company's products enable realistic 3D acoustic rendering on personal computers and workstations. CRE's first product was the NASA-commissioned Convolvotron, the world's first real-time 3D sound simulator (discussed in chapter 3). Since then, CRE's products have become standard equipment in many psychoacoustic research labs, million dollar flight and driving simulators, and high-end virtual reality environments.[CRYS96a]

Naval Research Labs, Naval Air (NAVAIR) recently granted CRE funding for Phase II Small Business Initiative Research (SBIR) to develop methods for improving 3D acoustic rendering. The primary emphasis of this study is

- modeling ground reflection to increase the accuracy of 3D localization, particularly elevation cues.

- modeling Doppler shift to convey an accurate sense of motion in dynamic systems.

- customizing HRTFs for individual listeners.

- creating a scalable architecture and applications programmer interface to more efficiently utilize the underlying hardware resources.

- investigating more efficient algorithms for spatializing audio.[DARK95]

C. HARDWARE

The Acoustetron II is a stand-alone, single client, 3D sound server that is controlled via a communication line by a workstation client. NPSNET's Acoustetron II uses an RS-232 serial connection as its default communications link. An ethernet communications link is also an available option. The Acoustetron II is an Intel-based 486DX4 PC with four DSP cards installed to accomplish the 3D sound rendering. Each DSP card holds a Motorola DSP56001 chip clocked at 40MHz and high resolution stereo analog-to-digital and digital-to-analog converters with input and output sampling rates of up to 44,100 samples per second [CRYS96b]. Each of the DSP cards in turn sends their processed digital sound samples to a Turtle Beach MultiSound Tahiti sound card. Connected to the output channel of the sound card is a Symetrix SX204 Headphone Amplifier. The SX204 is a 1-in 4-out amplifier designed to drive multiple headphones or PC speaker configurations. Connected to the SX204 is a Cambridge *SoundWorks*™ PC speaker system as well as a pair of Sennheiser HD 540 Reference II Headphones. Figure 9shows the Acoustetron II configuration.

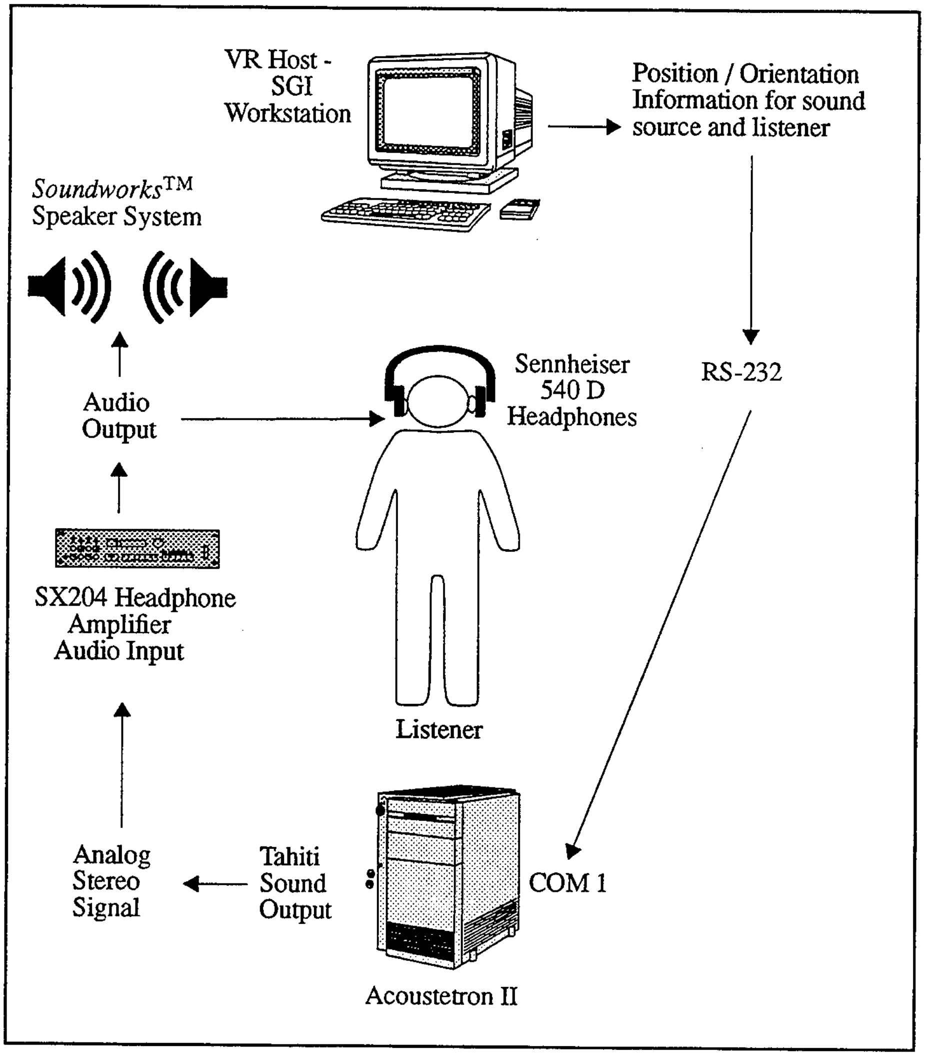

Figure 8. Overview of Acoustetron II 3D Sound Server.

The workstation client sends information such as audio source and listener positions to the Acoustetron II via RS-232. The Acoustetron II continually computes source, listener, and surface refections and velocities, and renders up to 24 separate spatialized sound sources accordingly. The audio output can be presented over headphones, nearphones, or speakers. Sounds can originate from digitized sound samples (Microsoft RIFF wave files) or external live inputs such as CD tracks or microphones. The sounds are processed at a rate of 44,100 Hz, 16-bit samples per second (CD quality) for 12 simultaneous sources or 22,050 Hz, 16-bit samples per second for 24 simultaneous sources. An ANSI C function interface allows for fast, high-level development of 3D sound spaces and integration of 3D sound into existing virtual environments such as NPSNET. At an update rate of 44 MHz, sounds are rendered at their exact position and orientation in space as perceived by the listener and appear to move seamlessly in the virtual environment.[CRYS96b]

D. SOFTWARE

The spatialization software included with the Acoustetron II comprises both a software library and several demo programs. The library routines provide automatic detection of the Acoustetron II sound server and translate high-level commands describing source and listener positioning into the low-level format needed by the system. CRE_TRON is the name of the library.

E. IMPLEMENTATION

1. Approach

To integrate the Acoustetron II into NPSNET, modifications to the applicable source code routines for NPSNET sound were required. NPSNET sound had been accomplished in three ways -- Russell Storms' NPS-3DSS, Paul Barham's NPS-MONO and direct calls to the sound libraries from within NPSNET itself. This last method replays mono sound samples on the same workstation as is rendering the graphics for the virtual world simulation providing the workstation is capable of sound replay. However, this method did not address in any way rendering of 3D sound. At first, the best approach seemed to be to modify NPSNET source code so the Acoustetron II would be available directly from within NPSNET. The user would determine which alternative to use

(Acoustetron II or direct sound replay) at NPSNET start-up time. A command line option would be added to NPSNET start-up routines that would designate the particular method of sound delivery. However, as this option was explored further, it was realized that this would constrain the Acoustetron II to be connected to the same workstation as was running NPSNET. Because the Acoustetron II requires a serial port connection, it follows that a serial port would have to be available on the client workstation. Some of the candidate workstations (such as Elvis and Gravy 5) did not have an available serial port because other peripherals were already using those resources. Also, the graphical display device (computer display, TV or HMD) for NPSNET was not necessarily co-located with the workstation rendering the graphics. For example, the three screen TV setup in the NPS graphics lab displays the version of NPSNET running on the workstation Meatloaf. Meatloaf is located several feet away from the three screen TV setup. It was not feasible nor desired to connect the Acoustetron II to Meatloaf and then run the headphone cable across the walking area to where the user would sit in front of the three screen TVs to interact with NPSNET. To maintain the most amount of flexibility, it was decided to integrate the Acoustetron II so that it was workstation independent.

The software written to interface with the Acoustetron II is able to run from any workstation and look to any other NPSNET participating workstation as its master. This was the same approach taken by two of the other current sound implementations for NPSNET -- NPS-3DSS and NPS-MONO. In the example of the three screen TVs given above, the current implementation of the Acoustetron II interface and connection of the Acoustetron II can be run from the workstation Rambo which sits in front of the three screen TV. This allows the Acoustetron II to deliver 3D sound in a convenient manner. The only drawback to this approach is the network latency introduced while waiting for DIS ESPDUs to arrive from the master. This latency issue will be addressed in more detail later in this chapter.

2. Source Code

NPS-MONO was configured to monitor network traffic from a designated master and replay sound files based on DIS PDU-supplied information. It was decided to reuse the source code from NPS-MONO and adapt it to address the Acoustetron II. The adaptation of Paul Barham's code is called NPS-ACOUST. The main differences between the two implementations is the information required when requesting the replay of a sound file and the number of sound events that are tracked and presented. The NPS-MONO methods require the identification of the sound event that occurred and the location of both the sound event and the listener for each sound event called. Software calculations are made based on this information and sound is replayed with proper distance and loudness cues. The Acoustetron II also needs the sound event data but only needs the location of the sound event, not the listener's position. The position and orientation of the listener is updated as a separate function call to the Acoustetron II. Additionally, distance and loudness cues, as well as spatial rendering, are calculated on the Acoustetron II's DSP cards for specific sound events relieving the client of those expensive software calculations.

NPS-ACOUST also goes further in addressing sound event presentation. NPS-MONO only presents sound events particular to the master's vehicle. NPS-ACOUST not only addresses more completely the master's vehicle sounds (such as the continuous sound of the vehicle's engine noise) but replays other vehicle engine and weapons noises as well. All sound events are presented spatially. Also, Doppler shift is added to give a more realistic presentation of moving vehicles. Doppler shift is a very effective sound cue in presenting the illusion of 3D sound motion associated with a virtual vehicle. The addition of other vehicle sounds presents a more realistic acoustic portrait of an environment which further immerses a player in the virtual world of NPSNET. In short, the new functionality in NPS-ACOUST represents a significant advance in NPSNET sound presentation.

At start-up, NPS-ACOUST is told which workstation to consider as the master. ESPDUs are received from the master at which time the entity type and location information is determined. The Acoustetron II then renders sound based on continual updates to the entity's state information (location, orientation, speed, etc.). This approach mandated a program that would monitor DIS PDU traffic on the LAN and glean salient information about the NPSNET environment. NPS-ACOUST not only monitors ESPDUs from the master (a feature inherited from NPS-MONO) but environmental PDUs like detonations and fires as well. Also, the interface monitors the activities of other entities and presents vehicle sounds for those vehicles as well if they are within hearing range.

3. Network Monitoring Routines

One idea that was investigated but eventually discarded was to have a separate process continually monitoring the master's entity state information and another process monitor environmental sound events such as other vehicles, detonations and weapons firing. The motivation behind this idea was that a process devoted to servicing only the master's entity state information could more quickly and efficiently present the data to the Acoustetron II for 3D sound rendering. The main program used the ANSI C sproc() function to create this monitoring process. However, as the separate process began to issue commands to the Acoustetron II, resource contention problems were created with the serial port and the Acoustetron II. Because both processes were sending commands to the Acoustetron II via a common serial port, command collisions were occurring causing the Acoustetron II to malfunction. Semaphores were considered as a remedy but then discarded when it was realized that locking the serial port while it was busy would introduce too much latency into the real-time requirements for sound event presentation. A command would have to wait for the release of the lock on the serial port before it could be sent to the Acoustetron II for processing. The idea of separate processes was eventually discarded in favor of managing all calls to the Acoustetron II in a single process loop.

4. Command Line Options

Because NPS-ACOUST descended from NPS-MONO, all of the usual NPSNET command line options are available. One command line option that is specific to NPS-ACOUST is the datafile used. NPS-ACOUST must use its own datafile to populate the program with the available sound filenames on the Acoustetron II. This datafile is called "acoustetron.dat" and is addressed using the command line option "-DATAFILE datafiles/ acoustetron.dat".

The datafile that is used by NPS-ACOUST is formatted differently than that of NPS-MONO. The "acoustetron.dat" datafile contains a list of all potential sounds that could be called while servicing a client running NPSNET. NPS-MONO lists all the sound files that will be pre-loaded into workstation memory for replay. Another difference is the float value that follows each sound filename listed in both datafiles. In NPS-MONO, the float value is used as the clipping distance. The Acoustetron II computes the clipping distance based on the reported position. Instead, the float value reported in the acoustetron.dat file is used to set the initial decibel levels for each sound.

5. Listener's Head Orientation Constraints

One constraint in NPS-ACOUST is the orientation of the listener's head. Specifically, the listener's head position and orientation must be constrained to that of the master's virtual vehicle. Presenting 3D spatial sound over a set of headphones assumes that the listener's head orientation is consistent with that reported to the Acoustetron II. Without headtracking capabilities, the assumption is the listener is looking straight ahead. If the listener turns his head away from the screen, a sound event cannot be delivered correctly relative to its virtual environment placement. For example, if a sound event occurs to the left of a player in the virtual world simulation and the listener turns his head, the headphones turn as well and the sound is still heard to the listener's left in reference to the orientation of the head. Headtracking capabilities that report head orientation are needed to overcome this limitation. This limitation comes into play in NPS-ACOUST because ESPDUs received for a particular vehicle do not contain pilot/driver location and orientation data. Rather, the ESPDU reports the location and orientation of the vehicle only. There is a small caveat to this statement because some DIS-standard vehicles are articulated. For example, a tank can still be oriented in a north-south posture but turn its turret to an east-west posture. Both sets of posture data are presented in the tank's ESPDU. However, as is the case with non-articulated vehicles (such as jets and helicopters), the information about the driver's location and orientation is still not available. In the example of the tank above, the driver could be looking out of a side porthole in the turret making his head orientation different from the turret's orientation. In any case, with the current DIS standard, it is impossible to tell the location and orientation of a pilot/driver's head within a virtual vehicle. Because the orientation and location of the listener's head is crucial data for the Acoustetron II, it must be assumed that the head is co-located and co-oriented with the vehicle.

