Techniques for Low Cost Spatial Audio

David A. Burgess

Graphics Visualization and Usability Center - Multimedia Group
Georgia Institute of Technology
Atlanta, Georgia 30332
burgess@cc.gatech.edu

Abstract

There are a variety of potential uses for interactive spatial sound in human-computer interfaces, but hardware costs have made most of these applications impractical. Recently, however, single-chip digital signal processors have made real-time spatial audio an affordable possibility for many workstations. This paper describes an efficient spatialization technique and the associated computational requirements. Issues specific to the use of spatial audio in user interfaces are addressed. The paper also describes the design of a network server for spatial audio that can support a number of users at modest cost.

Introduction

There are two basic ways of making two-channel audio recordings. The most common is stereo. A stereo recording captures differences in intensity and, possibly, differences in phase between points in a sound field. From these differences, the listener can gain a sense of the movement and position of a sound source. However, the perceived position of a sound source is usually along a line between the two playback speakers, and when monitored with headphones, sound sources appear along an axis through the middle of the head. This effect is due to the fact that the microphones used for stereo recording provide a poor model of the way sound really arrives at the ears. Human ears are not several feet apart, they do not have symmetric field patterns, and they are not separated by empty space.

The other method for two-channel audio recording is binaural[1]. Binaural recordings are intended to be reproduced through headphones, and can give the listener a very realistic sense of sound sources being located in the space outside of the head. Sounds can be in front of, behind, or even above or below the listener. This effect is achieved by using a better model of the human acoustic system, such as a dummy head with microphones embedded in the ears (Plenge, 1974). Because of the better model, the sound waves that arrive at the eardrums during playback are a close approximation of what would have actually arrived at a listener's eardrums during the original performance.

Along with greater realism, binaural sound provides a number of other advantages over plain stereo. It conveys spatial information about each sound source to the listener. Furthermore, when sounds are spatially separated, a listener can easily distinguish different sources, and focus on those sources which are of interest while ignoring others. This is the so-called "cocktail party effect" (Cherry, 1953).

If sounds can be recorded in this manner, an obvious next step is to convert monaural sounds to binaural sounds by artificially spatializing them. Given this ability, people who use sound in human-machine interfaces can gain the advantages that spatial sound offers. This goal has lead to research interest in the subject by the military, by NASA (Wenzel, et al, 1988), and by user interface designers (Ludwig, et al, 1990, 1991).

This research is part of Mercator, a project to develop a nonvisual interface to X Window System applications for visually impaired software developers (Mynatt & Edwards, 1992). Until the proliferation of graphical user interfaces, blind professionals could excel in many fields that relied on the frequent use of computers. The all-text output of a TTY mapped reasonably well into a world of voice synthesizers and Braille devices. Today, however, as mice, icons, and pop-up menus invade their workplaces, these professionals are finding it progressively more difficult to use the applications they need. The goal of Mercator is to map the behaviors of generic, unmodified X applications into an auditory space. An important feature of Mercator is the use of spatial sound as the primary organizational cue, and for Mercator to be useful, this feature must be supported at modest cost. While a few commercial spatial sound systems exist, most are prohibitively expensive. One low-cost system, Focal Point^{tm}, is presently available but does not provide an adequately open architecture for Mercator and is not available for most platforms which support the X Window Systems.

This paper is organized in seven sections:

[1] In strict terms, "binaural" and "stereo" mean exactly the same thing—two channels of sound. However, in the music recording field, these terms often carry the different meanings given here.

- How We Localize Sounds - a brief tutorial on auditory localization cues
- Basic Technique for Synthetic Spatialization - how spatialization filters are generated and used
- Computational Costs for Spatialization - estimating the computing power needed to apply a given set of filters in real time
- Problems with Spatial Sound - common shortcomings of spatial sound systems
- Getting Cheap - methods for reducing the computational cost of spatialization filters
- Network Servers for Spatial Audio - techniques for increasing the utilization of DSP resources
- Future Directions - low-cost modifications which can improve the quality of spatializing systems and the need for a spatial sound control protocol

How We Localize Sounds

In order to produce convincing spatial sound, we must know how auditory localization works, or at least what *cues* influence our sense of location for sound sources[2]. There are eight types of cues that are of particular importance in determining direction and distance. The four cues we will initially concern ourselves with are interaural delay time, or IDT (Rayleigh, 1907), head shadow (Mills, 1972), pinna response (Gardener, 1973), and shoulder echoes (Searle, et al, 1976). Together, these form the *head-related transfer function* (HRTF) (Searle, et al, 1976, Blauert, 1983).

