bers are used much like MIDI channel numbers.

Synchronization

To support the choreography and ventriloquism of virtual sources, the server must provide mechanisms by which programs can query the server for status information and synchronize on points in the audio stream(s). This synchronization can be provided by an RPC callback mechanism. Many commands, especially movement commands, should include options for automatic synchronization with an audio stream.

Distributed Design

One can distribute the work of the server over several workstations by running a server on each machine, with each server processing a subset of the physical sources. Each server is given the same stream(s) of control information, and the outputs of the servers are summed to give a final signal.

High Level Interface

The high level interface for a VR system would allow sound sources to be attached to objects and then updated transparently as the user interacts with the environment. Several extensions have been made to the SVE library, to initialize and exit the spatial sound control channels and for attaching a virtual sound source to an arbitrary object. The eventloop-manager automatically updates position to the audio server. Environmental acoustics should also change as the user moves through the environment. Automatically updating these requires the definition of “areas” within the SVE environment.

FUTURE DIRECTIONS

Once all of the mechanisms of the spatial audio server are in place, there will be a new set of questions concerning the policies which should govern their use. For example, we’d like to be able to prioritize audio sources, but still need to determine what priority schemes are actually useful and under which conditions. We will also need to determine a suitable means for programmers to specify the acoustical properties of environments and audio sources.

Furthermore, the specification and choreography of auditory elements will remain a tedious task. The VR research community must develop high-level modeling methods and tools: world builders should support specification techniques for both graphics and sound, as the visual and auditory elements are quite related to each other.

ACKNOWLEDGEMENTS

Work on the Mercator project has been supported by Sun Microsystems. Additional funding, specifically for spatial sound research, has also been provided by Sun.

The authors would also like to thank Elizabeth Mynatt of the Georgia Tech GVU Center and Mark Lee of the Georgia Tech Psychology Department.

REFERENCES

1. Birrell, A.D., Nelson, B.J. (1984) Implementing remote procedure calls, ACM Transactions on Computer Systems, Vol. 2, No. 1, Feb. 1984, pp. 39-59.

2. Blauert, J. (1983) *Spatial Hearing: The Psychophysics of Human Sound Localization*, MIT Press: Cambridge, MA.

3. Borish, J. (1984) Extension of the image method to arbitrary polyhedra, J. Acoust. Soc. Am., 75, 1827-1836.

4. Burgess, D.A. (1992) Techniques for low cost spatial audio, Proceedings of the Fifth Annual Symposium on User Interface Software and Technology (UIST ‘92), ACM, New York, 1992, pp. 53-59.

5. Liang, J., Shaw, C., Green, M., (1991) On temporal-spatial realism in the virtual reality environment, Proceedings of the Fourth Annual Symposium on User Interface SOftware and Technology (UIST ‘91), ACM, New York, 1991, pp.19-26.

6. Postel, J. (1980) User datagram protocol, RFC 768, Network Information Center, SRI International, Menlo Park, Calif., August 1980.

7. Verlinden, J.C., Kessler, A., Hodges, L.F. (1993) The Simple Virtual Environment (SVE) Library: Users Guide, Tech Report GIT-GVU-93-24.

8. Wenzel, E.M. (1992) Localization in virtual acoustic displays, Presence, 1, 80-107.

the low-level interface is that parameters are computed to give the best possible acoustic models with the available hardware and computing power.

The low-level interface also contain drivers to actually transfer operating parameters to the hardware and software of the audio processing system.

Mid-Level Interface

The purpose of the mid-level interface is to manage the limited resources of the physical channels and to provide mechanisms for the control and synchronization of the physical sources. Most of the functionality of the mid-level interface will be in the spatialization host (server). A client (such as a user interface or VR system) sends commands to the spatialization server via an RPC[1] protocol. Although RPC's are synchronous, the commands themselves are not—the return of the RPC simply means that a request has been acknowledged, not that any actual action has occurred. However, some commands can specifically provide synchronization. Along with the RPC-based control channel, a faster, less reliable UDP connection can be used to communicate head-tracking information in real-time.

Resource Management

The virtual environment may contain an arbitrary number of virtual sources, but in any given system only a finite number of physical sources and channels are available to generate and process sound. There are two schemes available for dealing with this problem: clustering and prioritizing.

If a number of virtual sources are clustered together at a distance from the user, the difference in position between sources in the cluster may be below the localization resolution of the human auditory system, a phenomena called localization blur. The user will perceive the sounds as all coming from the same place. If source positions are blurred like this, then audio streams for the sources need not be processed separately. The audio streams from the physical sources can be mixed before spatialization and environmental effects are computed. After mixing, these expensive effects are only computed once for the entire cluster of sources, freeing up hardware for other physical channels.