This constraint is not as severe as it seems. A listener's head orientation only becomes an issue in virtual display setups where displays are on sides other than directly in front of a user and the user is wearing headphones. Usually, a listener will be participating in NPSNET viewing the graphical display of NPSNET on a computer monitor. The player's view is constrained to that of the vehicle. In order for a player to see an event that caused a sound to his left, he would have to re-orient the vehicle's view towards that sound. In this case, the position of the player as reported to the Acoustetron II is the same as the vehicle. It would make little sense for a player to look away from the monitor towards a sound event without re-orienting the viewport of the monitor as well. (As an aside, were the user to look away from the monitor as a result of a 3D auditory cue, it could be considered a small victory for the effectiveness of the presented spatial audio).

There are two examples when a player's head orientation is important. The first example is the CAVE project at the Electronic Visualization Laboratory at the University of Illinois at Chicago. The CAVE is a multi-person, room-sized, high-resolution, 3D video and audio environment. The room is constructed of large screens on which graphics are projected onto two or three walls and/or the floor which allows the graphical display of the virtual environment to surround the viewer. As a viewer wearing a location sensor and lightweight stereo glasses moves within its display boundaries, the correct perspective and stereo projections of the environment are updated, and the image moves with and surrounds the viewer[NCSA96]. The focus on the audio presentation for this project is not on headphones but on loudspeakers. The sound is presented spatially over loudspeakers independent of the listener's head location and orientation. If a sound event occurs to the listener's left, this time when he turns his head, the sound does not translate with his head turn. In other words, the listener is free to move his head about without effecting the position of the sound.

The other instance where a listener's head orientation is important is when spatial sound cues are used in conjunction with a HMD. In this case, the user re-orients his head and is displayed a different graphical view of the virtual environment. This is a simple case to handle for Acoustetron II implementations. The Acoustetron II comes with a software interface for devices such as the Polhemus Head Tracking System. All that is needed is to glean whatever orientation data the headtracker is reporting and supply that to the Acoustetron II. The Acoustetron II in turn takes care of re-rendering the sound for a re-oriented head.

6. Vehicle Engine Noises

One of the significant advances offered to NPSNET by NPS-ACOUST is the ability to continuously play multiple vehicle engine sounds. Additionally, Doppler shift as well as engine pitch variance is possible with the Acoustetron II and are important sound cues in conveying vehicle movement and velocity. Pitch variance is especially important. The faster the virtual vehicle travels, the harder the virtual engine must work and the higher the virtual engine pitch must sound. The Acoustetron II allows the ability to easily vary the pitch of a playing sound. However, the only indication that the engine's sound pitch must be varied is the reported speed of the vehicle. Unfortunately, NPSNET does not send out an update ESPDU when the vehicle's speed changes. NPS-ACOUST must wait for the "heartbeat" ESPDU from the master to determine any changes in the vehicle speed and make the corresponding engine pitch changes.

Additionally, a bug was discovered in the software of the Acoustetron II when implementing the vehicle engine sounds. By convention, the vehicle engine sound is reported to the Acoustetron II as being at the same location and orientation as the vehicle. As discussed earlier, the listener's head location is constrained to the same location and orientation of the vehicle as well. It follows that if the engine sound and listener's head are co-located and co-oriented, the engine sound's spatial placement should remain consistent. This was not the case. The symptom was that as the vehicle changed its yaw either for the negative or positive, the engine sound presented by the Acoustetron II convolved (or changed its spatial properties). This was not an intended or desired effect. The sound should not have changed at all because as a position update was received from a master ESPDU, the location/orientation information was provided to the Acoustetron II for both the head location and the engine sound sample. After trying to remedy this problem for a good deal of time, CRE technical support was called at which time it was verified that this was a known bug and was being addressing by CRE. Mr. Paul Sparling, the technical support representative, suggested moving the location of the engine sound a small distance away from the head so that the sound and head were not co-located. This did not solve the problem and will be documented in the conclusions and recommendations chapter as a topic for further research.

7. Acoustetron Update Cycles

It is important to update the location and orientation of the listener's head at every opportunity. Because the head location data is gathered from master vehicle ESPDUs, every ESPDU received was used to update the head location and orientation and reported to the Acoustetron II. However, because a master has the potential to only send out a "heartbeat" ESPDU every five seconds, a dead reckoning algorithm was used to move the master vehicle based on heading and velocity in absence of updates to its state. The vehicle was dead reckon moved and the resulting new location and orientation data was used to update the head location for the Acoustetron II. However, it was not enough to simply update the head location. There is a function in the CRE_TRON library called cre_update_audio() that must be called when head location and orientation changes are made. Spatial rendering in reference to a new head location and orientation is not accomplished until cre_update_audio() is called. CRE recommends that a call to this function be made to coincide with every presented graphical frame as part of a virtual applications's graphical rendering loop. However, because NPS-ACOUST is a separate program from NPSNET, it was impossible to synchronize calls to the CRE function with the graphics loop in NPSNET. Instead, the function call was made at every iteration of the network monitoring loop in NPS-ACOUST. Either an update ESPDU was received from the master updating the location data for the master's vehicle or a call to the dead reckoning algorithm was made. In either case, location data was updated, reported to the Acoustetron II and a call to cre_update_audio() made. There were no perceived latency or synchronization issues between the ESPDUs received from the master and the update cycle of NPS-ACOUST because of the dead reckoning algorithm used in NPS-ACOUST. The issue of network latency will be discussed later in this chapter.

8. Gain

Attenuation of a sound over a distance is a very important 3D sound cue. "Gain" is the amplification or attenuation of sound over distance measured in decibels (dB). 0 dB represents no amplification and no attenuation. A positive dB value amplifies a sound while a negative value attenuates it. As a sound source gets closer to a listener, its sound pressure level increases exponentially. However, there is a maximum volume that audio hardware can reach. The Acoustetron II sets a maximum volume to be reached for a 0 dB sound source to be at 2.5 units of measurement from the listener. This means that a 0.0 dB sound is replayed at its maximum recorded level at 2.5 units or closer and exponentially attenuates as the distance increases. For the purposes of NPSNET, most sounds (detonations and other vehicle sounds) are played at a significant distance away from the listener. As a result, the gain for these sound events is substantially increased in NPS-ACOUST, in some cases as high as 60 dB. There are two major factors that come into play in the attentuation of distant sound sources -- Atmospheric Absorption and Spreading Loss Roll-Off.

9. Atmospheric Absorption

The Acoustetron II takes into account the effects of atmosphere by attenuating the higher pitches of a sound at a higher rate than the lower pitches. The amount that is attenuated depends on the distance of the sound -- the greater the distance, the more attenuation of the higher frequencies. The result is the familiar low rumbling effect for distant sounds. Thunder is a good example to illustrate this point. Thunder is sound produced when a flash of lightning passes through air. Thunder at a distance has a rumbling quality to it while thunder that occurs nearby sounds very crisp. This a good example of the effect of atmospheric absorption on a traveling distant sound.

10. Spreading Loss Roll-Off

As sound waves travel outward from the location of the sound event, the power (or pressure level) of the sound dissipates over an increasing spherical area. This is called "spreading loss roll-off." Spreading loss roll-off is a factor that is used to help determine at what minimum distance a sound begins to attenuate. This loss of sound power is mathematically modeled in Equation 3.

$$\text{Clipping Distance} = \left(\text{Gain Ratio} \times 10^{\frac{\text{dB Gain}}{20.0}}\right)^{\frac{1}{\text{Spreading Rolloff}}} \qquad \text{Eq 3}$$

The distance model applies a relative attenuation to the sound source on the dynamic range gain (Gain Ratio in dB) multiplied by the ear to sound source distance raised to the power of the inverse of the spreading loss roll-off exponent. The result of this relationship is that, at some small distance (the clip distance), the distance model attenuation goes to zero (the DSP filters the source signal at full input level) and shorter distances have no gain amplification cues. The gain ratio is set at a value that optimizes dynamic range versus near-field effects[CRYS96b]. Given a gain ratio = 2.1dB and a spreading loss roll-off factor = 0.80 (both recommended by CRE), Table 6 gives clipping distances in terms of dB. In layman's terms, a sound presented at a given decibel range will sound no louder than its maximum volume within the clipping distance radius. For example, in Table 6, a sound presented at 20.0 decibels will be at its maximum volume at a distance of 45.0 units and closer. Any distances greater than 45.0 units will suffer some amount of attenuation.

Gain (in dB)	Clipping Distance (in units)
-20.0	0.1
-10.0	0.6
-5.0	1.3
0.0	2.5
5.0	5.2
10.0	10.7
20.0	45.0

Table 6. NPSNET Sound Clipping Distances

11. Speed of Sound

A related issue to sound attenuation is the speed at which sound travels. When a sound event occurs at a distance, there is a delay between the time the sound event occurs and when it is heard by a listener. It was thought that the Acoustetron II would take into account the distance of the sound and when given the command to play a sound, pause for the appropriate amount of time it would take for the sound to travel to the listener's position. However, this was not the case. The Acoustetron II played the sound immediately when commanded only adding in the appropriate attenuation and absorption cues. In order to model accurate distance cues, the Acoustetron II command to play a sound had to be delayed based on the distance and the speed in which sound travels. Fortunately, NPS-MONO already had functionality implemented that took into account the speed and delay of sound travel. With slight modifications, this functionality was applied to the Acoustetron II. Sound play commands are only issued to the Acoustetron II after an appropriate delay to take into account the speed and distance a sound event in NPSNET must travel before it reaches the listener.

12. Latency

The implementation of the speed of sound functionality unexpectedly benefited the NPS-ACOUST in another way. As discussed earlier, the approach taken to implement NPS-ACOUST introduced some amount of latency in replaying appropriate sound events in the virtual environment. This latency was superseded by the amount of time required for sounds to travel to the listener. Most sound events in NPSNET occurred at a reasonable distance away from the listener's position. The only exceptions were the listener's own vehicle engine sound and weapons firing sounds. The vehicle engine sound is played in a continuous loop thus side-stepping any concerns about introduced latency. It was decided to pre-load and keep ready the vehicle's weapon firing sounds so that there was no latency involved in loading the sound before it could be rendered and presented. The result was an acoustic environment for NPSNET that was appropriate in its delayed presentation of distant spatial sound cues. In other words, the introduced latency discussed earlier became a non-issue.

13. Units of Measurement

After implementing the speed of sound functionality, the important issue of units of measurement was discovered. It was assumed that the NPSNET unit of measurement was meters. The Acoustetron II requires to be notified what the units of measurement are so that it can properly render a sound (i.e. a sound 1000 inches away from a listener is presented much louder than a sound 1000 meters way). After getting the speed of sound functionality compatible with Acoustetron II calls, distant sound events still did not sound "right." Events that were visually placed only a few meters away sounded like they were much further away. It was initially thought that decibel levels for individual sounds needed to be adjusted on a case-by-case basis. But as this was done, not much progress was made at matching up an appropriate sound volume level with its distance placement. Finally, it was realized that NPSNET was reporting its measurements in feet and the Acoustetron II had been set to expect meters. An explosion that was reported in NPSNET to be 1000 feet from the observer was being played by the Acoustetron II at 1000 meters -- presented at approximately three times the intended distance. Once the Acoustetron II was reset to expect feet, the sound cues were much more appropriate.

14. World Coordinate System

As NPS-ACOUST was developed and the challenges discussed earlier were overcome, the interface to the Acoustetron II was better able to interpret ESPDUs, load appropriate sounds and replay them spatially. However, it was at this point that one of the most perplexing problems arose in this implementation. Occasionally, sounds were not being spatially placed correctly in reference to the reported orientation of the vehicle/ listener's head. Sometimes it worked -- sometimes it did not. The sounds were correctly placed by the Acoustetron II only when one orientation parameter was changed (yaw, pitch, or roll). However, when two or all three of the orientation parameters were changed in combination (yaw and roll for example), the sounds would not be placed correctly. Because the Acoustetron II specified orientation in right-handed radian Euler rotations, it was thought that a singularity was being encountered thus causing incorrect spatial sound calculations. But this was quickly discarded because singularities in Euler rotations are encountered at combinations of ninety degree changes. The orientation changes that caused this problem were much less than ninety degrees. It was finally discovered that the Acoustetron II was using a different coordinate system than was NPSNET. The coordinate system is all important to the calculations performed by the Acoustetron II in spatially presented sound. Location and orientation data from NPSNET was being fed directly to the Acoustetron II without appropriate coordinate system transformations. Figure 9 and Figure 10 show the coordinate systems for both environments.

A matrix transformation was considered to translate NPSNET coordinates into Acoustetron II coordinates. But again, the problem of introduced latency was considered and a simpler solution was found. By studying the diagrams of the two coordinate systems, it appeared that the only difference was a rotation about the Z-axis of ninety degrees. It was decided to simply add pi halve (1.5708) radians to the yaw reported by NPSNET. This had

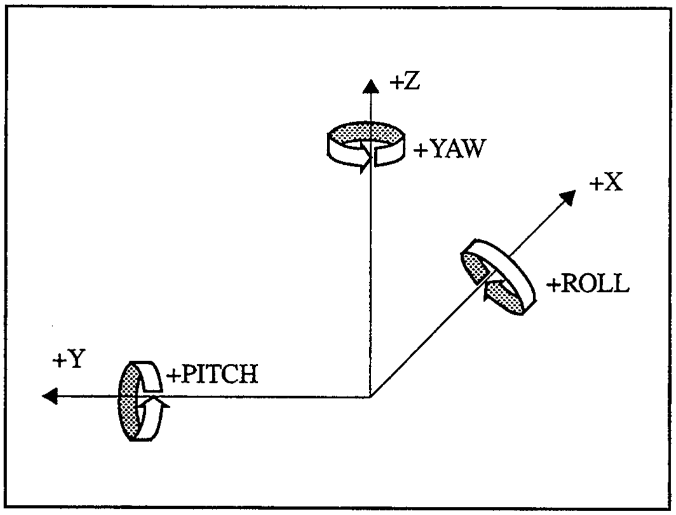

Figure 9. Acoustetron II Coordinate System.

