EFFICIENT AND EFFECTIVE USE OF LOW-COST 3D AUDIO SYSTEMS

Kenneth Wang, Venkataraman Sundareswaran, Clement Tam, Philbert Bangayan[1]

Rockwell Scientific
1049 Camino Dos Rios
Thousand Oaks, CA 91360 USA
kkwang@rwsc.com

Pavel Zahorik

Waisman Center
University of Wisconsin – Madison
Madison, WI 53705 USA
zahorik@waisman.wisc.edu

ABSTRACT

Commercial, off-the-shelf (COTS) 3D sound cards offer a readily available low-cost option to consumers, researchers, and developers of 3D auditory displays. However, drawbacks of current 3D sound cards include computing platform support limitations, Application Programming Interface (API) complexity, vendor instabilities, and the lack of individualized Head-Related Transfer Functions (HRTFs). To address these issues, we have developed a client/server system utilizing COTS 3D sound cards and have investigated a method of visual-feedback training for 3D sound localization.

1. INTRODUCTION

In an ideal 3D audio environment, a listener perceives sound sources as if they are emanating from designated locations in 3D space. As a human-computer interface modality, 3D audio can be used to indicate spatial locations, including those outside the current visual field of the user. Thus, visual display clutter as well as the time required for a user to interpret a cue may be reduced by utilizing a 3D auditory display in conjunction with or in place of a visual display. In addition, the ability to differentiate multiple concurrent sound sources may be increased through 3D audio spatialization. Researchers and developers are investigating the application of 3D audio for improving human-computer interfaces, virtual and augmented reality, and communications.

A 3D audio system can be implemented by incorporating the three basic spatial hearing cues: Interaural Time Difference (ITD), Interaural Intensity Difference (IID), and spectral cues. The ITD is the phase difference of a sound signal between the left and right ears of the listener, whereas the IID is the amplitude difference of a sound signal between the two ears. Spectral cues result from the effects of a sound signal reflecting off of the listener's head, shoulders, and pinnae (outer ears) [1]. In order to characterize all three cues empirically, Head-Related Transfer Functions (HRTFs) are often measured by placing probe microphones inside the ear canals of a human or mannequin situated inside an anechoic chamber using standard system-identification techniques. The measurement signal is sequentially transmitted through an array of speakers positioned at a constant distance but varying azimuths and elevations around the subject in 3D space [2]. The inputs to an HRTF-based 3D audio system include the sound source signals (one signal per sound source), the listener's position and orientation, and the positions of the sound sources. An HRTF corresponding to a sound source's azimuth and elevation relative to the listener is determined through interpolation of the spatially nearest measured HRTFs. The sound source signal is convolved with the HRTF (which contains ITD, IID, and spectral cues) and further, in many implementations, a simple gain-based distance model is applied. The resulting 3D audio signal is presented to the listener over a pair of headphones or two speakers. In presentation over speakers, an additional algorithm is applied to cancel crosstalk between the left speaker and the right ear and the right speaker and the left ear.

Research in the use of 3D auditory displays typically requires the use of a system incorporating HRTFs and real-time updates of the position and orientation of the listener and multiple concurrent sound sources. A custom-built 3D audio system offers the researcher the ability to tailor the system to his/her own needs, modify and expand the system as needed, and utilize HRTFs specific to each listener. However, developing and maintaining such a system is often prohibitive in terms of time and cost. Measuring individualized HRTFs requires an anechoic chamber, and building a real-time HRTF-based 3D audio system requires significant expertise in digital signal processing, computer programming, and digital audio. An alternative to custom-built systems is COTS 3D sound card systems which have been widely available since 1997. These cards are marketed primarily to the computer gaming community and are priced at the US$100 range. Typical COTS 3D audio systems support up to 32 concurrent HRTF-spatialized sound sources based on 44.1kHz-sampled Pulse-Coded Modulation (PCM) audio, operate in real-time with at least 30 updates per second, and support some level of environment modeling. In order to incorporate 3D audio into a software application, a developer utilizes a C++ API to specify the PCM audio data, sound source and listener positions and orientations, and environment parameters.