1. Remote Procedure Call (Birrell & Nelson, 1984). RPC routines allow C programs to make procedure calls on other machines across a network. First, the client calls a procedure to send a request to the server. Upon receipt of the request, the server calls a dispatch routine to perform the requested service, and then sends back a reply. Finally, the procedure call returns to the client.

After clustering, each cluster and remaining single source is assigned some rank on the basis of distance, intensity, and programmer-supplied priority. The highest ranking sources get access to the physical channels. Lower ranking sources can then be either mixed into the audio output with no special processing, can be processed with less expensive acoustic models, or can simply be silent until their ranking improves. Normally, these silent audio streams from low ranking virtual sources should still be updated just as if they had been played, otherwise low priority audio channels would be constantly be out of synchronization with their accompanying visual cues and the number of active sound sources would grow continuously over time.

The mid-level interface can also incorporate schemes to reduce the amount of control overhead due to head motion. An obvious approach is to update head tracker position only when it is changing. We can extend this idea by applying hysteresis conditioning to the inputs. With this type of conditioning, a position parameter is not updated unless it changes by more than some given threshold from the last value used.

To use hysteresis conditioning to its fullest advantage, we must use the widest thresholds possible. In our testbed system, we use a threshold of 1˚ for both azimuth and elevation, which does not seem to affect the user's perception of a source's position. Previous experiments measuring localization blur (Blauert, 1983) suggest that azimuth thresholds as large as 3˚ can be used directly in front of and behind the listener, and as large as 9˚ to the far right and far left. Elevation thresholds of 4˚ may be adequate near the median plane, and much larger thresholds might be possible for higher elevations.

We can also apply hysteresis conditioning to the distance of a source. We have found that it is more effective for the distance threshold to be a fraction of its current value rather than some absolute offset from the current value. For our testbed system, we use a distance threshold of 10%, which gives an illusion of continuous motion along an axis away from the listener.

In addition to hysteresis conditioning, Kalman predictive filtering might be applied to tracker data to decrease the effective latency of the system by as many as three tracker sampling periods (Liang, Shaw & Green, 1991).

Physical Sources

The mid-level interface also controls the physical sources in the forms of files, algorithms, and external hardware. The physical source is defined at initialization and assigned a tag. Once assigned, the programmer simply refers to the source by this tag and the mid-level interface translates the tag into a buffer address, MIDI channel number, or what ever sort of source identification is needed by the hardware. When controlling sources and changing source attributes, these tag num-

Doppler Shift

This is the well known pitch-shift caused by the source's motion relative to the listener.

Field Pattern

This is the dependency of the source's volume and spectrum on its orientation relative to the listener. For example, the interactivity of our testbed system could be improved if the virtual radio were louder from the front than from the back.

Environmental Effects

These are the effects of the listening space. Environmental effects are important for the listener to have a sense of the distance of a sound source and an impression of the size and composition of a space.

Early Echoes

These are the first reflections to reach the listener, and are perceived as distinct echoes from distinct directions. For some given geometry, surface reflection, source position, and listener position, we can compute the direction, delay, spectrum, and attenuation of each early echo using the method of images (Borish, 1984). Once these are known, each early echo can be modelled using a delay line, a filter, and an HRTF model. The required accuracy of the HRTF model for realistic and useful early echoes is not yet known, and the topic is open for further research.

Dense Reverberation

In an enclosed space, echo density increases over time while echo intensity decreases. Eventually, individual echoes are too quiet to be heard, but are packed so densely in time that sum is still audible. This situation is called dense reverberation. Dense reverberation greatly improves externalization—the sense that a sound is coming from outside of the head. The ratio of direct to reverberant energy is also an important distance cue.

Dispersion

In any given medium, the speed and attenuation of an acoustic wave are functions of its frequency. For example, in air, high frequencies are attenuated more rapidly than low frequencies, causing distant sources to sound muffled. This effect can make dispersion a useful distance cue in large listening spaces.

We propose that environmental effects be further divided into two stages: local effects and gross effects. For example, consider an environment with many rooms and a sound source in a different room from the listener. Local effects would model the room where the sound source acts, and gross effects would model the space between that room and the listener. Although this division of labor may not result in completely accurate acoustic models, it is computationally tractable and requires only a fixed amount of specification data.

SERVER ARCHITECTURE

The VR audio server is a system of software for connecting the virtual environment (or other application) to the facilities of the physical sources and channels. Its architecture is designed in three levels (see figure 2). The lowest of these levels is heavily dependent on the implementation of the physical channel. The highest level is application dependent. The middle level is application and hardware independent and provides the bulk of the server's functionality.

