Extending the Notion of a Window System to Audio

Lester F. Ludwig, Bell Communications Research*
Natalio Pincever, Massachusetts Institute of Technology Media Lab
Michael Cohen, Northwestern University

Just as visual window systems let multiple applications share display resources, an audio window system could bring order to the cacophony of multiple simultaneous audio sources.

Visual window systems have become successful elements of human-machine interfaces, allowing multiple applications to simultaneously use a common visual display resource. This capability is really the combination of two abilities:

(1) The user can control the spatial organization of the multiple visual objects (windows) via a window manager utility.
(2) The user can shift visual attention as needed among the various displayed objects.

With audio's increasing importance in computer applications (multimedia or otherwise),[1,2] users will soon need similar presentation, management, and organizational capabilities to avoid a confusing cacophony of multiple audio sources sounding at once.

In our everyday lives, avoiding such confusion presents no real problem as our hearing system sorts out sound sources of differing timbre[3] and differing locations in space.[4,5] In office, home, and social settings we can shift attention among various audio sources. We perceive each source as coming from a specific location or direction and as exhibiting other qualities, such as being muffled or echoed.

In today's PCs and workstations, sound sources are either switched on-off or mixed with simple audio mixers. These techniques, however, do little to merge sound sources in the way our ears expect, that is, with differing binaural delay and frequency-response signatures.[4,5] Can we extend PC and workstation audio to use human ability to deal with multiple sound sources?

The work summarized in this article examines how to take advantage of the ear's valuable sorting and labeling potential. By using audio signal processing to exploit the psychoacoustic properties of hearing, we can simultaneously present multiple audio sources in a way that lets a user shift attention among them. We can organize audio sources spatially in one, two, or three dimensions (that is, the sources are perceived to be at different locations in a line, plane, or three space), and we can introduce hierarchical levels of emphasis. By dynamically reconfiguring the signal processing pipelines via simple human interfaces, we can interactively alter the organization of multiple audio sources. From the user's view, the resulting functionality has a strong similarity to visual window systems; hence, we refer to it as an audio window system. The similarity to visual systems lets audio window systems complement or work in synergy with visual window systems.

Envisioned use. A user constructing a multimedia document or presentation might need to manage several audio sources. Audio from various multimedia databases, editors, and message systems would have to be presented in an environment that includes other application output and prompts, telephone calls, voice mail, and collaborative teleconferences that might need to access the editors and data-

* Lester Ludwig has since become principal scientist at Vicor, Inc.

0018-9162/90/0800-0066$01.00 © 1990 IEEE

bases. A graphical user interface would feature a map of the sound sources' virtual locations. The user could reposition a source by dragging its visual icon with a mouse, and the perceived source location would dynamically track the icon position. For sound-annotated documents, audio would seem to come from the same position as the corresponding visual window.

Spatial, hierarchical, and exclusion management capabilities could also be used within an individual application just as child windows can be used within a parent window in visual window systems. Therefore, audio windowing can be used both within a single application and as part of the audio management of all applications active at the user terminal.

For example, audio windowing can be used to implement spatial data management systems[6] in which users browse through a data "world" of relocatable objects. Simulations and scientific visualizations could use audio windows to hierarchically and spatially organize sonic cues corresponding to activity on a factory floor or propagation of a crack through a solid. Messaging applications could use audio windowing to manage voice annotations to a multimedia message (or even annotations to forwarded voice mail) so that the annotations are spatially separated from the rest of the message's audio. In entertainment applications, audio windowing can be used for special effects or to support theatrical sound systems such as Surround Sound 360 or THX.

Although much technology has been applied to teleconferencing and computer-supported cooperative work,[7] little has been done to improve how audio is handled and presented in a teleconference. Audio windowing can help introduce new, natural metaphors into electronic meetings, and hierarchies can help manage the audio for the main conference and side discussions. Spatial metaphors improve speaker recognition and reduce cacophonous effects that occur when multiple speakers talk simultaneously. Teleconferences might also be presented as informal interaction systems. As an extension of increasingly popular "chat lines," we can imagine virtual gatherings of geographically separated users who "circulate" around an electronic "room," listening in or temporarily joining conversations and moving on as would conventioners or minglers at a cocktail party. Caucuses wishing to meet privately could electronically adjourn to an adjacent private audio chamber.

Relationship with graphics window servers and the Vox audio server. Our notion of audio windowing focuses on the user's audio interface and is therefore analogous to the interface of a visual window system. A visual window system interface uses a graphics window server and a window manager. Similarly, an audio window system must include audio processing for implementing the audio presentation and a means of controling this processing. Graphical window servers also give application programmers an interface to such functions as resource management, event notification, and other operations not directly involved in display. The Vox Audio Server[2] focuses on resource management of audio peripherals such as mixers, tape recorders, and speech interfaces. However, it does not directly give the user an audio presentation interface. Vox could be used to manage and control audio processing allocations and configurations for an audio window system. The scope of an audio window system, therefore, partially overlaps the scope of Vox (see Figure 1).