While COTS 3D sound systems offer advantages in cost and convenience over a custom-built system, they suffer from significant limitations. To date, the Windows/Intel ("Wintel")-based PC is the only platform supported by COTS 3D sound systems. In addition to being platform-limited, current COTS 3D sound systems do not inherently offer distributed software architecture support. A distributed architecture is necessary when an application requires more computing resources than are available in a single Wintel PC (e.g., an application's 3D graphics rendering engine may be very resource-intensive and not leave adequate amounts of CPU cycles or memory for 3D

[1] Current affiliation of Bangayan: MIT

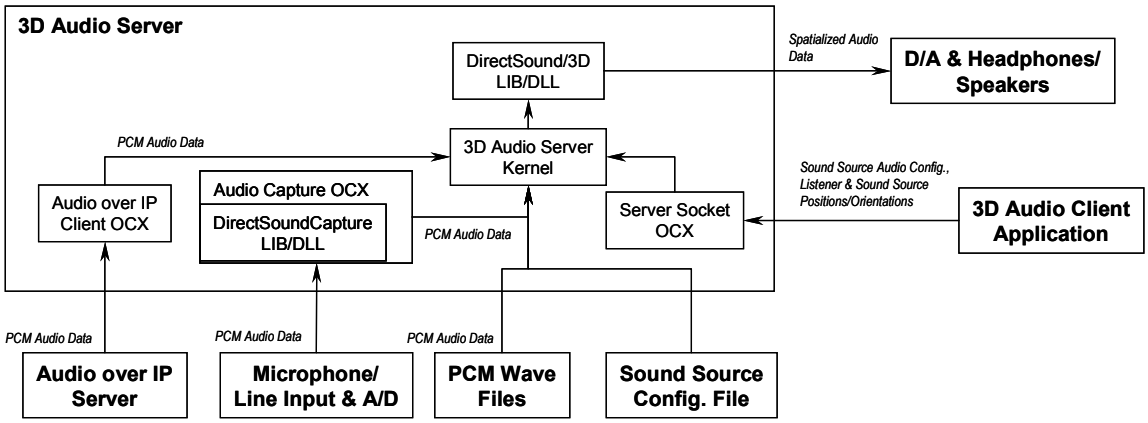

Figure 1. *3D audio server architecture.*

sound processing). Furthermore, while Microsoft has made an attempt to standardize an audio API in the form of DirectSound, not all features or systems are supported under this API (e.g. environment modeling); this requires developers to learn and program in multiple APIs in order to exploit all available features. Maintaining a COTS-based 3D audio system is further complicated by the instability of 3D audio vendors in a volatile PC industry. A developer may add support for a 3D audio system to an application only to find the company no longer in existence a short time later, rendering the audio code obsolete. Several software-based audio rendering systems [3][4][5] are currently under development and offer low-cost 3D audio functionality. However, these systems do not conform to a single API and may not offer capabilities available in COTS 3D audio systems such as Sensaura's Virtual Ear HRTF library [6]. In order to support 3D audio in non-Wintel-based applications, to support distributed software architectures, and to shield application developers from API and vendor instabilities, we have developed a "3D audio server".

2. 3D AUDIO SERVER

Features of the Rockwell Scientific Company (RSC) 3D audio server include:

- Capability to integrate 3D audio functionality into applications executing on any platform supporting Transport Control Protocol/Internet Protocol (TCP/IP).
- Support of all DirectSound-compatible PC sound devices, including those integrating hardware-accelerated real-time HRTF-based 3D audio systems from Sensaura and Aureal.
- Support of pregenerated wave file sound source signals:
  - Arbitrary sampling rate
  - 8 or 16-bit resolution
  - Monaural or stereo
- Support of live sound source signals captured through microphone/line input:
  - 44.1/22.05/11.025 kHz sampling rate
  - 8 or 16-bit resolution
  - Monaural or stereo
- Support of live sound source signals streamed over IP:
  - Arbitrary sampling rate
  - 8 or 16-bit resolution
  - Arbitrary number of channels
- Support of up to 32 concurrent sound sources (dependent upon choice of sound hardware).
- Support of over 30 updates/sec.

The RSC 3D audio server is implemented modularly through the use of ActiveX controls (a Microsoft software component object technology) and utilizes the DirectSound API for audio spatialization as well as live capture. The server socket, audio capture, and audio streaming functionalities are encapsulated into individual ActiveX controls. Figure 1 depicts the 3D audio server architecture.