Low-Level Interface

The purpose of the low-level interface is to provide a common control protocol to the various types of possible physical channels and sources. For example, given the positions and the source and listener, and the geometry of the listening space, the low-level interface computes all of the parameters required by the physical channel for spatialization and local environmental effects and then transfers the parameters to the signal processing hardware or software. The exact nature of this required parameters will depend on the implementation of the physical channels.An important principle of

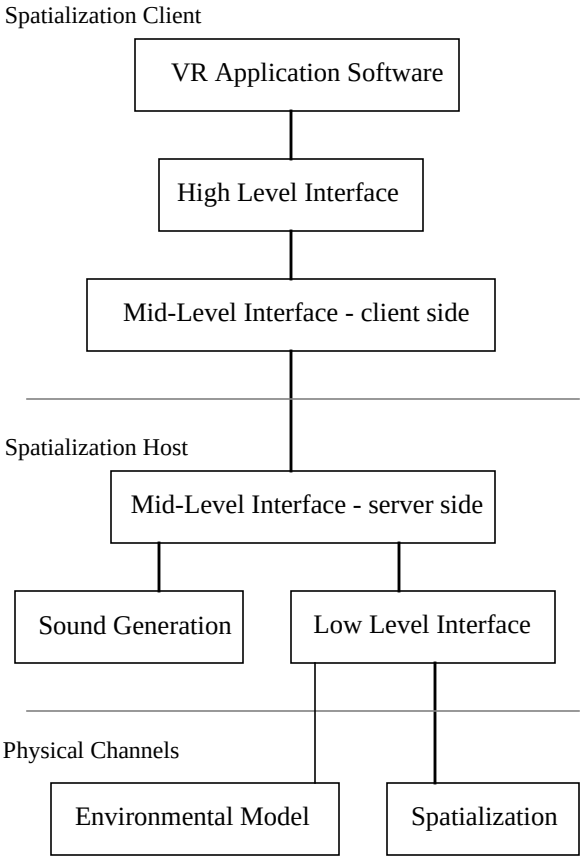

Figure 2. Organization of spatial sound control software

tiveness of the spatial audio is somewhat diminished. Hardware limitations of the HMD, such as weight, low resolution, and narrow field of view seemed to weaken any augmentation of the spatial cues. However, the application provides few visual cues, with an extremely low level of detail, and lacking sufficient optic flow and stationary features. Because there were no dynamic features on the radio, like moving level meters, we did not have the benefit of ventriloquism, either.

Performance

Previous work with audio in virtual environments suggests that an update period of 90ms is adequate to give an illusion of continuous motion for angular velocities less than 360˚/sec (Wenzel, 1992). The testbed system generates a position update every 50ms, although the latency for a position update is about 80ms. As a result, the system produces smooth audio source motion, even though there may be a perceptible lag between a user action and the movement of the virtual sound source.

Reliability

The two weaknesses in the audio system are the UDP-based control link and the interference of the headphones with the magnetic trackers.

Software for the UDP control link does not check for out-of-order or incomplete packets. Surprisingly, the control link rarely fails, and then only during periods of unusually heavy network use. Because the spatialization host keeps no state information, it recovers from errors quickly.

We also found that the magnetic fields generated by the headphones disrupt the operation of the trackers when a tracking sensor is closer than 10cm to a headphone earpiece (Sennheiser HD540 headphones). Ferrous shielding plates inside the HMD may be a solution to this problem.

PROGRAMMING SUPPORT REALIZATIONS

An important question we hope to explore with this testbed system is that of how to control a spatial audio system in real time. The software for the testbed system was hand-built for a simple set of requirements, but for spatial audio to be more than a toy, programmers must be able to integrate it into other complex software systems with minimal effort.

In working with the testbed system, we came to the conclusion that an audio server for VR should manage the entire acoustic environment, much like our existing virtual worlds package manages the visual environment.

PHYSICAL SOURCES

A physical source is the hardware or software that generates sound prior to the application of the effects of orientation, movement, and environment. The sound from a physical source is independent of the listener or the listening space. A physical source may be sampled sound on a disk, it may be a program, or it may be an external device such as a MIDI synthesizer. Physical sources may have attributes that can be changed over time for the purposes of sonification.

PHYSICAL CHANNELS

The physical channel is the set of hardware and/or software that takes an audio stream from a physical source and converts it into a final form ready for presentation to the listener. To generate audio for a virtual environment, the outputs from the set of physical channels are summed and presented to the user via headphones or speakers. Although the actual implementation of such a channel is site-dependent, in general, each physical channel produces two types of effects: source-relative and environmental. (We must stress that this division of effects does not necessarily correspond to a division of hardware tasks.)

Source-Relative Effects

These are effects due to the relationship between the listener and the source that are not greatly influenced by the listening environment.

Spatialization

Spatialization means the application of filters modelling the HRTF. The sophistication of these filters may vary with available computing power.