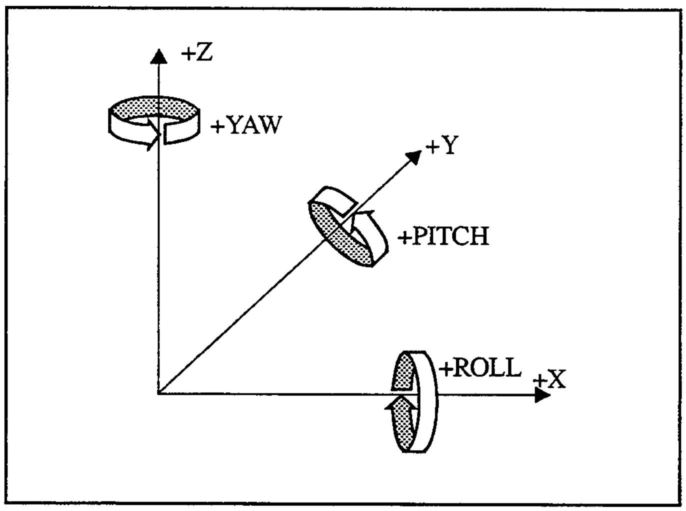

Figure 10. NPSNET Coordinate System.

the effect of translating the heading appropriately so that the Acoustetron II was able to replay sounds correctly in reference to the reported orientation of the vehicle/listener's head.

15. Acoustetron II Resource Management

Depending on the desired sampling rate (44.1 or 22.05 MHz), the Acoustetron II is capable of either 12 or 24 simultaneous sound channel manipulations (respectively).

Although this represents a significant increase in the capability of NPSNET sound, Acoustetron II sound channel resources are still limited and must be managed. It is impossible to predict which sounds will be required due to the dynamic and unpredictable nature of an interactive multi-player virtual simulation. Players can join, leave and rejoin the simulation at will, often inserting different virtual vehicles. It was decided to reserve only the channels needed for the master's vehicle sounds and dynamically load and unload sounds into the remaining Acoustetron II channels. The first decision was to keep track of which Acoustetron II channels were allocated from within NPS-ACOUST. This would reduce the communications latency of making expensive calls via the RS-232 serial line to the Acoustetron II. However, knowing which channel had been assigned did not fully solve the problem of resource monitoring. A channel was considered "assigned" when a sound was loaded to it and then played until the sound sample was complete. Because each sound sample varies in replay length, it was difficult to determine when the sound was finished playing and thus able to release the channel for other sounds to be assigned. A function called cre_get_sources_playing() was found that issues a one-time call to the Acoustetron II and returns a list of the channels and the status of the sound loaded (playing or not playing). After implementing this function, it was relatively easy to manage the sound channel resources on the Acoustetron II. In general, NPS-ACOUST reserves three or four channels on the Acoustetron II for the master vehicle and leaves the remaining twenty or so channels for dynamic sound event insertion.

16. Product Verification

CRE's proprietary algorithms and filters have been psychoacoustically verified to reproduce signals that closely match the ones perceived by a human listener in the real world. Multiple sounds are capable of moving dynamically throughout the entire 3D space surrounding a listener. An experiment was devised to verify that the Acoustetron II was able to deliver 3D sound as advertised. Two areas were addressed:

- placement of individual sounds anywhere in the 3D space surrounding a listener.

- realistic presentation of sound movement including Doppler shifts and sound source and listener motion.

The experiment was set up so that listeners were subjected to a random set of positioned sounds, both stationary and moving. Because the NPS graphics lab does not have the equipment needed to set up an accurate experiment of sound placement perceptions, a crude experiment was developed. The listener was required to report the location of presented stationary sounds and the location and perceived movement of moving sound sources. The listener was asked to point to the direction from which a sound was coming and also to follow the movement of a sound as it moved about his position in NPSNET. Although the level of accuracy in measuring the listener's perceived sound placements left much to be desired, the empirical results gained from this experiment did verify that sounds were being placed in the NPSNET environment and replayed very close to their intended spatial placements. Because the HRTFs used in the Acoustetron II are generic and publicly available, some listeners in this experiment experienced the back-front reversal confusion discussed in Chapter III. However in all cases, listeners were able to place the positioned sounds to within fifteen degrees.

F. CONCLUSION

The result of this implementation is a single client sound server capable of presenting up to 24 simultaneous, spatially rendered sound cues. NPS-ACOUST is written to provide the interface for the Acoustetron II to NPSNET sound events. The additional 3D sound capabilities introduced to NPSNET significantly improve the capability to immerse a player in the virtual world simulation. There are many more NPSNET 3D sound possibilities that can be realized using the Acoustetron II. These possibilities will be discussed in the recommendations and conclusions chapter.

G. ADDITIONAL CAPABILITIES WITH CRE PRODUCTS

CRE has a product called *AudioReality*™ Room Simulation (RS). *AudioReality*™ RS represents the next major technology improvement in interactive 3D sound systems. The software combines proprietary *AudioReality*™ 3D sound algorithms and audio room simulation methods to reproduce the complete acoustics of a virtual environment. 3D sound systems aim to recreate sound sources and listeners in a 3D space. The *AudioReality*™ RS technology provides the additional ability to place passive acoustic objects, such as sound reflecting walls and surfaces, in such a space. Materials from a palette including wood, marble, carpet, or glass are applied to the surfaces to model the amount of sound that reflects off a surface or transmits through it. The result is an immersive sound space in which listeners, sound emitters, and sound reflecting or absorbing objects can be placed and moved interactively.

This may be an important capability to have as NPSNET explores the dismounted infantry paradigm. Virtual military operations in an urban terrain simulations will certainly involve entering virtual buildings and rooms. The ability to acoustically model these areas would provide an important step forward in spatial acoustic presentations. This topic will be listed in the conclusions and recommendations chapter as a topic for further research.

VII. RECOMMENDATIONS AND CONCLUSIONS

A. GENERAL

As mentioned in the introduction, people trying to participate in a virtual world must have some sense of immersion and interaction with objects simulated in the 3D environment. If a participant sees a 3D graphical object and hears a non-3D audio event that is supposed to be connected to that object, the participant is confused and suffers from a lack of immersion. If the same object is coupled with realistic and appropriate 3D sound cues, immersion and emotional response increase dramatically because visual and audio cues are synchronized, and the overall experience appears to be much more believable. This thesis addressed methods of introducing believable 3D audio into virtual world simulations using headphones.

The primary goal of this thesis was to implement a headphone-delivered spatialized sound system for use within NPSNET. This goal was accomplished. NPS-ACOUST provides the capability of presenting 24 simultaneous spatialized sounds to an NPSNET participant. The dramatic increase in the realism of the presented aural cues significantly contributes to the NPSNET virtual experience. Secondary goals for this thesis included implementing a locally developed, low-cost solution and developing a method of 3D sound production that was easy to use.

A locally developed, low-cost solution is considered important because as mentioned in the introduction, every virtual world participant should be presented with realistic 3D audio. Moreover, one goal of on-going virtual simulation research within the DoD is to provide the capability for hundreds or thousands of players to simultaneously participate in the same virtual world. This mandates low-cost solutions for all aspects of virtual world production, not the least of which is 3D audio. Unfortunately, this goal was not reached in this thesis. The Acoustetron II that was purchased for NPS-ACOUST costs roughly $10,000. It is unreasonable to expect that every participant would have their own $10,000 Acoustetron II.

It should go without saying that any implementation that is developed should be easy to use. However, this is not always the case. Implementations that are difficult to understand are rarely used and become "shelfware." In NPS-ACOUST it is very easy to use. Delivering spatial sound to NPSNET is as simple as turning on the Acoustetron II and running NPS-ACOUST naming the appropriate master. The ease-of-use goal for this thesis was definitely met.

B. CONCLUSIONS

As discussed in previous chapters, inserting realistic 3D audio in a virtual world is not an easy task. The main obstacle is the processor intensive requirement to synthesize spatial sound from monaural sound samples in a real-time manner. The prohibitive costs involved in installing 3D sound in present day virtual world systems mandates the research of low-cost alternatives. Three different alternatives were studied in an attempt to deliver a locally developed solution. Obstacles were encountered for each alternative that could not be overcome given the current inventory of computer equipment in the NPS graphics lab. This section summarizes the three attempts and their shortcomings.

1. Workstation Rendering Sound same as Graphics

No workstation exists in the NPS graphics lab that was able to provide 3D graphics and sound rendering using the same system resources. The graphics lab's most powerfully configured workstation was only able to render two simultaneous spatial sounds while rendering dynamic scenes for NPSNET. More sounds were attempted and the presentation was degraded. One alternative considered was to install specialized audio hardware in the graphics workstation. Although the products developed by VSI and Paradigm Simulations, Inc. demonstrate progress towards the goal of real-time production of multiple spatialized sound in a virtual world, their solutions did not go far enough to met the 3D auditory expectations of NPSNET players. Succinctly put, there were no clear audio hardware solutions. There is still much work to be done in this area.

2. Library of Pre-Positioned Sounds

Pursuing the creation of a library of pre-positioned 3D sound cues seemed to be the most promising and best low-cost alternative. Given a robust enough sound file inventory and an efficient method of determining and recalling the appropriate sound file, discrete virtual audio cues could be presented with enough fidelity to service some measure of spatial audio requirements. However, too much latency was introduced in retrieving the sound file from disk storage. Moreover, the library would have to be manually created. The time required to create this library "by hand" was unreasonable.

Although this approach turned out to be less promising that had been hoped, it still remains a topic worthy of further research. Research connected to this thesis failed to find an application that could create a HRTF filtered sound file and store the result to a unique filename. It is recommended that further research be conducted into products that are commercially or publicly available that will capture filtered sound to a stored sound file format. Once this capability is realized, this alternative can be revisited.

3. Multiple Client Sound Server

A single sound server servicing multiple client sound requests was an attractive alternative for its economical considerations. This alternative required a workstation capable of rendering several sound requests simultaneously and a network with enough bandwidth to handle the resulting sizeable spatialized sound files. Neither existed and this alternative was abandoned.

C. TOPICS OF RESEARCH

Although this research went far to increase the level of immersion in as much as audio cues are concerned, much work remains in this area. The follow specific topics were issues that were left unresolved for one reason or another and in need of further research.

1. Special 3D Audio Cards

Finding a robust 3D audio card could greatly simplify 3D sound production in NPSNET. A 3D audio card that is a component of a graphics workstation is obviously a simple and optimal approach. The card would have to be robust enough to offer the same capabilities as does the Acoustetron II. Effort should be devoted to researching and reviewing new audio cards as they become available. It is only a matter of time before a 3D audio card is available that will adequately service the 3D acoustic needs of a virtual environment.

2. More Capable Workstations

Workstations will also continue to grow in capacity and capability. Mr. E.R. McCracken, CEO of Silicon Graphics Inc., said that he expects computing power to increase by 1000 times in the next decade and corresponding costs lowering by a similar margin. It follows (crudely perhaps) that a workstation capable of only two simultaneous sounds today will be capable of 2000 sounds in ten years.

3. NPSNET Heartbeat Entity State PDUs

Consideration to change NPSNET's policy of not sending out update ESPDUs as a result of vehicle velocity changes is recommended. The impact of sending out ESPDUs in this manner will be in the area of increased network load. Investigation into the costs and benefits of this change should be conducted with an eye towards sending out update ESPDUs for vehicle velocity changes if feasible.

4. Acoustetron II Software Bug

When a listener's head and a sound event are co-located and co-oriented, any like changes in those states should not result in a change in the binaural properties of the sound event. A bug was discovered in the Acoustetron II software that was causing this to happen. Attention to this problem should be occasional in the form of contact with CRE until the error is resolved.

5. Spatial Sound Manipulation Software Tool

Although investigation into the library of pre-positioned spatial files was discontinued, a tool should be found that is capable of filtering sounds and saving the output to disk in a batch-oriented, background process. If such a tool can be found, the sound library approach can be revisited if the introduced latency can be improved or accepted as a limitation to this specific method of spatial sound delivery.

D. RECOMMENDATIONS FOR FUTURE WORK

There are several opportunities that are presented as areas for follow-on work to this thesis. The Acoustetron II is a very powerful spatial sound tool and could contribute to several other sound applications within NPSNET as well as other projects requiring spatial sound. This section addresses some of those areas worthy of further exploration.

1. Create a Standard NPSNET Sound Class Interface

NPS-MONO, NPS-3DSS and NPS-ACOUST are very similar applications in the way they are implemented. In fact, they are all descended from a common network-based DIS-monitoring application. Because the applications are so similar, it is desirable to combine all three applications into one. A C++ sound class interface could be developed in which a generic public interface could service the functional requirements for applications requiring sound. The method in which the sound would be delivered would be determined at application start-up time. For example, if a workstation is capable of sound and the user wants to replay sounds using workstation resources, a command line option would be issued, interpreted and the appropriate class library would be used to instantiate a sound device object specific to replaying sound on the workstation's audio card. If on the next run of the application the Acoustetron II were desired, the appropriate command line option would be given and an Acoustetron II class object would be instantiated to service sound requests from the application to the Acoustetron II.

Appendix D presents the proposed public interface to a generic NPSNET sound class. Three distinct sound classes would need to be implemented (monoSoundClass, midiSoundClass, and acoustSoundClass), each with identical public interfaces. An example of the source code needed in the main application requiring sound to instantiate the proper sound class might look like the following:

```
#ifdef OPTION_MONO
  monoSoundClass soundDevice ( configuration_parameters );
#endif

#ifdef OPTION_MIDI
  midiSoundClass soundDevice ( configuration_parameters );
#endif

#ifdef OPTION_ACOUST
  acoustSoundClass soundDevice ( configuration_parameters );
#endif
```

Member functions for each of these objects would be identically implemented for each class and might look like the following in the main application:

```
soundDevice.init_sounds(soundDataFile);
soundDevice.playSound(soundEvent, soundPosition, listenerPosition);
soundDevice.updateListenerHeadPosition(vehicleLocation, vehicleOrientation);
```

Recall the discussion presented in the last chapter concerning the differing data requirements of NPS-ACOUST and NPS-MONO. The playSound() member function above passes as parameters the sound event to be played as well as the positions of the sound and the listener. All of this data is required by NPS-MONO while only the sound event and its position are required by NPS-ACOUST. In this case, the implementation of the member function for the NPS-MONO class would use all the data, while the NPS-ACOUST class would receive but ignore the listener position data. Another difference example is the updateListenerHeadPosition() member function. Updating the listener's head position is required for the Acoustetron II but not for NPS-MONO. In the interest of maintaining identical public interfaces for sound production, both classes would have this member function defined. The acoustSoundClass updateListenerHeadPosition() member function would be fully implemented. The corresponding monoSoundClass member function would essentially be a call to a dummy function (does nothing). While making a call to a function that does nothing is not necessarily economical programming, the cost is insignificant when compared to the overall benefit of creating identical interfaces to each of the three different methods of sound delivery. Application programmers implementing sound in their applications would simply make calls to the sound class and let the sound class determine the applicability of the function call to its particular method of delivering sound.