To describe the HRTF, we must establish a coordinate system about the head. We will define the center of the head as the point halfway between the ears. We will define the Z axis as running through the center of the head from the right ear to the left. We will define the angle from the Z axis as *theta*. We will define *azimuth* as the component of theta in the horizontal plane and *elevation* as the component of theta in the vertical plane. (See Figure 1.)

IDT (also called interaural group delay or interaural time difference), the delay between a sound reaching the closer ear and the farther one, provides a primary cue for determining the lateral position of a sound. The delay is zero for a source directly ahead, behind, or above the listener and roughly 0.63ms for a source to one's far left or far right. This delay is also dependent on both the frequency of the sound and the distance of the source (Blauert, 1983). IDT manifests itself as a phase difference for signals below 1.6kHz and as an envelope delay for higher frequency sounds. A given IDT value constrains the position of a sound source to a hyperbola having an axis coincident with Z. This hyperbola approximates a cone of constant theta, sometimes called a *cone of confusion*.

Head shadow is the effect of a sound having to pass through or around the head to reach the far ear. Not only does head shadow affect overall intensity by about 9dB, but the head also acts as a linear filter which varies with both the direction and distance of a sound source.

At frequencies of 4kHz and greater, the effect of the outer ear, or pinna, is important for determining both the azimuth and elevation of a sound source. Like the head itself, the pinna acts as a filter and has a response which is dependent on the direction of a sound. Knowing pinna response, it is speculated that direction can be estimated by comparing the spectra of sounds arriving at the two ears. For familiar sounds (*a priori* spectral information), the brain can estimate direction on the basis of pinna response from a single ear.

Certain frequencies (roughly 1-3kHz) reflect from the shoulders and upper body (Gardener, 1973). The shoulder echoes reach the ear with a delay which is dependent on the elevation of the source. Additionally, the effects of the reflection on the spectrum of the sound are direction-dependent. For familiar sounds, shoulder echoes may provide both elevation and azimuth information, although the previously mentioned cues are likely to be of greater importance (Searle, et al, 1976).

The other four cues are head motion (Thurlow & Runge, 1967), vision (Thomas, 1940), early echo response (Moore, 1990), and reverberation (Gardner, 1969). The lack of these cues can make spatial sound difficult to use. We tend to move our heads to get a better sense of a sound's direction (Thurlow, et al, 1967). This “closed-loop” cue can be added

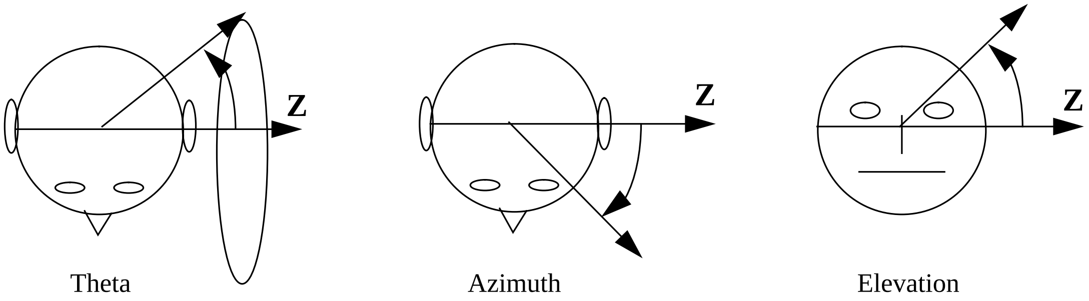

Figure 1. Coordinates About the Head

[2] For a comprehensive reference on this subject, see Blauert (1983).

to a spatial sound system through the use of a head-tracking device. Furthermore, we rely so heavily on where we see a sound source that we will ignore our auditory directional cues if they disagree with visual cues.