In addition to hardware and software required to support a COTS 3D sound card, an Ethernet network supporting IP and a network interface card (NIC) are required for use of the server with remote clients or sound sources streamed live over IP. While the RSC 3D audio server runs on the Microsoft Windows platform and audio is spatialized by and output through the sound card located in the server PC, a local or remote client application may control the 3D audio display by providing real-time listener and source position and orientation data to the server over a TCP socket connection. A simple ASCII protocol is used for communication between the client and server and is largely independent of the particular 3D audio system being employed by the server. This software architecture allows any number of individual application developers to simply and rapidly incorporate 3D sound into an application without being concerned with the lower-level intricacies of a particular 3D audio API. When a 3D audio vendor emerges with a new API, we, as the developers of the 3D audio server, will add support for the new system to the server, while no code changes to the client applications will be necessary. A bracket-delimited socket (BDS) protocol is utilized wherein the open bracket ("[") is used to indicate the beginning and closed bracket ("]") to indicate the end of a message, and a handshake is performed to verify that the client and server utilize compatible versions of the BDS protocol. After a 3D audio client/server protocol handshake, the following commands are sent from the client to the server in order to initialize the audio system (it is not always necessary to issue *all* of the commands – several of the parameters have default values):

```
3DASetConfig
3DASetDistanceFactor
3DASetDopplerFactor
3DASetRolloffFactor
3DASetHFFactor
```

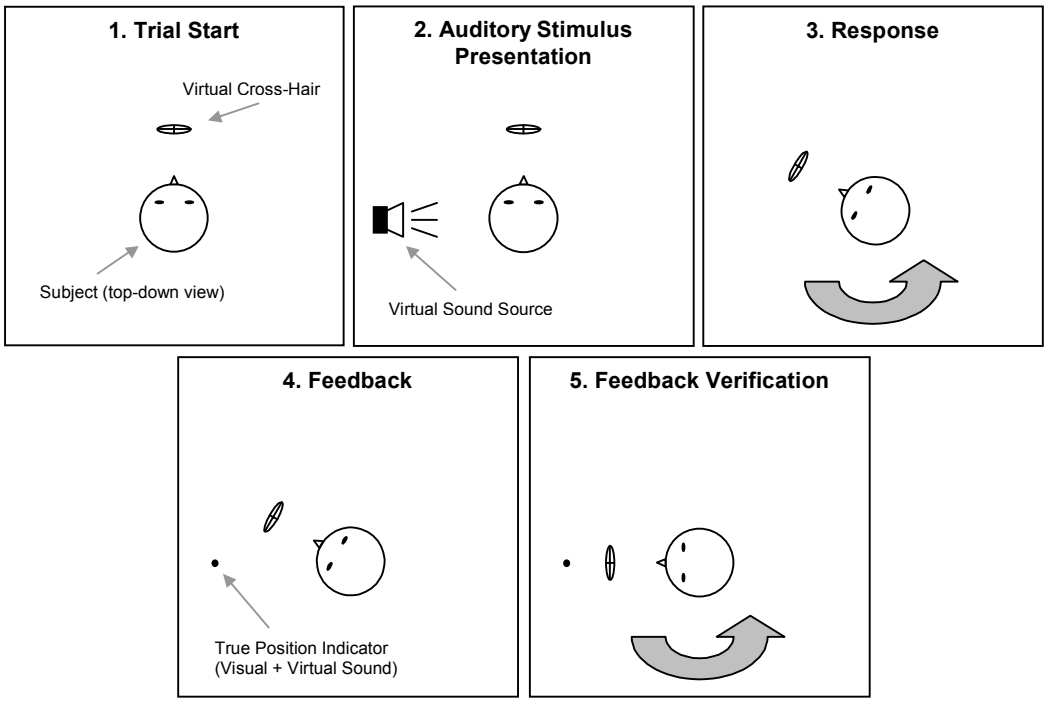

Figure 2. *Experimental procedure for a single trial. During the pre-test and post-test phases of the experiment, only steps 1-3 were implemented. During the training phases, all steps were implemented.*

```
3DASetListenerPosition
3DASetListenerOrientation
3DASetListenerDopplerVelocity

3DACreateSource (Server responds to
  client with 3DASourceID)
3DASetSource3DMode
3DASetSourceVolume
3DASetSourceMaxDist
3DASetSourceMinDist
3DASetSourceDopplerVelocity
3DASetSourceConeAngles
3DASetSourceConeOutsideVol
3DASetSourceConeOrientation
3DASetSourceTimePosition
3DASetSourcePosition
3DAPlaySource
3DASetSourcePan
```