2. The Acoustetron II and Head Tracking Functionality

The head constraint issue raised in the previous chapter can be solved by implementing head tracking capabilities. This functionality is required in order to use the Acoustetron II with devices such as a HMD and other alternative graphical display devices. The Acoustetron II comes with the necessary device drivers to interpret head tracking data and perform appropriate sound rendering calculations based on dynamic head location and orientation data.

3. Using the Acoustetron II in a Loudspeaker Environment

It is possible to use the Acoustetron II to drive loudspeakers. By allowing the Acoustetron II to handle the sound spatialization requirements for loudspeaker delivery, the elaborate setup of equipment needed for NPS-3DSS's MIDI-based implementation could be substantially reduced. Additionally, there is some speculation in the spatial sound community as to whether using the MIDI protocol as a means of communicating spatial sound requests is the most efficient implementation. Using the Acoustetron II as an alternative to MIDI would provide a tool to benchmark and validate alternatives in 3D audio environments. Ultimately, the Acoustetron II might fully replace the suite of equipment used in NPS-3DSS.

4. Using the Acoustetron II as a Sound Server for Multiple Clients

Because the Acoustetron II is capable of spatially rendering up to 24 simultaneous sounds, it might be possible to allow the Acoustetron II to service more than one client. Research has shown that a listener can only interpret up to five sounds at any one time before becoming overwhelmed with auditory input. The challenge to this approach would be determining which client sent which request and then deliver to that client only the sounds particular to his requests.

E. FINAL THOUGHTS

Admittedly, this research effort became narrow in scope toward its end. It was hoped that a local implementation of some sort could be found. However, NPSNET 3D sound capability was greatly enhanced with the integration of the Acoustetron II. This research also provided insight and direction for future NPSNET sound systems. It is hoped that this thesis contributed to on-going efforts to establish the NRG as a leader in the application of 3D sound for use in VEs.

LIST OF REFERENCES

[BEGA94] Begault, D. R., *3-D Sound for Virtual Reality and Multimedia*, Academic Press, Cambridge, Massachusetts, 1994.

[BIDL94] Bidlack, Rick, *Virtual Sonic Display*, Banff Centre for the Arts (ftp://accessone.com/pub/misc/release/), 1994.

[BURG92] Burgess, D., *Real-Time Audio Spatialization with Inexpensive Hardware*, Georgia Institute of Technology, October, 1992.

[BURG92a] Burgess, David A., Verlinden, Jouke C., *A First Experience with Spatial Audio in a Virtual Environment*, Multimedia Computing Group, GVU Center, Georgia Institute of Technology, 1992.

[BURG92b] Burgess, David A., *Techniques for Low Cost Spatial Audio*, Multimedia Computing Group, GVU Center, Georgia Institute of Technology, September, 1992.

[BURG93] Burgess, David A., *The NA3 Audio Server*, Multimedia Computing Group, GVU Center, Georgia Institute of Technology, 1993.

[CRYS95] Crystal River Engineering Inc., *AudioReality™ White Paper*, Crystal River Engineering Inc., Palo Alto, CA, 1995.

[CRYS96a] Crystal River Engineering Inc., *Acoustetron II*, Crystal River Engineering Inc. Homepage (http://www.cre.com), Palo Alto, CA, 1996.

[CRYS96b] Crystal River Engineering Inc., *Acoustetron II, The AudioReality™ Sound Server Manual*, Crystal River Engineering Inc., Palo Alto, CA, 1996.

[DARK95] Darken, Rudy, *Spatial Acoustic Sounds for Virtual Environment Applications*, Rudy Darken Electronic Mail of 8 December 1995, Naval Research Labs, Washington D.C., 1995.

[DURL95] Durlach, Nathaniel I., Mavor, Anne S., et al, *Virtual Reality- Scientific and Technological Challenges*, National Academy Press, Washington, D.C., 1995.

[ERBE94] Erbe, Tom R., *SoundHack User Manual*, Version 0.74/0.80, California Institute of the Arts, 1994.

[ERBE96] Erbe, Tom R., *Subj: Re: SoundHack*, Tom Erbe Electronic Mail of 10 January 1996, California Institute of the Arts, 1996.

[MACE94] Macedonia, M., Zyda, M., Pratt, D., Barham, P. and Zeswitz, S., "NPSNET: A Network Software Architecture for Large Scale Virtual Environments," *Presence*, Vol. 3, No. 4, pp. 265-287, Fall, 1994.

[MACE95] Macedonia, M., Brutzman, D., Zyda, M., Pratt, D., Barham, P., Falby, J., Locke, J., *NPSNET: A Multi-Player 3D Virtual Environment over the Internet*, Proceedings of the ACM, 1995 Symposium on Interactive 3D Graphics, April, 1995.

[NCSA96] National Center for Supercomputing Applications, *The CAVE : A Virtual Reality Theater,* The Electronic Visualization Laboratory Homepage (http://notme.ncsa.uiuc.edu/EVL/docs/html/CAVE.overview.html), The University of Illinois at Urbana-Champaign, 1996.

[ROES94] Roesli, J., *Free-Field Spatialized Aural Cues for Synthetic Environments*, Master of Computer Science Thesis, Naval Postgraduate School, September, 1994.

[SALU93] Salute, Joan S., Begault Durand D., *Ames Auditory Display,* National Aeronautics and Space Administration, Ames Research Center Homepage (http://www.irsociety.com/nasa/asad.html), Office of Commercial Technology, 1993.

[STOR95] Storms, R., *NPSNET-3D Sound Server: An Effective Use of the Auditory Channel,* Master of Computer Science Thesis, Naval Postgraduate School, September, 1995.

[TONN94] Tonnesen, C. and Steinmetz, J., "3D Sound Synthesis," *Encyclopedia of Virtual Environments Homepage* (http://gimble.cs.umd.edu/vrtp/eve-main.html), Department of Computer Science, University of Maryland, 1994.

[WHEE93] Wheeler, Andrew, Ellinger, Joshua and Glicker, Steven, *The Design and Implementation of an Experimental Acoustical Display,* Applied Research Labs, University of Texas at Austin, 14 February 1993.

[ZESW93] Zeswitz, S., *NPSNET: Integration of Distributed Interactive Simulation (DIS) Protocol for Communication Architecture and Information Interchange,* Master of Computer Science Thesis, Naval Postgraduate School, September, 1993.

[ZYDA93] Zyda, M., Pratt, D., Falby, J., Barham, P. and Kelleher, K., "NPSNET and the Naval Postgraduate School Graphics and Video Laboratory," *Presence*, Vol. 2, No. 3, pp. 244-258, Summer, 1993.

[ZYDA94] Zyda, M., Pratt, D., Falby, J., Lombardo, C. and Kelleher, K., "The Software Required for the Computer Generation of Virtual Environments," *Presence*, Vol. 2, No. 2, pp. 130-140, Spring, 1993.

BIBLIOGRAPHY

Allen, J. B., and Berkeley, D. A., "Image Model for Efficiently Modeling Small-Room Acoustics", *Journal of the Acoustical Society of America*, Vol. 65, pp. 943-950, 1979.

Ando, Y., *Concert Hall Acoustics,* Berlin: Springer-Verlag, 1985.

Asano, F., Suzuki, Y., and Stone, T., "Role of spectral cues in median plane localization", *Journal of the Acoustical Society of America*, Vol. 88, pp. 159-168, 1990.

Backus, J., *The Acoustical Foundations of Music*, W. W. Norton & Company, New York, 1977. (Wave physics fundamentals of music.)

Ballou, G. (Ed.), *Handbook for Sound Engineers: The New Audio Cyclopedia,* Howard W. Sames & Co., Carmel, Indiana, 1991. (A great source of much sound related information.)

Bauck, J. and Cooper, D., "Generalized Transaural Stereo", in *Proceedings of the 93rd Convention of the Audio Engineering Society*, San Francisco, CA, October, 1992. (Shows how to get 3D audio from two loudspeakers. See also Klayman)

Begault, D. R. & Wenzel, E. M., "Techniques and applications for binaural sound manipulation in man-machine interfaces," *NASA TM102279*, 1990.

Begault, D. R. & Wenzel, E. M., "Technical aspects of a demonstration tape for three-dimensional sound displays," *NASA TM102826*, 1990.

Begault, D. R., "Challenges to the successful implementation of 3-D sound," *Journal of the Audio Engineering Society*, Vol. 39, pp. 864-870, 1991.

Begault, D. R., "Preferred sound intensity increase for a sensation of half distance," *Perceptual and Motor Skills*, Vol. 72, pp. 1019-1029, 1991.

Begault, D. R., "Audio Spatialization Device for Radio Communications," *Report No. ARC 12013-1CU*, NASA-Ames Research Center, 1992.

Begault, D. R., "Binaural Auralization and Perceptual Veridicality," in *Audio Engineering Society 93rd Convention Preprint No. 3421 (M-3)*, Audio Engineering Society, New York, 1992.

Begault, D. R., "Perceptual effects of synthetic reverberation on three-dimensional audio systems," *Journal of the Audio Engineering Society*, Vol. 40, pp. 895-904, 1992.

Begault, D. R., "Perceptual similarity of measured and synthetic HRTF filtered speech stimuli," *Journal of the Acoustical Society of America*, Vol. 92, p. 2334, 1992.

Begault, D. R., "The Virtual Reality of 3-D Sound," in L. Jacobson (Ed.) *CyberArts: Exploring Art and Technology*, Miller-Freeman, San Francisco, CA, 1992.

Begault, D. R., "Call sign intelligibility improvement using a spatial auditory display," *Report No. 104014*, NASA-Ames Research Center, 1993.

Begault, D. R., "Head-up Auditory Displays for Traffic Collision Avoidance System Advisories: A Preliminary Investigation," *Human Factors*, Vol. 35, pp. 707-717, 1993.

Begault, D. R., *3-D Sound for Virtual Reality and Multimedia*, Academic Press Professional, Cambridge, MA, 1994.

Begault, D. R., and Erbe, T., "Multichannel spatial auditory display for speech communications," *Journal of the Audio Engineering Society*, Vol. 42, pp. 819-826, 1994.

Begault, D. R., "Virtual acoustic displays for teleconferencing: Intelligibility advantage for "telephone grade" audio," in *Audio Engineering Society 98th Convention (preprints)*, 1995.

Begault, D. R. and Pittman, M. T., "3-D Audio Versus Head Down TCAS Displays," *International Journal of Aviation Psychology*, (in press).

Begault, D. R. and Wenzel, E. M., "Technical aspects of a demonstration tape for three-dimensional auditory displays," *Report No. TM 102286*, NASA-Ames Research Center, 1990.

Begault, D. R. and Wenzel, E. M., "Headphone localization of speech," *Human Factors*, Vol. 35, pp. 361-376, 1993.

Begault, D. R. and Wenzel, E. M., "Techniques and applications for binaural sound manipulation in human-machine interfaces," in *International Journal of Aviation Psychology*, Vol. 2, pp. 1-22, 1992.

von Bekesy, G., *Experiments in Hearing*, McGraw-Hill, New York, 1960.

Benade, A., *Fundamentals of Musical Acoustics*, Dover Publications, New York, 1976. (Wave physics fundamentals of music.)

Bishop, G., ... Wenzel, E. M., et al., "Research Directions in Virtual Environments: Report of an NSF Invitational Workshop," *Computer Graphics*, Vol. 26, pp. 154-177, 1992.

Blauert, J., "An introduction to binaural technology," in R. Gilkey and T. Anderson (Eds.) *Binaural and Spatial Hearing*, Lawrence Elbaum Associates, Hillsdale, NJ, (in press). (A recent survey of the uses for 3D audio.)

Blauert, J., *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, MA, 1983. (This is the standard book on the psychoacoustics of spatial hearing. Detailed and through.)

Blauert, J. (guest Ed.), "Special issue on auditory virtual environment and telepresence," *Applied Acoustics*, Vol. 36, Elsevier Applied Science, England, 1992.

Bloom, P. J., "Creating source elevation illusions by spectral manipulation," *Journal of the Audio Engineering Society*, Vol. 25, pp. 560-565, 1977.

Bregman, A. S., *Auditory Scene Analysis*, MIT Press, Cambridge, MA, 1990.

Bronkhorst, A. W. and Plomp, R., "The Effect of Head-Induced Interaural Time and Level Differences on Speech Intelligibility in Noise," *Journal of the Acoustical Society of America*, Vol. 83, pp. 1508-1516, 1988.

Burger, J. F., "Front-back discrimination of the hearing system," *Acustica*, Vol. 8, pp. 301-302, 1958.

Butler, R. A. and Belendiuk, K., "Spectral cues utilized in the localization of sound in the median sagittal plane," *Journal of the Acoustical Society of America*, Vol. 61, pp. 1264-1269, 1977.

Calhoun, G. L., Valencia, G. and Furness, T. A. III, "Three-dimensional auditory cue simulation for crew station design/evaluation," in *Proceedings of the Human Factors Society*, Vol. 31, pp. 1398-1402, 1987.

Cannon, R., *Dynamics of Physical Systems*, McGraw-Hill, New York, 1967. (Wave physics.)

Chan, C. J., "Sound Localization and Spatial Enhancement with the Roland Sound Space Processor," in *CyberArts: Exploring Art and Technology*, L. Jacobson (Ed.), pp. 95-104, Miller-Freeman Inc., San Francisco, CA, 1992.

Cherry, E. C., "Some Experiments of the Recognition of Speech with One and Two Ears," *Journal of the Acoustical Society of America*, Vol. 22, pp. 61-62, 1953.