When a sound travels to our ears, it is almost always accompanied by reflections from surfaces in the listening environment. Early echo response is a term for the clear echoes we hear (but do not consciously perceive) in the first 50 to100ms after a sound starts. Early echo response and the dense reverberation that follows are believed to be important cues for distance and direction. Early echo response and reverberation will be addressed later in the paper.

Basic Technique for Synthetic Spatialization

The HRTF represents a linear system that is a function of sound source position. Therefore, sounds can be spatialized artificially using well-known digital filter algorithms, given the right parameters and enough computing power. Although the HRTF can vary substantially from individual to individual, people who localize well tend to have similar HRTF's (Wenzel, et al, 1988). It is believed that the important features of the HRTF are consistent enough that one such set of filters may be suitable for a large portion of the population (Wenzel, 1992).

The spatialization technique we will focus on is the use of an empirical HRTF measured from the ears of a specific person (Wightman and Kistler, 1989a, 1989b). Special probe microphones are used to monitor sound near the eardrum of a test subject, including all of the effects of the HRTF. This is similar to the technique for binaural recording—the primary difference is that a living human is used instead of a mannequin. Bursts of pseudorandom noise are played from a digital source through speakers at various positions about the subject. The noise bursts are recorded through the subject's ears with the special microphones. Having both the original and recorded sounds, the subject's HRTF can be computed at each speaker position.

Once measured in this manner, the HRTF can be expressed as a set of convolution operators, or finite impulse response (FIR) filters. For each position (azimuth, elevation), there is a pair of filters: one for each ear. To place a sound in a given direction, simply apply the corresponding filter pair to the sound stream. The filters are typically measured at 10 to 20 degree intervals, and filters for intermediate angles can be bilinearly interpolated from the empirical set (Wenzel, et al, 1988). The algorithm for the real-time spatializer is simple and it presented here as two concurrent processes:

```
Process 1: update position of the sound
    get desired azimuth and elevation
    look-up filter pairs for four nearest available positions in the filter tables
    interpolate to get the desired filter pair
    send the new filter pair to process 2
Process 2: apply the filter
    split monaural source into left and right channels
    apply the current left and right filters to the sound streams
    output sound streams for conversion to two-channel analog audio
    replace the current filter pair whenever a new filter pair is sent
```

For Mercator, these two processes are implemented in separate hardware. Process 2, which has a heavy computational demand, is executed by a dedicated DSP (a DSP56001). This process consists of a number of interrupt-driven subprocesses. Process 1, which has a relatively light computational demand, runs on the DSP's host machine (a SPARCstation).

Computational Costs for Real-Time Spatialization

Most of the time for a convolution is spent computing a dot product of two vectors: the filter operator and a segment of the signal being filtered. Figure 2 shows the C code for the central loop.

This loop (lines 1-6) must be executed for every sample in the output signal. For example, at a sample rate of 50kHz, this loop must be executed 100,000 times a second for real-time performance (50kHz for each ear). Let us call one iteration of the loop in lines 4 and 5 a *convolution point*. This operation requires two fetches, two increments, one multiply, one accumulate, a compare, and a jump. If the loop is unrolled, the compare and jump are eliminated. The convolution rate, in points per second, for a given real-time filter is equal to the sample rate x number of output channels x size of the filter. Notice that complexity grows linearly with filter size. At first glance, it seems that complexity is linear with sample rate as well, but in practice complexity will grow with the *square* of the sample rate—as the sample rate increases, the filter size must also increase to cover the same

```
/* SIZE is the number of filter
coefficients. it is assumed to be
even. n is the current index into
the input signal array x. y is the
output array. filter is the array
of filter coefficients */

ins = &x[n-SIZE/2];                  1
flt = &filter[0];                    2
sum = 0;                             3
while (flt<(filter+SIZE))            4
   sum += (*(ins++))*(*(flt++));     5
y[n] = sum;                          6
```

Figure 2. Code for Convolution

Table 1: Convolution Performance for Various Machines

Machine	Convolution Points per Second	Code
Convolvotron	>200M	Hardware
DSP56001, 27MHz	>11.3M	Assembler
SPARCstation2	~0.88M	cc -O4
NeXT Cube, MC68030	~.0.48M	gcc -O

time span.