Source command calls are repeated for each desired source. In a typical client application, the following commands are sent to the server from within the main application loop (once per frame):

```
3DASetListenerPosition
3DASetListenerOrientation
3DASetSourcePosition
3DAUpdateAudio
3DASetSourcePan
```

Source audio may be configured immediately prior to runtime. In collaboration with the client developer, an end user creates a configuration file with a name and in a location referred to by the client application. Sound sources are specified one per line, with each line comprising an identifier and the audio source location. Identifiers are strings which are referred to by the client application. For wave file sources, the audio source location is the full path. For live sources, the audio source location comprises the live source module type immediately followed by the module number, followed by a space and a 1-indexed channel number.

For stereo sources, left corresponds to channel 1 and right to channel 2. AudioCaptureModule (microphone/line input live capture) module type is specified as ACM, while AoIPClient (live streaming over IP) module type is specified as AoIP. Live sources are specified as single channels. Thus, each of the two channels of a stereo line input may be used as a separate source. AoIP streams may contain an arbitrary number of time division multiplexed channels.

The following is an example configuration file:

```
heli          = D:\sounds\wav\heli.wav
catsi1        = AoIP2 1
winshuman     = :\sounds\wav\footsteps.wav
talker2       = ACM1 2
water         = :\sounds\4water1.wav
```

We have used the 3D audio server in several virtual reality, augmented reality, and communications demonstrations as well as 3D audio localization experiments [7]. Possible future enhancements to the RSC 3D audio server include support for environment modeling, Text-To-Speech (TTS)-based sources, and Musical Instrument Digital Interface (MIDI)-based sources. The RSC 3D Audio Server is available to partners of Rockwell Scientific and to U.S. Government agencies.

3. VISUAL-FEEDBACK TRAINING

While the client/server architecture provides an efficient means of developing and maintaining 3D audio-enabled applications, COTS 3D audio systems still suffer from lack of support for individualized HRTFs, instead offering one or a limited number of HRTF sets, typically an averaged set or sets from individuals determined to exhibit accurate sound localization ability ("good localizers"). As a result, sound source localization accuracy is often degraded when compared to the accuracy using real sources, or to higher quality displays using individualized HRTFs. However, a method in which listeners are provided with paired auditory and visual feedback as to the correct sound source location can effectively facilitate what appears to be perceptual re-mapping to modified spatial cues, and therefore could be employed to mitigate technical deficiencies in 3D sound systems due to nonindividualized HRTFs.

The experiment is described here briefly for completeness. Full details can be found in a paper published recently [8]. In the experiment, subjects identified the direction of a sound source presented through headphones using a COTS 3D audio card (Turtle Beach Montego II A3D, with Aureal Vortex2 chipset) by means of the 3D audio server described in the previous section. In addition to headphones, the subjects wore a Head-Mounted Display (HMD) and a six degree-of-freedom (6DOF) head tracker. There were three phases in the experiment: a "pre-test" baseline phase, a training phase, and a final "post-test" phase. In the pre-test phase, each listener's ability to judge the apparent angular position of sound sources was evaluated. No feedback as to the correct sound source position was given in this phase. As such, results from this phase represented a baseline level of localization accuracy for a given listener while using a generalized HRTF. A total of 144 spatial positions were tested, 18 from each of 8 spatial regions. The training phase of the experiment took place after the pre-test phase. This phase was similar to the pre-test phase, except that visual feedback as to the correct source location was now provided to the listeners via the HMD. The procedure for a training phase trial is shown graphically in panels 1-5 of Figure 2. The initial portions of a trial were identical to those of the pre-test phase (panels 1-3). After the listener had input his/her apparent position response (panel 3), the feedback portion of the trial began. A visual indicator of the correct source position paired with a repeating spatialized auditory stimulus (the same stimulus as in panel 2, but repeating) was then displayed to the subject via the HMD (panel 4). To verify that the listener was able to find this indicator, the listener was asked to aim a virtual cross-hair (via head rotation) at the correct position indicator (panel 5). When the listener was confident that he/she had pointed to the indicator as accurately as possible, he/she pressed a button, and the cross-hair position was inferred from the measured head position, just as after the source position judgment (panel 3). Hence, this feedback procedure forced the listener to actively orient to the correct sound source position. The selection procedure for sound source positions was similar to that used in the pre-test phase. The training phase of the experiment lasted two experimental sessions. In each session, the listeners completed a block of 72 trials. After the training phase, listeners completed a final post-test phase of the experiment. This phase was identical to the pre-test phase, and was used to assess lasting effects of the training. The post-test phase was conducted at least four days after completing the requisite training phases of the experiment. The largest improvements in accuracy appeared in the listeners' enhanced abilities to distinguish sources in front from sources behind (a reduction in front-back confusion [9]). Further, these improvements were not transient short-term effects, but lasted at least several days.