Cherry, E. C. and Taylor, W. K., "Some further experiments on the recognition of speech with one and with two ears," *Journal of the Acoustical Society of America*, Vol. 26, pp. 549-554, 1954.

Churchland, P. S., *The Computational Brain*, MIT Press, Cambridge, MA, 1992.

Cohen, M. and Wenzel, E. M., "The Design of Multichannel Sound Interfaces," in W. Barfield and T. Furness III (Eds.) *Virtual Environments and Advanced Interface Design*, Oxford University Press, (in press).

Coleman, P. D., "Failure to localize the source distance of an unfamiliar sound," *Journal of the Acoustical Society of America*, Vol. 34(3), pp. 345-346, 1962.

Coleman, P. D., "An analysis of cues to auditory depth perception in free space," *Psychological Bulletin*, Vol. 60, pp. 302-315, 1963.

Critchley, M. and Henson, R. (Eds.), *Music and the Brain: Studies in the Neurology of Music*, Charles C. Thomas Publisher, Springfield, Illinois, 1977. (A collection of many engaging articles of interest to musicians.)

Doll, T. J., Gerth, J. M., Engleman, W. R. and Folds, D. J., "Development of simulated directional audio for cockpit applications," *USAF Report No. AAMRL-TR-86-014*, 1986.

Deutsch, Diana (Ed.), *The Psychology of Music*, Academic Press, 1982.

Dowling, W. J. and Harwood, D. L., *Music Cognition*, Academic Press, New York, 1986.

Durlach, N. I. and Colburn, H. S., "Binaural Phenomena," in E.C. Carterette and M. P. Friedman (Eds.) *Handbook of Perception*, Vol 4., New York, Academic Press, 1978.

Durlach, N. I., Rigopulos, A., Pang, X. D., Woods, W. S., Kulkarni, A., Colburen, H. S. and Wenzel, E. M., "On the externalization of auditory images," *Presence*, Vol. 1 (2), pp. 251-257, Spring 1992. (Aimed at virtual reality applications. See also Wenzel.)

Durlach, N. I. and Mavor, A. S., Eds., *Virtual Reality: Scientific and Technological Challenges: Report of the Committee on Virtual Reality Research and Development*, Washington, D.C., National Academy Press, 1994.

Fisher, H. and Freeman, S. J., "The role of the pinna in auditory localization," *Journal of Audio Research*, Vol. 8, pp. 15-26, 1968.

Fisher, S. S., Wenzel, E. M., Coler, C. and McGreevy, M. W., "Virtual interface environment workstations," in *Proceedings of the Human Factors Society*, Vol. 32, pp. 91-95, 1988.

Foster, S. H., *Convolvotron™ User's Manual*, Crystal River Engineering, Inc., 12350 Wards Ferry Road, Groveland, CA 95321, 1988.

Foster, S. H., Wenzel, E. M. and Taylor, R. M., "Real-time synthesis of complex acoustic environments [Summary]," in *Proceedings of the ASSP (IEEE) Workshop on Applications of Signal Processing to Audio & Acoustics*, New Paltz, New York, 1991.

Foster, S. H. and Wenzel, E. M., "Virtual acoustic environments: The Convolvotron [Summary]," *Computer Graphics*, Vol. 25 (4), p. 386, Demonstration system at the 1st annual "Tomorrow's Realities Gallery", SIGGRAPH '91, 18th ACM Conference on Computer Graphics and Interactive Techniques, Las Vegas, Nevada, July 27 - August 2, 1991.

Gardner, M. B. and Gardner, R. S., "Problem of Localization in the Median Plane: Effect of Pinnae Cavity Occlusion," *Journal of the Acoustical Society of America*, Vol. 53, pp. 400-408, 1973.

Gehring, B., *Focal Point™ 3D Sound User's Manual*, Gehring Research Corporation, 189 Madison Avenue, Toronto, Canada, M5R 2S6, 1990.

Gierlich, H. W., "The Application of Binaural Technology," *Applied Acoustics*, Vol. 36, pp. 219-244, 1992.

Gilkey, R. and Anderson, T., (Eds.), *Binaural and Spatial Hearing*, Lawrence Erlbaum Associates, Inc., New Jersey, (in press).

Hahn, J., "An Integrated Virtual Environment System," *Presence*, Vol. 2, pp. 353-360, 1944.

Hall, D., *Musical Acoustics*, 2nd Ed., Brooks/Cole Publishing, Belmont CA, 1991. (Wave physics fundamentals of music.)

Hartman, W. M., "Localization of sound in rooms," *Journal of the Acoustical Society of America*, Vol. 74, pp. 1380-1391, 1983.

HEAD Acoustics, *Binaural Mixing Console* [product literature], Contact: Sonic Perceptions, 114A Washington Street, Norwalk, CT 06854.

Helmholtz, H., *Sensations of Tone*, Dover Publications, New York, 1954. (A classic. Originally published in 1885. Still a very good book. Great experimental techniques.)

Humanski, R. A. and Butler, R. A., "The contribution of the near and far ear toward localization of sound sources on the median plan," *Journal of the Acoustical Society of America*, Vol. 83, pp. 2300-2310, 1988.

Kendall, G. S. and Martens, W. L., "Simulating the Cues of Spatial Hearing in Natural Environments," in *Proceedings of the International Computer Music Conference*, 1984.

Klayman, A. I., "SRS: Surround sound with only two speakers," *Audio*, Vol. 8, pp. 32-37, August 1992. (Probably the most successful commercial system for spatialized audio.)

Kistler, D. K. and Wightman, F. L., "A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction," *Journal of the Acoustical Society of America*, Vol. 91, pp. 1637-1647, 1992.

Kramer, G. (Ed.), "Auditory Display: Sonification, Audification, and Auditory Interfaces," in *Proceedings Volume XVIII*, Santa Fe Institute Studies in the Sciences of Complexity, Reading MA, Addison-Wesley, 1994.

Kuhn, G. F., "Model for the interaural time differences in the azimuthal plane," *Journal of the Acoustical Society of America*, Vol. 62, pp. 157-167, 1977.

Loomis, J. M., Hebert, C. and Cicinelli, J. G., "Active localization of virtual sounds," *Journal of the Acoustical Society of America*, Vol. 88, pp. 1757-1764, 1990.

Macpherson, E. A., "On the role of head-related transfer function spectral notches in the judgement of sound source elevation," In G. Kramer (Ed.) *Proceedings of the 1994 International Conference on Auditory Displays*, Santa Fe, NM, (in press).

Makous J. C. and Middlebrooks, J. C., "Two-dimensional sound localization by human listeners," *Journal of the Acoustical Society of America*, Vol. 87, pp. 2188-2200, 1990.

McKinley, R. L. and Ericson, M. A., "Digital synthesis of binaural auditory localization azimuth cues using headphones," *Journal of the Acoustical Society of America*, Vol. 83, S18, 1988.

Mehrgardt, S. and Mellert, V., "Transformation characteristics of the external human ear," *Journal of the Acoustical Society of America*, Vol. 61, pp. 1567-1576, 1977.

Mershon, D. H. and King, L. E., "Intensity and reverberation as factors in the auditory perception of egocentric distance," *Perception and Psychophysics*, Vol. 18, pp. 409-415, 1975.

Middlebrooks, J. C., Makous, J. C. and Green, D. M., "Directional sensitivity of sound-pressure levels in the human ear canal," *Journal of the Acoustical Society of America*, Vol. 86, pp. 89-108, 1989.

Middlebrooks, J. C., and Green, D. M., "Directional dependence of interaural envelope delays," *Journal of the Acoustical Society of America*, Vol. 87, pp. 2149-2162, 1990.

Middlebrooks, J. C., "Narrow-band sound localization related to external ear acoustics," *Journal of the Acoustical Society of America*, Vol. 92, pp. 2607-2624, 1992.

Middlebrooks, J. C. and Green, D. M., "Sound Localization by Human Listeners," *Annual Review of Psychology*, Vol. 42, pp. 135-159, 1991.

Mills, A. W., "Auditory Localization", in J. V. Tobias (Ed.) *Foundations of Modern Auditory Theory*, Vol. II, pp. 303-348, Academic Press, New York, 1972. (A slightly dated but easy to understand survey.)

Mowbray, G. H. and Gebhard, J. W., "Man's senses as informational channels," in H. W. Sinaiko (Ed.) *Human Factors in the Design and Use of Control Systems*, pp. 115-149, Dover Publications, New York, 1961.

Oldfield, S. R. and Parker, S. P. A., "Acuity of sound localization: a topography of auditory space. I. Normal hearing conditions," *Perception*, Vol. 13, pp. 601-617, 1984.

Oldfield, S. R. and Parker, S. P. A., "Acuity of sound localization: a topography of auditory space. II. Pinna cues absent," *Perception*, Vol. 13, pp. 601-617, 1984.

Oldfield, S. R. and Parker, S. P. A., "Acuity of sound localization: a topography of auditory space. III. Monaural hearing conditions," *Perception*, Vol. 15, pp. 67-81, 1986.

Perrot, D. R., "Studies in the perception of auditory motion," in R. W. Gatehouse (Ed.) *Localization of Sound: Theory and Applications*, pp. 169-193, Amphora Press, Groton, CN, 1982.

Perrott, D. R., "Concurrent minimum audible angle: a re-examination of the concept of auditory spatial acuity," *Journal of the Acoustical Society of America*, Vol. 75, pp. 1201-1206, 1984.

Perrott, D. R., "Discrimination of the spatial distribution of concurrently active sound sources: Some experiments with stereophonic arrays," *Journal of the Acoustical Society of America*, Vol. 76, pp. 1704-1712, 1984.

Perrott, D. R. and Tucker, J., "Minimum audible movement angle as a function of signal frequency and the velocity of the source," *Journal of the Acoustical Society of America*, Vol. 83, pp. 1522-1527, 1988.

Perrott, D. R., Sadralodabai, T., Saberi, K. and Strybel, T. Z., "Aurally aided visual search in the central vision field: Effects of visual load and visual enhancement of the target," *Human Factors*, Vol. 33, pp. 389-400, 1991.

Persterer, A., "A very high performance digital audio processing system," in *Proceedings of the ASSP (IEEE) Workshop on Applications of Signal Processing to Audio & Acoustics*, New Paltz, New York, 1989.

Pierce, J. R., *The Science of Musical Sound*, revised edition, W. H. Freeman, New York, 1992. (Wave physics fundamentals of music.)

"The Physics of Music," *Scientific American*, W. H. Freeman and Company, San Francisco, CA, 1978. (Wave physics fundamentals of music.)

Plenge, G., "On the difference between localization and lateralization," *Journal of the Acoustical Society of America*, Vol. 56, pp. 944-951, 1974.

Plomp, R., *Aspects of the Tone Sensation*, Academic Press, London, 1976. (A compendium of psychoacoustic experiments and results.)

Lord Rayleigh [Strutt, J. W.], "On Our Perception of Sound Direction," *Philosophical Magazine*, Vol. 13, pp. 214-232, 1907.

Roffler, S. K. and Butler, R. A., "Factors that influence the localization of sound in the vertical plane," *Journal of the Acoustical Society of America*, Vol. 43, pp. 1255-1259, 1968.

Roffler, S. K., and Butler, R. A., "Localization of tonal stimuli in the vertical plane," *Journal of the Acoustical Society of America*, Vol. 43, pp. 1260-1266, 1968.

Rossing, T., *The Science of Sound*, Addison-Wesley, Reading, MA, 1990. (The acoustics of musical instruments is covered in an organized manner.)

Sakamoto, N., Gotoh, T., and Kimura, Y., "On 'out-of-head localization' in headphone listening," *Journal of the Audio Engineering Society*, Vol. 24, pp. 710-716, 1976.

Schroeder, M. R., "Digital Simulation of Sound Transmission in Reverberant Spaces," *Journal of the Acoustical Society of America*, Vol. 47, pp. 424-431, 1970.

Schubert, E. D., *Hearing: Its Function and Dysfunction*, Springer-Verlag/Wien, New York, 1980. (Although out of date, it was the state of the art for the 1980's, and is presented in breadth and depth.)

Seashore, C., *Psychology of Music*, Dover Publications, New York, 1967. (Although not explicitly referenced, the principles of Gestalt theorists come into play.)

Shaw, E. A. G., "The External Ear," in W. D. Keidel and W. D. Neff (Eds.) *Handbook of Sensory Physiology, Vol. V/1, Auditory System*, pp. 455-490, Springer-Verlag, New York, 1974.

Shinn-Cunningham, B. G., Lehnert, H., Kramer, G., Wenzel, E. M. and Durlach, N. I., "Auditory Displays," in R. Gilkey and T. Anderson (Eds.) *Binaural Hearing*, Lawrence Erlbaum Associates Inc., New Jersey, (in press).

Spiegle, J. M. and Loomis, J. M., "Auditory distance perception by translating observers," in *Proceedings of the IEEE Symposium in Research Frontiers in Virtual Reality*, San Jose, CA, October 25-26, 1993.

Strum, R. and Kirk, D., *First Principals of Discrete Systems and Digital Signal Processing*, Addison-Wesley, Reading, MA, 1989. (A good book for understanding convolution and spectral analysis having many figures.)

Sunier, J., "Binaural overview: Ears where the mikes are. Part I," *Audio*, Vol. 73, pp. 75-84, November 1989. (Excellent survey of binaural recording practices.)

Sunier, J., "Binaural overview: Ears where the mikes are. Part II," *Audio*, Vol. 73, pp. 49-57, December 1989. (Excellent survey of binaural recording practices.)

Takala, T., Hahn, J., Gritz, L., Geigel, J. and Lee, J., "Using physically-based models and genetic algorithms for functional composition of sound signals, synchronized to animated motion," *International Computer Music Conference (ICMC)*, Tokyo, Japan, September 10-15, 1993.

Thurlow, W. R. and Runge, P. S., "Effects of induced head movements on localization of direction of sound sources," *Journal of the Acoustical Society of America*, Vol. 42, pp. 480-488, 1967.

Wallach, H., "On sound localization," *Journal of the Acoustical Society of America*, Vol. 10, pp. 270-274, 1939.

Wallach, H., *"The role of head movements and vestibular and visual cues in sound localization," Journal of Experimental Psychology*, Vol. 27, pp. 339-368, 1940.