For research purposes, these filters are typically produced with a sample rate of 50kHz and a duration of 512 samples (10.2ms). To fill the awesome computational demand of applying these filters in real time (51.2M points per second), Crystal River Engineering and NASA Ames Research Center have designed a special signal processing engine called the Convolvotron, which computes 128 convolution points in parallel on every 400ns cycle. Unfortunately, the current cost of the Convolvotron prohibits its widespread use. Single-chip digital signal processors (DSP's) are designed to perform one convolution point on every instruction cycle. A typical RISC machine needs at least 6 instruction cycles for a convolution point and might need dozens if multiplication hardware is not available. Performance figures for a variety of machines are given in Table 1. Figures for the Convolvotron and DSP56001 are based on complete audio filtering implementations. Figures for workstations do not include the effects of I/O or finite buffer space; they do, however, include the effects of operating systems and user interfaces.

Problems with Spatial Sound

A classic problem with spatial sound is an inability of listeners to tell whether sound sources are in front of or behind them. This problem is not nessecarily due to some failing of the spatial sound system, because front-back confusion can occur with real sound sources as well.

Another common shortcoming is lack of externalization. As a result, sound may appear to emanate from points inside the head. Externalization is lost when signals reaching the ears are not adequately consistent with those that would be produced by external sources (Plenge, 1974). In practical terms, this means that externalization requires a faithful model of the HRTF.

Localization is the sense that a sound is coming from a particular direction, instead of just vaguely from one side or the other. Like externalization, localization requires a good model of the HRTF.

A minimum requirement for any useful spatial sound system is a monotonically increasing relationship between perceived position and target position.

Position update rates should be high enough to give an illusion of continuous movement. In tests for Mercator, an update rate of 10Hz has been found to be adequate for rotational speeds of up to 180 degrees/second.

Getting Cheap

A set of spatialization filters were provided to Mercator by Professor Fredric Wightman of the University of Wisconsin at Madison. The test subject who provided these filters is referred to in several references as SDO (Wenzel, et al, 1988, Wightman & Kistler, 1989a&b, Wenzel, 1992). The filters were produced from recordings made in an anechoic chamber at a sample rate of 50kHz and each had a duration of 512 samples (10.24ms).

Because computational cost grows quadratically with sample rate, we gain a great deal by using the lowest rate possible. This particular set of filters contained little or no useful information above 16kHz (Wightman; personal correspondence). Furthermore, most adults hear very poorly at such high frequencies. It follows that these filters can be resampled at rates as low as 32kHz with little or no loss of effectiveness. At 32kHz, the length of the filter is reduced to 328 points, and the real-time computational requirement is reduced to 21M points per second—less than half the original cost.

Much of the measured HRTF between 12kHz and 16kHz is believed to be inaccurate or excessively noisey due to the limitations of existing recording equipment (Wightman; personal correspondence). Only those measurements below 12kHz are know to be accurate. A bandwidth of 12kHz corresponds to a sample rate of 24kHz. When resampled at 24kHz, the filters shrink to 247 points. For real-time use, a 24kHz filter set requires 11.9M convolution points per second - less than one quarter the demand of the original filters. However, it should be pointed out that at rates below 32kHz the filters are not equivalent to the original 50kHz filters, and are not guaranteed to perform as well.

Another loss of the efficiency of the original filters was due to long periods (at least 3.40ms) of silence preceding each impulse response. If silence is removed from the beginning or end of a filter, its spectrum is not changed, because the points removed are effectively zero. If the same duration of silence is removed from the same end of each filter, the relative delays of the filters are also preserved. Furthermore, these silent periods gave an indication of the noise floor of the original filter set. Once this noise floor was known, it was determined that each filter ends in at least 0.54ms of silence, as well. With all 3.94ms of silence removed, the 50kHz filters shrink to 315 points. The 32kHz filters shrink to 202 points, and require only 13M points per second for real-time application. The 24kHz filters shrink to 139 points and require 6.7M points per second, but may not be as effective as the original set.