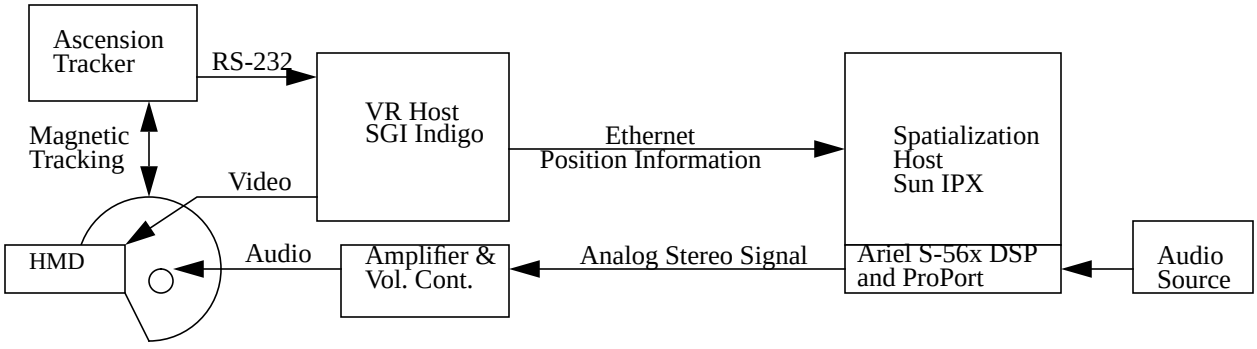

Figure 1. Hardware for testbed virtual environment with spatial audio

An Architecture for Spatial Audio Servers

David A. Burgess and Jouke C. Verlinden

Graphics, Visualization, and Usability Center
Georgia Institute of Technology

ABSTRACT

Before spatial sound can be useful in large software systems, there must be a sufficient application programmer's interface (API) for the control of the spatial audio system and for the management of finite audio processing resources. Researchers are using virtual environments as driving applications to determine the requirements of such an interface.

INTRODUCTION

As spatial audio becomes more widely used in virtual environments, we realize the need for better software support for sound. Audio servers for VR need to provide more than a simple ability to play sounds; they must be able coordinate sounds from diverse sources, provide synchronization facilities, and model the acoustic properties of the virtual environment.

To explore some of the design issues of VR audio servers, we have built a testbed VR system with spatial audio capabilities.

TESTBED VR SYSTEM

The testbed for auditory virtual environments was constructed from existing hardware and software in the Georgia Tech Graphics, Visualization, and Usability Center (GVU Center).

The GVU Center VR system consists of a Virtual Research HMD, a Virtual Research Cyberglove, and a dual-receiver Ascension Technologies Bird magnetic tracker system connected to an SGI Indigo Elan workstation. An extensive library, called SVE, has been written to support the construction of virtual environments with this hardware (Verlinden, Kessler & Hodges, 1993).

The GVU Multimedia Computing Group has developed a low-cost spatial audio system (Burgess, 1992). The spatialization system operates by applying pairs of digital filters to an audio stream[1]. Additional processing can be applied to generate other effects, such as dense reverberation. The system hardware consists of an Ariel S-56x digital signal processing (DSP) board, a SPARCstation IPX host machine, and an Ariel ProPort 656 for digital to analog conversion. DSP microcode and low-level utilities are available for controlling the spatialization engine.

A set of UDP-based[2] procedures were written to allow the VR host to transmit tracker positions to the spatial sound system through a local area network. The VR host sends a packet specifying the source and listener positions. From this packet, the spatialization host generates a pair of spatialization filters and sends them to the Ariel DSP board. The DSP board samples and processes an analog audio source with the specified filters. Processed audio is then sent back to the VR area in analog form through two coaxial cables. The system is diagrammed in figure 1.

OBSERVATIONS OF THE TEST SYSTEM

Using the resulting VR system, a demonstration application was developed. We present some observations based on experience with the testbed system.

Visual Cues and Immersion

The first testbed application was simple; the virtual world consisted of a green field and a hand-held radio. A virtual sound source was attached to this radio; as the user moved the cursor (hand-held tracker), he or she heard the sound source moving as well. The audio system supported spatialization, dispersion, and dense reverberation.

When first introduced to this virtual world, a user may not immediately perceive a connection between the visual cue and the virtual audio source. There is a brief training period involved, and experiments are planned to determine the typical length of adaptation.

Our hope that the visual cues would improve the effec-

---

1. This filter pair is designed to approximate the Head-Related Transfer Function (HRTF), which a set of effects imposed on sounds by the shape of the outer ears, head, and upper body (Blauert, 1983). By artificially generating these effects with sufficient accuracy, it is possible to create the sense of a sound coming from a particular direction.
2. UDP is a simple, unreliable datagram protocol which is layered directly above the Internet Protocol (IP). (Postel, 1980)