Warren, D. H., Welch, R. B. and McCarthy, T. J., "The Role of Visual-Auditory 'Compellingness' in the Ventriloquism Effect: Implications for Transitivity Among the Spatial Senses," *Perception and Psychophysics*, Vol. 30, pp. 557-564, 1981.

Watkins, A. J., "Psychoacoustical aspects of synthesized vertical locale cues," *Journal of the Acoustical Society of America*, Vol. 63, pp. 1152-1165, 1978.

Welch, R. B., *Perceptual Modification: Adapting to Altered Sensory Environments*, New York, Academic Press, 1978.

Wenzel, E. M., "Perceptual factors in virtual acoustic displays" [Invited Keynote Speaker], in *Proceedings of ICAT' 94*, 4th International Conference on Artificial Reality and Tele-Existence, Tokyo, Japan, pp. 83-98, 1994.

Wenzel, E. M., "Spatial Sound and Sonification," in G. Kramer (Ed.) *Auditory Display: Sonification, Audification, and Auditory Interfaces*, Addison-Wesley, Reading, MA, pp. 127-150, 1994.

Wenzel, E. M. and Foster, S. H., "Perceptual consequences of interpolating head-related transfer functions during spatial synthesis," in *Proceedings of the ASSP (IEEE) Workshop on Applications of Signal Processing to Audio & Acoustics*, New Paltz, New York, October 17-20, 1993.

Wenzel, E. M., Gaver, W., Foster, S. H., Levkowitz, H. and Powell, R., "Perceptual vs. hardware performance in advanced acoustic interface design," in *Proceedings of INTERCHI' 93*, Conference on Human Factors in Computing Systems, Amsterdam, pp. 363-366, 1993.

Wenzel, E. M., Arruda, M., Kistler, D. J. and Wightman, F. L., "Localization using non-individualized head-related transfer functions," *Journal of the Acoustical Society of America*, Vol. 94, pp. 111-123, 1993.

Wenzel, E. M., "Launching sounds into space," in L. Jacobson (Ed.) *CyberArts: Exploring Art and Technology*, Miller-Freeman Inc., San Francisco, CA, 1992.

Wenzel, E. M., "Three-dimensional virtual acoustic displays," in M. Blattner and R. Dannenberg (Eds.) *Multimedia Interface Design*, ACM Press, New York, 1992.

Wenzel, E. M., and Foster, S. H., *"Virtual Acoustic Environments*. [Summary: demonstration system]," in *Proceedings of the CHI' 92*, ACM Conference on Computer-Human Interaction, Monterey, CA, p. 676, 1992.

Wenzel, E. M., "Localization on virtual acoustic displays," *Presence*, Vol. 1, pp. 80-107, Winter 1992.

Wenzel, E. M., "Virtual Acoustic Displays: Localization in Synthetic Acoustic Environments [Plenary speech]," in *Proceedings of Speech Tech'92*, February 4-5, New York, NY, 1992.

Wenzel, E. M., "Three-dimensional virtual acoustic displays," *NASA TM103835*, 1991.

Wenzel, E. M., Wightman, F. L. and Kistler, D. J., "Localization of non-individualized virtual acoustic display cues," in *Proceedings of the CHI'91*, ACM Conference on Computer-Human Interaction, New Orleans, LA, April 27-May 2, 1991.

Wenzel, E. M., Stone, P. K., Fisher, S. S. and Foster, S. H., "A system for three-dimensional acoustic 'visualization' in a virtual environment workstation," in *Proceedings of the IEEE Visualization'90 Conference,* San Francisco, CA October 23-26, pp. 329-337, 1990.

Wenzel, E. M., *"Virtual acoustic displays,"* in *Human Machine Interfaces for Teleoperators and Virtual Environments*, Santa Barbara, CA, March 4-9, NASA Conference Publication 10071, 1990.

Wenzel, E. M., and Foster, S. H., "Real-time digital synthesis of virtual acoustic environments," *Computer Graphics*, 1990.

Wenzel, E. M., Foster, S. H., Wightman, F. L. and Kistler, D. J., "Real-time Digital Synthesis of Localized Auditory Cues Over Headphones," in *Proceedings of the ASSP (IEEE) Workshop on Applications of Signal Processing to Audio & Acoustics*, New Paltz, NY, October 15-18, 1989.

Wenzel, E. M., Foster, S. H., Wightman, F. L. and Kistler, D. J., "Real-time Synthesis of Localized Auditory Cues," in *Proceedings of CHI'89*, ACM Conference of Computer-Human Interaction, Austin, TX, April 30 - May 5, 1989.

Wenzel, E. M., Wightman, F. L., Kistler, D. J. and Foster, S. H., "Acoustic origins of individual differences in sound localization behavior," *Journal of the Acoustical Society of America*, Vol. 84, S79(A), 1988.

Wenzel, E. M., Wightman, F. L., and Foster, S. H., "A virtual display system for conveying three-dimensional acoustic information," in *Proceedings of the Human Factors Society*, Vol. 32, pp. 86-90, 1988.

Wenzel, E. M., Fisher, S. S., Wightman, F. L. and Foster, S. H., "Application of auditory spatial information in virtual display systems," *CHABA Symposium on Sound Localization*, Sponsored by the National Academy of Science and the AFOSR, Washington, D. C., October 14-16, 1988.

Wenzel, E. M., Wightman, F. L. and Foster, S. H., "Development of a three-dimensional auditory display system," *SIGCHI Bulletin*, Vol. 20, pp. 52-57, 1988.

Wenzel, E. M., Wightman, F. L. and Foster, S. H., "Development of a three-dimensional auditory display system," in *Proceedings of CHI'88*, ACM Conference on Computer-Human Interaction, Washington, D. C., May 15-19, 1988.

Wightman, F. L. and Kistler, D. J., "Headphone Simulation of Free-field Listening I: Stimulus Synthesis," *Journal of the Acoustical Society of America*, Vol. 85, pp. 858-867, 1989.

Wightman, F. L. and Kistler, D. J., "Headphone Simulation of Free-field Listening II: Psychophysical Validation," *Journal of the Acoustical Society of America*, Vol. 85, pp. 868-878, 1989.

Wightman, F. L. and Kistler, D. J., "The Dominant Role of Low-frequency Interaural Time Differences in Sound Localization," *Journal of the Acoustical Society of America*, Vol. 91, pp. 1648-1661, 1992.

Wightman, F. L. and Kistler, D. J., "Multidimensional Scaling Analysis of Head-Related Transfer Functions," in *Proceedings of the ASSP (IEEE) Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE Press, New York, 1993.

Wightman, F. L., Kistler, D. J. and Anderson, K., "Reassessment of the role of Head Movements in Human Sound Localization," *Journal of the Acoustical Society of America*, Vol. 95, pp. 3003-3004, 1994.

Wightman, F. L. and Kistler, D. J., "The Importance of Head Movements for Localizing Virtual Auditory Display Objects," in G. Kramer (Ed.) *Proceedings of the 1994 International Conference on Auditory Displays*, Santa Fe, NM, (in press).

Zahorik, P. A., Kistler, D. J. and Wightman, F. L., "Sound Localization in varying virtual acoustic environments," in G. Kramer (Ed.) *Proceedings of the 1994 International Conference on Auditory Displays*, Santa Fe, NM, (in press).

Zurek, P. M., "Binaural Advantages and Directional Effects in Speech Intelligibility," in G. A. Studebaker and I. Hochberg (Eds.) *Acoustical Factors Affecting Hearing Aid Performance*, Allyn and Bacon, Needham Heights, MASS, 1993.

APPENDIX A: DEFINITIONS AND ABBREVIATIONS

A. DEFINITIONS

3D Sound: refers to the fact that sounds in the real world are three-dimensional. Human beings have the ability to perceive sound spatially, meaning that they can figure out where a sound is coming from, and where sounds are in relation to their surroundings and in relation to each other. There are three main pieces of information that are essential for the human brain to perform these functions:

Interaural Time Difference (ITD) means that unless a sound is located at exactly the same distance from each ear (e.g. directly in front), it will arrive earlier at one ear than the other. If it arrives at the right ear first, the brain knows that the sound is somewhere to the right.

Interaural Intensity Difference (IID) is similar to ITD. It says that if a sound is closer to one ear, the sound's intensity at that ear will be higher than the intensity at the other ear, which is not only further away, but usually receives a signal that has been shadowed by the listener's head.

Finally, the trickiest part of spatialization is the fact that a sound bounces off a listener's shoulders, face, and outer ear, before it reaches the ear drum. The pattern that is created by those reflections is unique for each location in space relative to the listener. A human brain can therefore learn to associate a given pattern with a location in space.

Since 3D sound consists of two signals (left and right ear) it can be rendered on conventional stereo equipment, preferably headphones (because of the clean separation of the two signals). The 3D sound produced by a direct path Aureal 3D system is combined with sound reflections (wavetracing) to create a very high level of realism and immersion in a sound space.

Ambient Channel: a way of displaying sounds as coming from everywhere - all around the listener. This is useful for background music or ambiance sounds such as rain.

Atmospheric Absorption: the attenuation of sounds as they propagate through a medium. For example, in air the high frequency components of sound attenuate faster than the lower frequency components.

Aureal 3D: binaural, immersive, interactive, real-time 3D audio technology by Crystal River Engineering (a trademarked term).

Auralization: the process of rendering audio by physically or mathematically modeling a soundfield of a source in space in such a way as to simulate the binaural listening experience at any given position in a modeled space.

Binaural: two audio tracks, one for each ear (as opposed to stereo, which is one for each speaker). Binaural sounds are what we hear in everyday life.

Convolvotron: the world's first multi-source, real-time, digital spatialization system built by Crystal River Engineering for NASA in 1987.

Direct path: the direct path from a sound source to a listener's ears (as opposed to reflections off of surfaces). The direct path allows a listener to tell where each sound is coming from, 360 degrees both in azimuth and elevation. This is the main concept of any 3D sound system.

Doppler Effect: the change in frequency of a sound wave due to the motion of a sound source or of a listener. For example, if a car moves past a listener while sounding its horn, the listener will hear a sudden drop in pitch as the car passes.

Extended Stereo: a term that summarizes a number of techniques that involve processing of traditional stereo sounds with the goal of making them appear to originate from a range which extends beyond the physical speaker locations. The effect is often limited to a planar arc in front of the listener with everything at the same elevation. Extended stereo effects tend to be incompatible with headphone listening and to only have the intended effect if the listener is located at a particular spot in relation to the speakers (see "sweet spot").

Foster, Scott: the founder of Crystal River Engineering and inventor of the Convolvotron. Often confused with Scott Fisher, his friend and founder of Telepresence Research.

Gain: the amplification or attenuation of a sound source, usually measured in dB (decibels). 0 dB means no amplification and no attenuation. A positive value amplifies a source, a negative value attenuates it.

HRTF: Head Related Transfer Functions (HRTFs) are a set of mathematical transformations which can be applied to a mono sound signal. The resulting left and right signals are the same as the signals that someone perceives when listening to a sound that is coming from a location in real-life 3D space. HRTFs are the core concept behind Aureal 3D, since they contain the information that is necessary to simulate a realistic sound space (see spatialization). Once the HRTF of a generic person is captured, it can be used to create Aureal 3D sound for a large percentage of the population (most people's heads and ears, and therefore their HRTFs, are similar enough for the filters to be interchangeable).

IID: Interaural Intensity Difference, see "3D sound".

ITD: Interaural Time Difference, see "3D sound".

Listener: an object in a sound space that is sampling ("listening to") sound, usually a head with associated HRTF characteristics.

Materials: by absorbing sound energy at different frequencies, the material of which an object is made effects the way the sound reflects off and transmits through the object. A carpeted room sounds very different from a glass room. An object's material characteristics can be measured empirically by recording known sounds as they bounce off of materials.

Medium: see "atmospheric absorption" and "transmission loss".

Mono/Monophonic: refers to a single audio signal, usually rendered on a single speaker. Mono sounds appear to originate from the speaker, or from the center of a listener's head in the case of headphones.

MIDI: Musical Instrument Digital Interface (MIDI) is a standard control language that is used for communication between electronic music and effects devices.

Psychoacoustics: an area of psychology that studies the structure and performance of human auditory perception.

Quadraphonic Sound: refers to four audio signals, usually rendered on four separate speakers. Quadraphonic sounds appear to originate from somewhere in-between the four speakers. The inconvenience associated with the amount of equipment necessary to produce quadraphonic sound, coupled with the fact that it is not compatible with conventional stereo equipment (and therefore headphones), makes quadraphonic sound an unpopular choice.

Radiation Pattern: each sound-emitting object can optionally radiate sound in a certain pattern (rather than uniformly all around it). For example, a head should emit sounds in the direction that its nose is pointing.

Reflection: a sound reflection off of a surface. It gives a listener information about the listening environment and the location and motion of sound sources. See "surfaces".

Refraction: sounds get refracted as they travel around the edges and through openings of objects.

Reverberation: or reverb, refers to the sum of all sound reflections in a listening environment.

Sample Rate: the number of samples per second at which a sound is processed (usually ranges from 8kHz to 50kHz (CD quality is 44.1kHz, or 44,100 samples per second).

Source: refers to an object in 3D space that emits sound. The actual sound signal that it sends out can be a live signal, a wave file, a MIDI voice, or any other audio signal. A 3D sound device often gets rated on how many different sources it can independently position at any one time. Realistic sound spaces can be created with as few as four concurrent sources, very complex spaces can have dozens of separate sounds at a time.

Speaker Arrays: an installation of multiple speakers in a certain pattern, usually designed to create a sound field within the space defined by the speakers. Examples are stereo speakers, or quadraphonic speakers.

Stereo/Stereophonic: refers to two audio signals, usually rendered on two separate speakers. Stereo sounds appear to originate from somewhere between the two speakers, or between the ears of a listener in the case of headphones.

Surfaces: sounds not only travel to a pair of ears on a direct path, but they also bounce off of objects in the world. Most natural listening environments contain at least a sound reflecting ground plane, such as a floor. Therefore, reflecting objects are necessary to make virtual environments sound natural and realistic. They help listeners navigate and enhance the overall effect of immersion in a virtual environment. Almost as important as reflections, is the absence of a reflection. For example, the brain can tell the change in a sound space when A reflection is removed by opening a door or a window.