Further truncation would remove non-zero portions of some of the filters. It is possible to multiply by some function to zero-out samples at the beginning and/or end of a FIR. These samples can then be truncated from the rest of the filter. This process is called *windowing*, and the function used for the multiplication is called a *window function*. As a filter is windowed to smaller and smaller sets of points, it looses spectral resolution—there are fewer points in its Fourier transform. If the windowing is done properly, however, the general shape of the spectrum (and thus the general effect of the filter) will remain the same as its details disappear.

It would be useful to know just how small HRTF filters can be before they stop working or, equivalently, how much detail the HRTF must have in the frequency domain to produce convincing spatial sound. More aggressive truncation and windowing can be used to further reduce the lengths of filters. Filters as short as 128 points (2.56ms) are used effectively with the Convolvotron at 50kHz. An equivalent filter at 32kHz would have only 82 points and require 5.3M points per second, which would allow a DSP56001 to spatialize two channels in real time. In an informal test, it was found that for most listeners, monotonicity of perceived position with respect to target position can be maintained with FIR's as short as 1.45ms (Burgess, 1992). Further study is needed to know the localization accuracy of such filters, but if found to perform adequately, they would allow as many as four 32kHz signals or eight 24kHz signals to be spatialized in real time.

Network Servers for Spatial Audio

In Mercator, most of the sounds used will be auditory icons (Buxton, et al, 1991, Gaver, 1986, 1989). They will be short in duration and only presented sporadically. Other sounds, such as background noises, will be periodic and completely characterized by a single period. In the current Mercator design, these sounds will be sampled or precomputed for speed. To conserve memory and increase flexibility, effects such as those described in Ludwig, et al (1990) may be implemented in real-time by the DSP. It is likely that in a distributed environment, the library of base sounds will kept on a central server, called up by client workstations as they are needed, and cached locally while they are used.

If a particular sound is put at a particular position in space, we call it an acoustic event (Blauert, 1983). When spatial sound is used, the client workstation should not simply cache sounds—it should cache acoustic events. If an acoustic event cache is used, sounds only need to be spatialized when they are moved to new positions. When used in such a manner, the spatialization engine is no longer part of the real-time audio stream, but is a service which is utilized on an as-needed basis.

Mercator is largely intended to be a quiet interface. While low-level background sounds may play continuously for navigational purposes, most potential users have said that a continuous stream of acoustic events would be annoying. With an acoustic event cache handling background sounds, it is speculated that a single-user DSP will usually be idle. By combining the spatialization engine with the audio server to create a single acoustic event server, expensive DSP resources can be better utilized. Another advantage of a server architecture is that DSP hardware need not be compatible with the various user machines, reducing the need for expensive, machine-specific ports of low-level driver code and the headache of locating affordable DSP hardware for older or poorly supported platforms.

Here is an example of the interaction between a user interface and an acoustic event server:

1) In response to a user action, Mercator generates a request for a particular sound at a particular location—an acoustic event.

2) If this event is cached locally, it is played immediately. Otherwise, the request is forwarded to the server.

3) The server retrieves or generates the desired sound and then sends it through the spatialization engine (DSP or other hardware) along with the proper filters.

4) The event is then sent back to the user machine, where it is cached and presented over headphones.

A diagram of the system is shown in Figure 3.

Future Directions

There are a number of possible methods for improving both the quality and cost of spatial audio. User training is a important factor. The user is, in effect, listening through another person's ears. The signals that reach listener‘s eardrums are intended to be the same as those which would reach to ears of the subject from whom the filters were produced. New users may need time to adapt.