The results of this experiment support a training paradigm for end users in which the users could be trained periodically to adapt their perception of 3D audio presented using COTS hardware. They may be trained either while using the system for normal operations or in dedicated sessions. We believe that both of these approaches are tenable, based on the conclusions of our experiment.

4. CONCLUSIONS

We have described methods for improving the efficiency and effectiveness of employment of commercial 3D audio systems. First we described a client/server paradigm useful for rapid prototyping of 3D audio-enabled applications using COTS hardware. By utilizing a standard network interface, the end application programmer is isolated from the vagaries of the 3D audio hardware and API and therefore can focus on the application, improving efficiency. In the second part, we described the results of a psychophysical experiment in which subjects were able to learn to adapt to the *shortcomings* of low-cost 3D audio hardware that likely result from the use of nonindividualized HRTFs. In summary, we suggest that with the use of the RSC 3D audio server (an engineering solution) and visual feedback training (a psychophysics-based solution), 3D audio-enabled applications can be efficiently and effectively deployed using low-cost COTS 3D audio hardware.

5. REFERENCES

[1] J. Blauert, *Spatial Hearing*, MIT Press, Cambridge, MA, 1983.

[2] F. L. Wightman and D. J. Kistler, "Headphone Simulation of Free-Field Listening. I: Stimulus Synthesis," *J. Acoust. Soc. Am.*, vol. 85, no.2, pp. 858-867, 1989.

[3] E. M. Wenzel, J. D. Miller, and J. S. Abel, "A software-based system for interactive spatial sound synthesis," in *Proceedings of the International Conference on Auditory Display 2000*, Atlanta, 2000.

[4] K. van den Doel and D. K. Pai, "JASS: A Java Audio Synthesis System For Programmers," in *Proceedings of the International Conference on Auditory Display 2001*, Espoo, 2001.

[5] N. Tsingos, "A Versatile Software Architecture For Virtual Audio Simulations," in *Proceedings of the International Conference on Auditory Display 2001*, Espoo, 2001.

[6] A. Sibbald, "Virtual Ear Technology," 1999. Available from http://www.sensaura.co.uk

[7] K. Wang, V. Sundareswaran, P. Bangayan, and C. Tam, "Applying Head-Related Transfer Function-Based 3D Audio," in M. Vassiliou and T. Huang (ed.), *Computer-Science Handbook for Displays: Summary of Findings from the Army Research Lab's Advanced Displays & Interactive Displays Federated Laboratory*, Government Printing Office, 2001, pp. 157-164.

[8] P. Zahorik, C. Tam, K. Wang, P. Bangayan, and V. Sundareswaran, "Localization Accuracy in 3-D Sound Displays: The Role of Visual-Feedback Training," in *Proc. Advanced Displays and Interactive Displays ARL Federated Laboratory 5th Annual Symposium*, College Park, MD, March 2001, pp. 17-22.

[9] A. W. Mills, "Auditory Localization," in J. V. Tobias (ed.), *Foundations of Modern Auditory Theory, Vol. II*, Academic Press, New York, 1972, pp. 303-348.