Sweet Spot: the location where a listener has to be placed to get the optimal effect when listening to a specific speaker setup.

Transmission Loss: sounds get absorbed as they travel through objects such as walls (similar to atmospheric absorption in the case of traveling through a medium). Transmission loss models are needed to realistically simulate sounds outside a window or in the next room.

Update Rate: the number of times that a specific instance of a sound space gets re-computed and updated per second. Each time any object moves (most often the listener), the space needs to get updated. The higher the update rate, the faster objects can move without creating audio artifacts, such as clicking. Audio update rates generally range from a minimum of 20Hz to 100Hz. Video update rates are usually in the same range (TV signals are updated at 30Hz).

Wave File: a digital sound file stored in the Microsoft RIFF file format.

Wavetracing: the idea of tracing sound waves as they emit from a source and bounce around an environment (walls, objects, openings). The resulting sound reflections are rendered to a listener to create a more convincing 3D effect, as well as a more immersive, familiar, and realistic sound space.

B. ABBREVIATIONS

3D	Three Dimensional
C++	A Programming Language

CD	Compact Disc (16 bit audio)
CP-1 Plus	Lexicon Digital Audio Environment Processor
CPU	Central Processing Unit
DAT	Digital Audio Tape
dB	Decibel
DIS	Distributed Interactive Simulation
DSP	Digital Signal Processor/Processing
EMAX II	16 bit digital sound system keyboard/sampler manufactured by E-Mu Corporation
Ensoniq DP/4	MIDI capable parallel effects processor containing 4 processors manufactured by Ensoniq Corporation
FIR	Finite Impulse Response
HRTF	Head-Related Transfer Function
IID	Interaural Intensity Difference
ITD	Interaural Time Difference
IP	Internet Protocol
LAN	Local Area Network
MHz	Mega Hertz
MIDI	Musical Instrument Digital Interface
ms	milliseconds
NPS	Naval Postgraduate School
NPSNET	Naval Postgraduate School Networked Vehicle Simulator
NPSNET-PAS	NPSNET-Polyphonic Audio Spatializer
NRG	NPSNET Research Group
PDU	Protocol Data Unit
Polhemus Fastrack	Motion Tracker

SGI	Silicon Graphics Incorporated
Speed of Sound	335.28 meters per second in air at sea level and 70 degrees Fahrenheit
RAM	Random Access Memory
VE	Virtual Environment

APPENDIX B: NPS-ACOUST SETUP GUIDE

A. HARDWARE

The following setup steps are necessary in order to use NPS-ACOUST in conjunction with the Acoustetron II.

- Connect the Acoustetron II to your client workstation using the provided serial cable. On the Acoustetron II side, connect the serial cable to COM1. On the client workstation, connect the serial cable to an available serial port (default is TTYD1) (See the SOFTWARE SETUP section below if a serial port besides TTYD1 is desired. If so, an environmental variable will need to be set).

- Connect the monitor, keyboard, mouse and power cables to the Acoustetron II.

- Connect the Acoustetron II sound outputs to the Symetrix headphone amplifier using the 1/4 inch stereo cables.

- Connect the Sennheiser headphones to the Symetrix headphone amplifier.

B. SOFTWARE

The following steps are necessary to run NPS-ACOUST.

- If a serial port on the client workstation other than TTYD1 is desired, make sure the following environment variable is set using the command (usually located in the .cshrc file) **setenv TRONCOM x@yyy,zzz** where x is the serial port number (TTYDx), yyy is the baudrate divided by 100, and zzz the time-out period (the amount of time the client will wait for a response from the Acoustetron II on an init() call).

- To test the Acoustetron II locally, power up the Acoustetron II. When the initial menu appears, press the '2' key twice. You should hear a demo running on your system. If not, check the master volume control as well as the individual volume control on the Symetrix headphone amplifier.

- To check the Acoustetron II as a sound server controlled by the client workstation, on the workstation change to the subdirectory that contains the current version of NPSNET. From there, change to the src/apps/acoustsound/bin subdirectory. Run the **demo** or **test** programs to start up a demo sequence controlled by the client workstation. If the demo sequence fails to run on the Acoustetron II, refer to the Acoustetron II user guide for troubleshooting instructions.

- To run NPS-ACOUST, on the client workstation change to the subdirectory that contains the current version of NPSNET. Issue the following command:

npsacoust -MASTER master_workstation -DIS_EXERCISE 5 -SOUND_FILE datafiles/acoustetron.dat -ROUND_WORLD_FILE datafiles/benning/utm.orgin.dat

APPENDIX C: SOUND FILES AVAILABLE ON THE ACOUSTETRON II

A. GENERAL

There is a large collection of sounds available on the Acoustetron II. Because the Acoustetron II is implemented on a PC platform using Windows 3.1, the software expects files to be named using the DOS 8.3 filenaming convention. Additionally, the Acoustetron II can only render sound files that are in the Microsoft wave file format (.wav). Any sound file samples that are not wave file formatted must be converted. The SFCONVERT utility available on Silicon Graphics workstations does a good job in converting sound files from most formats to wave file formats. The syntax for using SFCONVERT is as follows:

sfconvert sound.aiff sound.wav format wave int 16 2 chan 1 rate 22050 byteorder little

This command interpreted is "convert sound.aiff to sound.wav using the wave format (format wave), store it as an Integer 16 bits, 2's compliment (int 16 2), 1 channel (chan 1) at 22.050 KHz sampling rate (22050) and use the little endian integer data (byteorder little)."

Most sound files used in NPS-ACOUST use the 22.05 KHz sampling rate for sound files. The Acoustetron II can replay 24 simultaneous sounds when set for 22.05 KHz as opposed to 12 for 44.1 KHz. Also, a high degree of sound quality is not needed for battlefield sound events (explosions, etc.). However, it is appropriate to use 44.1 KHz sampled sounds in some instances. Therefore, for many of the sampled sounds stored on the Acoustetron II, both 22.05 and 44.1 KHz sampled versions are available. Filenames that start with a "2" are 22.05 KHz samples while filenames beginning with a "4" are 44.1 KHz sampled. You must set the Acoustetron II to replay the sound files at the desired sampling rate. One caveat here is that you can play either version of the sampled file at either rate. For example, you can set the Acoustetron II to replay the sound files at 22.05 KHz and then play a 44.1 KHz sampled sound file. The Acoustetron II automatically converts the file's sampling rate and then replays it at 22.05 KHz. Why have two different versions of the same file then? The main reason is fidelity of sound. If you are more interested in quality of sound then in quantity, you will want to use the 44.1 KHz sampled files (CD quality). Taking a 22.05 KHz sampled file and replaying it at 44.1 KHz does not improve the fidelity of the sound. The reason to have the 22.05 KHz sampled file versions is that they are half of the size of the 44.1 KHz sampled files and use less memory to load.

B. WAVEFILE LISTING

WAVE FILE	DESCRIPTION
225mm.wav \ 425mm.wav	- 25mm machine gun fire
23m60s.wav \ 43m60s.wav	- three M-60 machine guns firing
2500lb.wav \ 4500lb.wav	- 500 pound bomb explosion
250cal.wav \ 450cal.wav	- 50 caliber machine gun firing
250calld.wav \ 450calld.wav	- 50 caliber machine gun loading
250cal_1.wav \ 450cal_1.wav	- 50 caliber machine gun firing
250cal_2.wav \ 450cal_2.wav	- 50 caliber machine gun firing
250cal_4.wav \ 450cal_4.wav	- 50 caliber machine gun firing
250cal_5.wav \ 450cal_5.wav	- 50 caliber machine gun firing
250cal_7.wav \ 450cal_7.wav	- 50 caliber machine gun firing
2aaahhhh.wav \ 4aaahhhh.wav	- a man yelling
2ak47.wav \ 4ak47.wav	- AK-47 machine gun firing
2alarm.wav \ 4alarm.wav	- single alarm sound
2baa.wav \ 4baa.wav	- sheep noise
2bigbang.wav \ 4bigbang.wav	- explosion sound
2boilrrm.wav \ 4boilrrm.wav	- a man saying "class bravo fire, boiler room"
2boom3.wav \ 4boom3.wav	- explosion sound
2bump.wav \ 4bump.wav	- a man saying "Ummph"
2buzz1.wav	- a low pitch buzzing sound
2buzz2.wav	- a high pitch buzzing sound
2bycmnd.wav \ 4bycmnd.wav	- a robot saying "by your command"

2cal50c3.wav \ 4cal50c3.wav	- 50 caliber machine gun firing
2cal50c5.wav \ 4cal50c5.wav	- 50 caliber machine gun firing
2cal50c6.wav \ 4cal50c6.wav	- 50 caliber machine gun firing
2cal50c7.wav \ 4cal50c7.wav	- 50 caliber machine gun firing
2cannon1.wav \ 4cannon1.wav	- cannon firing sound
2cannon2.wav \ 4cannon2.wav	- cannon firing sound
2cannon3.wav \ 4cannon3.wav	- cannon firing sound
2cannon4.wav \ 4cannon4.wav	- cannon firing sound
2cannon5.wav \ 4cannon5.wav	- cannon firing sound
2cannon6.wav \ 4cannon6.wav	- cannon firing sound
2cannon7.wav \ 4cannon7.wav	- cannon firing sound
2ceasfir.wav \ 4ceasfir.wav	- a man yelling "Cease Fire"
2clr2fir.wav \ 4clr2fir.wav	- a man yelling "Clear to Fire"
2clr_min.wav \ 4clr_min.wav	- clearing a minefield explosion
2combo1.wav	- combination winding up sound effect
2cow.wav \ 4cow.wav	- a cow mooing
2crash.wav \ 4crash.wav	- vehicle crashing sound
2dism16a.wav \ 4dism16a.wav	- distant M-16 machine gun battle
2dism16b.wav \ 4dism16b.wav	- distant M-16 machine gun battle
2dolbthx.wav \ 4dolbthx.wav	- trademark Dolby sound
2dragon.wav \ 4dragon.wav	- Dragon missile explosion sound
2engaget.wav \ 4engaget.wav	- a man saying "engage that right target, over"
2engsnds.wav \ 4engsnds.wav	- engine sound
2enterer.wav \ 4enterer.wav	- man saying "follow Jack into the engine room"
2ernoise.wav \ 4ernoise.wav	- engine sound
2explsn1.wav \ 4explsn1.wav	- explosion sound
2explsn2.wav \ 4explsn2.wav	- explosion sound
2expolsn.wav \ 4expolsn.wav	- explosion sound

2fireout.wav \ 4fireout.wav	- man saying "fire's out, set the reflash watch"
2firstm1.wav \ 4firstm1.wav	- M-1 tank main gun firing
2flash1.wav	- flash sound effect
2follwme.wav \ 4follwme.wav	- a man saying "Follow Me"
2frstm16.wav \ 4frstm16.wav	- single M-16 rifle shot
2gq.wav \ \ 4gq.wav	- ship's general quarters alarm
2gqallhd.wav \ 4gqallhd.wav	- ship's general quarters alarm with a man saying "class bravo fire, boiler room, all hands general quarters"
2grenade.wav \ 4grenade.wav	- grenade explosion sound
2grind1.wav \ 4grind1.wav	- large object grinding sound
2halonon.wav \ 4halonon.wav	- man saying "Halon activated, evacuate space immediately"
2hatchpn.wav \ 4hatchpn.wav	- hatch opening
2helicpt.wav \ 4helicpt.wav	- helicopter engine sound
2in_humm.wav \ 4in_humm.wav	- noises inside of a moving HUMMV
2jackdmo.wav \ 4jackdmo.wav	- a man saying "start demonstration"
2jackdne.wav \ 4jackdne.wav	- a man saying "Jack demo completed"
2ldrwell.wav \ 4ldrwell.wav	- footsteps on a ladderwell
2m150cal.wav \ 4m150cal.wav	- M-1 tank 50 caliber machine gun firing
2m16.wav \ 4m16.wav	- M-16 machine gun firing
2m1coax1.wav \ 4m1coax1.wav	- M-1 tank coax machine gun firing
2m1coax2.wav \ 4m1coax2.wav	- M-1 tank coax machine gun firing
2m1coax3.wav \ 4m1coax3.wav	- M-1 tank coax machine gun firing
2m1idle.wav \ 4m1idle.wav	- M-1 tank engine idling
2m1idlef.wav \ 4m1idlef.wav	- M-1 tank engine fast idle
2m1idleh.wav \ 4m1idleh.wav	- M-1 tank engine high idle
2m1main1.wav \ 4m1main1.wav	- M-1 tank main gun firing

2m1main2.wav \ 4m1main2.wav - M-1 tank main gun firing

2m1main3.wav \ 4m1main3.wav - M-1 tank main gun firing

2m1main4.wav \ 4m1main4.wav - M-1 tank main gun firing

2m1move1.wav \ 4m1move1.wav - M-1 tank moving

2m1trck1.wav \ 4m1trck1.wav - M-1 tank track sounds

2m1trck2.wav \ 4m1trck2.wav - M-1 tank track sounds

2m1_coax.wav \ 4m1_coax.wav - M-1 tank coax machine gun firing

2m60.wav \ 4m60.wav - M-60 machine gun firing

2m601.wav \ 4m601.wav - M-60 machine gun firing

2m602.wav \ 4m602.wav - M-60 machine gun firing

2m603.wav \ 4m603.wav - M-60 machine gun firing

2m604.wav \ 4m604.wav - M-60 machine gun firing

2m605.wav \ 4m605.wav - M-60 machine gun firing

2machgn1.wav \ 4machgn1.wav - machine gun firing

2mark191.wav \ 4mark191.wav - Mark19 machine gun firing

2mark192.wav \ 4mark192.wav - Mark19 machine gun firing

2mark193.wav \ 4mark193.wav - Mark19 machine gun firing

2mark194.wav \ 4mark194.wav - Mark19 machine gun firing

2mark195.wav \ 4mark195.wav - Mark19 machine gun firing

2missle.wav \ 4missle.wav - missile firing sound

2missle1.wav \ 4missle1.wav - missile firing sound

2missle2.wav \ 4missle2.wav - missile firing sound

2missle3.wav \ 4missle3.wav - missile firing sound

2mmbeer.wav \ 4mmbeer.wav - Homer Simpson saying "mmmm, beeeerrr"