Front-back confusion is a classic problem for spatialization systems. Our primary cues for distinguishing front from back are pinna response and head movement. By coupling the spatialization system to a head tracker, front-back reversals can be eliminated. Another technique which can improve front-back differentiation as well as overall spatialization quality is the addition of early echoes from the walls of a simulated listening room. Generating realistic first order echoes for a small room would require filters of several thousand points. However, experiments for Mercator have shown that an echo from a single wall, computed at modest cost, allows reliable front-back differentiation for even our most difficult test subjects. The Mercator project is currently pursuing less expensive methods for simulating room acoustics.

For a familiar sound, a primary cue for distance perception is intensity (Gardner, 1968, Laws, 1973). At low frequencies (below 1kHz) intensity of a sound varies inversely with the square of the distance from its source. At higher frequencies, dispersion causes an inverse cubic variation.

Additionally, reverberation provides a important distance cue. As a sound source moves away from a listener, the ratio of direct energy to reverberant energy decreases. By controlling this ratio, we can impart a sense of distance. When used properly, this reverberation can also improve externalization (Plenge, 1974). Furthermore, reverberation can provide a navigational aid—different regions of a workspace can have different reverberant characteristics, just like different rooms in a building.

Another important requirement for spatial audio to be useful in an interface is a control protocol between the interface software and the spatialization system. It is already known that this protocol must meet several requirements:

- It should provide means for the interface software to present a script specifying the choreography of multiple sound sources.
- It should provide immediate update capabilities so that sounds (or scripts) may be initiated, interrupted, or changed in real time.
- Available methods for host-to-DSP communication vary widely. The protocol should assume a simple communication model (a lowest common denominator) to be portable to a variety of systems.
- It should allow for the insertion of acoustic event caches.
- It should allow for the integration of head-tracking devices.
- It should allow for the integration of suitable data compression schemes to reduce I/O bandwidth.

An important part of the development of spatial sound for Mercator will be the refinement of these requirements and the design and implementation of an effective spatial sound control protocol.

Acknowledgments

Most of the spatial sound work for Mercator has been sponsored by the NASA Marshall Space Flight Center under Research Grant NAG8-194. Additional work has been supported by Sun Microsystems. The author would also like to thank Professor Fredric Wightman of the University of Wisconsin at Madison for providing HRTF filters and related information, as well as Scott Foster of Crystal River Engineering for information about the Convolvotron.

References

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

2. Burgess, D.A (1992) Real-time audio spatialization with inexpensive hardware, GVU Tech. Report GIT-GVU-92-20.

3. Buxton, W., Gaver, W. & Bly, S. (1991) The Use of Non-Speech Audio at the Interface, Tutorial No. 8, CHI’91, ACM Conference on Human Factors in Computer Systems, ACM Press: New York.

4. Cherry, E.C. (1953) Some experiments on the recognition of speech with one or two ears, J. Acoust. Soc. Am., 22, 61-62.