2mortr81.wav \ 4mortr81.wav - 81mm mortar explosion

2nozzle.wav \ 4nozzle.wav - water nozzle whoosh sound

2releas1.wav - release sound effect

2rifle.wav \ 4rifle.wav - single rifle shot

2roesthm.wav \ 4roesthm.wav	- John Roesli's NPSNET theme song
2roger.wav \ 4roger.wav	- a man saying "Roger"
2sayagan.wav \ 4sayagan.wav	- a man saying "Say Again"
2shut1.wav	- shut sound effect
2shut2.wav	- shut sound effect
2shutdwn.wav \ 4shutdwn.wav	- a man saying "sound server, deactivated"
2sniper.wav \ 4sniper.wav	- a single rifle shot
2splash1.wav	- water splashing/fizzing sound effect
2startup.wav \ 4startup.wav	- a man saying "sound server, activated"
2step.wav \ 4step.wav	- a footstep sound effect
2tank.wav \ 4tank.wav	- tank main gun firing
2tankdep.wav \ 4tankdep.wav	- tank main gun firing
2thatcol.wav \ 4thatcol.wav	- Beavis saying "That was cool."
2turn1.wav	- turn sound effect
2turn2.wav	- turn sound effect
2uhh.wav \ 4uhh.wav	- a man saying "Ummph"
2valve.wav \ 4valve.wav	- valve sound effect
2ventilt.wav \ 4ventilt.wav	- ventilation sound effect
2whoah1.wav \ 4whoah1.wav	- a man saying "Whoah, follow me men!"
2whoahab.wav \ 4whoahab.wav	- a man saying "Whoah, airborne!"
2whoorah.wav \ 4whoorah.wav	- a man saying "Oohrah"
4beach1.wav	- sounds of the beach
4bell1.wav	- sound of a bell toll
4birds1.wav	- bird sounds
4birds2.wav	- bird sounds
4blip1.wav	- blip sound effect
4blip2.wav	- blip sound effect
4blip3.wav	- blip sound effect

4blurp1.wav	- water burbling sound
4boing1.wav	- boing sound effect
4brake1.wav	- rollercoaster brake sound effect
4bus1.wav	- bus engine sound
4car1.wav	- low pitch car engine sound
4car2.wav	- high pitch car engine sound
4chimes1.wav	- high pitch chime sounds
4city1.wav	- city traffic sounds
4clap1.wav	- audience clapping sound
4crash1.wav	- crashing sound
4crash2.wav	- crashing sound
4crowd1.wav	- crowd noise
4crowd2.wav	- crowd noise
4dolphn1.wav	- dolphin noise
4door1.wav	- car door closing sound
4engine1.wav	- low idle, large vehicle engine sound
4engine2.wav	- medium idle, large vehicle engine sound
4engine3.wav	- tracked vehicle engine sound
4engine4.wav	- tracked vehicle engine sound
4engine5.wav	- high idle, large vehicle engine sound
4engine6.wav	- high idle, large vehicle sound
4forest1.wav	- forest sounds
4heli1.wav	- helicopter engine sound
4heli2.wav	- helicopter engine sound
4horn1.wav	- instrumental horn sound
4horn2.wav	- instrumental horn sound
4horn3.wav	- instrumental horn sound
4hum1.wav	- large humming sound

4hum2.wav	- medium humming sound
4jet1.wav	- jet flying sound
4jet2.wav	- jet flying sound
4laser1.wav	- laser sound
4laser2.wav	- laser sound
4laser3.wav	- laser sound
4mcycle1.wav	- motorcycle sound
4mcycle2.wav	- motorcycle sound
4noise1.wav	- nosie sound effect
4plane1.wav	- propeller airplane sound
4plane2.wav	- propeller airplane sound
4quiet1.wav	- faint humming sound
4rumble1.wav	- rumble sound effect
4rumble2.wav	- rumble sound effect
4rumble3.wav	- rumble sound effect
4shot1.wav	- single shot sound effect
4shut1.wav	- shutting sound effect
4siren1.wav	- emergency vehicle siren sound
4siren2.wav	- emergency vehicle siren sound
4siren3.wav	- emergency vehicle siren sound
4spacy1.wav	- space sound effect
4start1.wav	- engine starting sound
4start2.wav	- engine starting sound
4street1.wav	- distant street traffic sound
4street2.wav	- street sound effect
4tire1.wav	- tire on road sound effect
4tire2.wav	- tire on road sound effect
4tire3.wav	- tire on road sound effect

4tire4.wav	- tire on road sound effect
4train1.wav	- railroad train sounds
4train2.wav	- railroad train sounds
4train3.wav	- railroad train sounds
4tram1.wav	- tram car sounds
4truck1.wav	- truck idling sound
4truck2.wav	- truck traveling sound
4truck3.wav	- truck traveling sound
4tumble1.wav	- tumble sound effect
4ufo1.wav	- UFO sound effect
4water1.wav	- splashing water sounds
4water2.wav	- splashing water sounds
4whale1.wav	- whale sounds
4xplsn1.wav	- explosion sound
4xplsn2.wav	- explosion sound
4xplsn3.wav	- explosion sound
4xplsn4.wav	- explosion sound
4xplsn5.wav	- explosion sound
4xplsn6.wav	- explosion sound
4xplsn7.wav	- explosion sound
4xplsn8.wav	- explosion sound
4xplsn9.wav	- explosion sound
welcome.wav	- helicopter engine sound

APPENDIX D: PROPOSED NPSNET SOUND CLASS INTERFACE

initializeSoundDevice()

Synopsis

void **initializeSoundDevice** (const char **datafile*);

Description

Initializes the sound output device and loads the appropriate sound files. For the Acoustetron II, this would entail calling the cre_init() function and reading the acoustetron.dat file, loading into an array all available NPSNET sound files on the Acoustetron II.

Parameters

datafile - the path and filename of the appropriate datafile.

Return Value

None.

Example

initializeSoundDevice (config.search_path);

Notes

None.

shutdown()

Synopsis

void **shutdown** ();

Description

Shuts down the sound output device releasing whatever resources were being used.

Parameters

None.

Return Value

None.

Example

shutdown ();

Notes

None.

loadMasterVehicleSounds()

Synopsis

void **loadMasterVehicleSounds** (const int **vehicle_sounds_array*);

Description

Loads all sounds specific to an NPSNET vehicle.

Parameters

vehicle_sounds_array - an array of integers that contain the integer values of the vehicle's engine, primary weapon, secondary weapon, and round detonation sounds. This function will reserve sound output resources for these sounds and once loaded, will start the continuous, looping replay of the vehicle's engine sound and make ready weapons firing and detonation sounds.

Return Value

None.

Example

```
int tank_sounds[4];
tank_sounds[1] = TANK_ENGINE_SOUND;
tank_sounds[2] = MAIN_GUN_SOUND;
tank_sounds[2] = COAX_GUN_SOUND;
tank_sounds[3] = 50_CAL_GUN_SOUND;
loadMasterVehicleSounds (tank_sounds);
```

Notes

The challenge with this function is to dynamically determine which sounds belong to a particular vehicle once it is identified. Also, if a vehicle has more than one secondary weapon, this will also have to be addressed. For example, an M1A2 tank has a main gun, coax gun and a 50 caliber machine gun as its suite of weapons.

updateMasterVehicleState()

Synopsis

void **updateMasterVehicleState** (const EntityLocation *location*, const EntityOrientation *orientation*, const float *speed*);

Description

This function is responsible for passing along entity state information to the sound output device. In the case of the Acoustetron II, the location and orientation parameters are needed to update the listener's head posture. Speed is needed to determine vehicle engine pitch in some cases.

Parameters

location - the location of the vehicle in NPSNET's EntityLocation type.

orientation - the orientation of the vehicle in NPSNET's EntityOrientation type.

speed - the speed of the vehicle.

Return Value

None.

Example

updateMasterVehicleState (my_info.location, my_info.orientation, my_info.speed);

Notes

This function was created for the Acoustetron II to pass along crucial vehicle posture information. This function would be needed for the Acoustetron II and MIDI class implementations but not for the mono class implementation.

playMasterVehicleSounds ()

Synopsis

void **playMasterVehicleSounds** (const EntityLocation *location*, const EntityOrientation *orientation*, const float *sound*);

Description

This function is responsible for passing along entity state information to the sound output device. In the case of the Acoustetron II, the location and orientation parameters are needed to update the listener's head posture. Speed is needed to determine vehicle engine pitch in some cases.

Parameters

location - the location of the vehicle in NPSNET's EntityLocation type.

orientation - the orientation of the vehicle in NPSNET's EntityOrientation type.

sound - the sound of the vehicle.

Return Value

None.

Example

playMasterVehicleSounds (my_info.location, my_info.orientation, my_info.speed);

Notes

None.

loadAndPlayEntityVehicleEngineSound()

Synopsis

int **loadAndPlayEntityVehicleEngineSound**(const EntityLocation *location*, const EntityOrientation *orientation*, const float *sound*)

Description

Loads and starts the continuous replay of an NPSNET entity engine sound.

Parameters

location - the location of the vehicle in NPSNET's EntityLocation type.

orientation - the orientation of the vehicle in NPSNET's EntityOrientation type.

sound - the sound of the vehicle.

Return Value

int - the sound resource ID nyumber assigned for the particular sound event. This allows for quick access and update in the updateEntityVehicleState() function where a sound resource ID is required..

Example

entity.soundResourceID = **loadAndPlayEntityVehicleEngineSound** (entity.entity_sound);

Notes

None.

updateEntityVehicleState()

Synopsis

void **updateEntityVehicleState**(const int *entity_ID,* const EntityLocation *location,* const EntityOrientation *orientation,* const int *vehicle_speed*);

Description

Updates the state for an identified NPSNET entity.

Parameters

entity_ID - this is the ID for the entity as assigned by the sound device. This allows for quick lookup of the entity sound resource vice using an expensive sound device query to determine which sound's status to update.

location - the location of the entity in NPSNET world coordinates.

orientation - the orientation of the entity in NPSNET world coordinates.

speed - the speed of the entity.

Return Value

None.

Example

updateEntityVehicleState(entityList[iX].soundResourceID, entityList[iX].location, entityList[iX].orientation, entityList[iX].speed);

Notes

None.

stopAndUnloadEntityVehicleEngineSound()

Synopsis

void **stopAndUnloadEntityVehicleEngineSound** (const int *entity_ID);*

Description

Stops and unloads an NPSNET entity engine sound

Parameters

entity_ID - this is the ID for the entity as assigned by the sound device. This allows for quick lookup of the entity sound resource vice using an expensive sound device query to determine which sound's status to update.

Return Value

None.

Example

stopAndUnloadEntityVehicleEngineSound (entityList[iX].soundResourceID);

Notes

None.

playSound()

Synopsis

void **playSound** (const EntityLocation *location*, const EntityOrientation *orientation*, const int *soundToPlay*);

Description

This function sends the command to the sound output device to play a particular sound at a particular location.

Parameters

location - the location of the vehicle in NPSNET's EntityLocation type.

orientation - the orientation of the vehicle in NPSNET's EntityOrientation type.

soundToPlay - the integer index number of the sound to play.

Return Value

None.

Example

playSound (my_info.location, my_info.orientation, TANK_ENGINE_SOUND);

Notes

This function will be used differently depending on implementation. In addition to the sound parameter, the Acoustetron class will use both location and orientation parameters while the mono class will only use the location parameter. In the original implementation of this function, many different overloaded versions of the function were created to accommodate different requirements. However, this approach leads to confusing implementations of the function. Rather, a common set of parameters should be passed in one definition of the function and have the class implementation determine which parameters are appropriate.

soundIsPlaying()

Synopsis

int **soundIsPlaying** (int *sound*);

Description

A boolean function that returns TRUE or FALSE as to whether a sound is playing or not.

Parameters

sound - the sound to check whether it is playing.

Return Value

fail - FALSE

success - TRUE

Example

if (**soundIsPlaying** (TANK_ENGINE_SOUND)) {

Notes

This function is useful to the Acoustetron II in order to determine a number of instances. For example, if a another player's vehicle is close enough for the listener to hear the other's vehicle engine sound, the appropriate vehicle engine sound is loaded and played in a continuous loop until the vehicle can no longer be heard. While the vehicle is within hearing range, for every DIS Entity State PDU received from that vehicle a check is made to see if the vehicle engine sound is playing. If so, continue with the processing loop. If not, load the vehicle engine sound and start its continuous replay. Another example is if a sound is requested to be played and it has not been loaded, then the sound is loaded, played then unloaded. But before a sound can be unloaded, it must finish playing. Because sounds all vary in length of replay, a boolean function such as this one is needed to check if the sound is still playing.

updateSoundDevice()

Synopsis

void **updateSoundDevice** ();

Description

Performs any periodic updates that is required by the sound output device.

Parameters

None.

Return Value

None.

Example

updateSoundDevice ();

Notes

This function is class implementation dependent. For example, the Acoustetron II function cre_update_audio() at every iteration of a processing loop. Similar requirements may need servicing on other sound output devices.

stopAllSounds()

Synopsis

void **stopAllSounds** ();

Description

Stops all sounds that are currently playing on the sound output device.

Parameters

None.

Return Value

None.

Example

stopAllSounds ();

Notes

None.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Rd.
   Quantico, VA 22134-5027

4. Director, Marine Corps Research Center 2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, VA 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC, Code C45
   300 Russell Road
   Quantico, VA 22134-5130

6. Chairman, Code 32 CS/Lt 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. Dr. Michael J. Zyda, Code 32 CS/Zk 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943-5000

8. John S. Falby, Code 32 CS/Fa 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943-5000

9. Russell Storms, Code 32 CS/St 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943-5000

10. Captain Lloyd J. Biggs, USMC 1
United States Marine Corps
5063 Palmera Drive
Oceanside, CA 92056

11. Lieutenant Commander Edmund L. Biggs 1
United States Navy (ret.)
11 Schutt Court
Grand Island, NY 14072

12. Master Sergeant Charles A. Miller 1
United States Marine Corps (ret.)
1166 Avenida Sobrinal
Oceanside, CA 92057