5. Gardner, M.B. (1968) Distance estimation of 0° or apparent 0°-oriented speech signals in anechoic space,

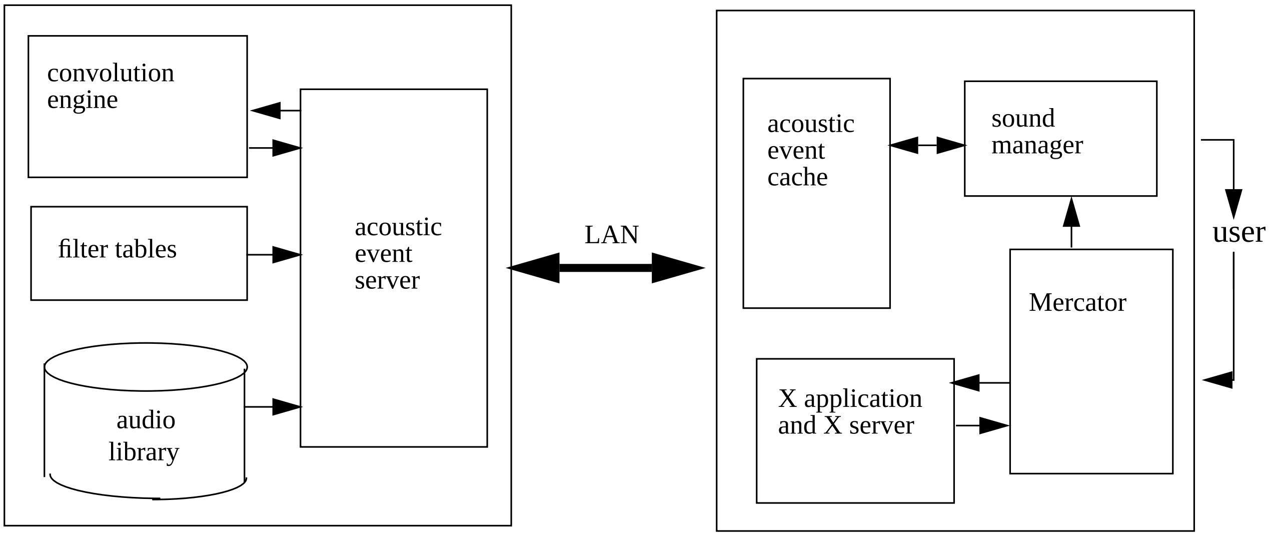

Figure 3. Acoustic Event Server

J. Acoust. Soc. Am., 45, 47-53.

6. Gardner, M.B. (1973) Some monaural and binaural facets of median plane localization, J. Acoust. Soc. Am., 54, 1489-1495.

7. Gaver, W.W. (1986) Auditory icons: Using sound in computer interfaces, Human-Computer Interaction, 2, 167-177

8. Gaver, W.W. (1989) The sonicfinder: An interface that uses auditory icons, Human-Computer Interaction, 4, 67-94

9. Laws, P. (1973) Entfernungshören und das Problem der Im-Kopf-Lokalisierheit von Hörereignissen [Auditory distance perception and the problem of in-head localization of sound images], Acustica, 29, 243-259

10. Ludwig, L.F., Pincever, N. & Cohen, M. (1990) Extending the notion of a window system to audio, Computer, Aug. 1990, 66-72.

11. Ludwig, L.F. & Cohen, M. (1991) Multidimensional audio window management, International J. Man-Machine Studies, 34(3), 319-336

12. Mills, A.W. (1972) Auditory localization, *Foundations of Modern Auditory Theory*, Vol. II/8, Academic: New York, NY.

13. Moore, F.R. (1990) *Elements of Computer Music*, Prentice Hall: Englewood Cliffs, NJ.

14. Mynatt, E. & Edwards, W.K. (1992) The Mercator environment: A nonvisual interface to X Windows and Unix workstations, ACM Symposium on User Interface Software and Technology, UIST '92.

15. Plenge, G. (1974) On the differences between localization and lateralization, J. Acoust. Soc. Am., 56, 944-951.

16. Lord Rayleigh [Strutt, J.W.] (1907) On our perception of sound direction, Phil. Mag., 13, 214-232.

17. Searle, C.L., Braida, L.D., Davis, M.F. & Colburn, H.S. (1976) Model for auditory localization, J. Acoust. Soc. Am., 60, 1164-1175.

18. Thomas, G.J (1940) Experimental study of the influence of vision on sound localization, J. Exper. Psych., 28, 163-177

19. Thurlow, W.R. & Runge, P.S. (1967) Effect of induced head movements on localization of direction of sounds, J. Acoust. Soc. Am., 42, 480-488.

20. Thurlow, R.W., Mangels, J.W. & Runge P.S. (1967) Head movements during sound localization, J. Acoust. Soc. Am., 42, 489-493.

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

22. Wenzel, E.M., Wightman, F.L. & Foster S.H. (1988) A virtual display system for conveying three-dimensional acoustic information, Proceedings of the Human Factors Society - 32nd Annual Meeting.

23. Wightman, F.L. & Kistler, D.J. (1989a) Headphone simulation of free-field listening I: stimulus synthesis, J. Acoust. Soc. Am., 85, 858-867.

24. Wightman, F.L. & Kistler, D.J. (1989b) Headphone simulation of free-field listening II: psychophysical validation, J. Acoust. Soc. Am., 85, 868-878.