Our goals. Our work at Bellcore's Integrated Media Architecture Laboratory[8-10] project attempted to find solutions to the audio management problem that arises when several communications services use audio simultaneously. We sought to

- investigate methods, costs, and effectiveness of managing several simultaneous audio sources;
- examine how to distribute audio windowing functions among user terminals, local networks, and public networks;
- consider terminal and network implementations of audio window systems that can work together; and
- explore how audio windowing can be used in stand-alone services (such as teleconferencing, spatial data management,[6] training, CAD, entertainment), some of which the network may provide directly and others the network must indirectly support.

We have not fully met any of these goals, but we can identify some promising techniques, observations, and findings from our work.

Techniques

We focus here on the signal processing methods used to create hierarchical and spatial distribution among "nearly arbitrary" (not pure sine wave*) audio sources. For more details, see Ludwig and Pincever.[10]

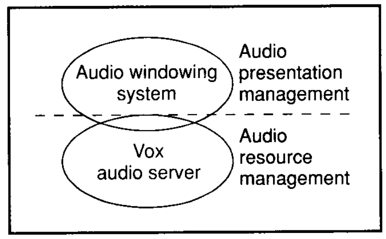


Figure 1. The relationship of audio windowing and the Vox Audio Server.

Hierarchies. The music, radio, and film industries have used special effects for years to make certain sound sources "stand out" or "stand back" from others. In particular, the music industry commonly uses electronic processing to emphasize solo instruments or vocalists. We believed we could use such processing to give different degrees of emphasis to individual audio signals within a collection. These differentiated signals would be naturally perceived as being in a hierarchical relationship. There are several techniques and commercial products that accomplish this by incorporating one or more of the following functions:

- Self-Animation (massive frequency-dependent phase distortion)
- Distortion (nonlinear wave-shaping, that is, amplitude distortion)
- Thickening (a chorused "doubling" effect via pitch-shifted signals)
- Peaking (linear band-emphasis filtering)
- Distancing (reverberation and echo)
- Muffling (linear low-pass filtering)
- Thinning (linear high-pass filtering)

A number of commercial products called exciters or imagers perform self-animation, both with and without distortion. Thickening (also called doubling) is available in devices called pitch shifters or harmonizers. Peaking can be realized by graphic, parametric, or shelving equalizers. Distancing is performed by echo, digi-

* In most of the techniques we have studied, pure sine waves act as singularities (that is, the effects break down) because artifacts in complex frequency spectra are key to producing effective cues for hearing. Most natural and machine-generated sounds are not pure sine waves, so we do not consider this a concern.

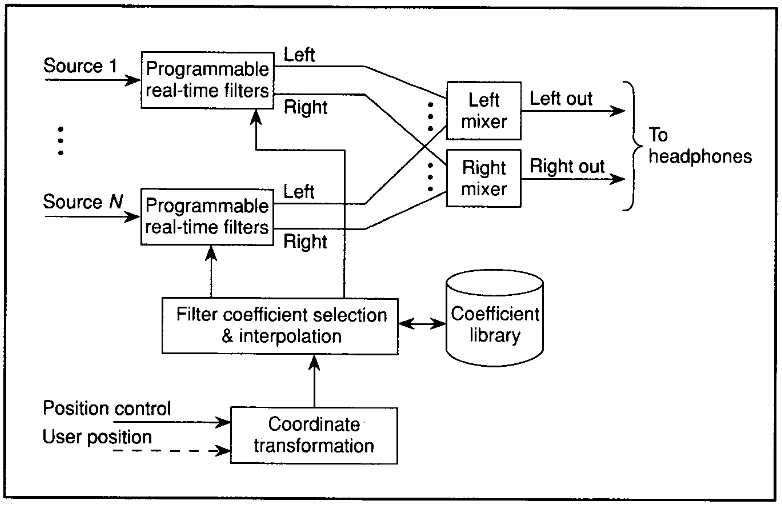


Figure 2. A spatial sound system based on the Crystal River Convolvatron.

tal delay, and reverb devices. Muffling and thinning can be realized by dedicated low-pass and high-pass filters but are also available on graphic, parametric, and shelving equalizers. Distortion-only devices (such as guitar "fuzz boxes") are also available. We restricted our signal processing to exciters, pitch shifters, and equalizers based on our experience with the various options and the effects we desired. However, we provide some background on the each of the functions below.

Self-animation makes a source sound more lively by accentuating frequency variations in the source signal, much as stones in a shallow creek accentuate water turbulence to produce more eye-catching patterns. Distortion produces a strained — perhaps "excited" — sound, but it reduces intelligibility and increases listener fatigue. Thickening produces a "thicker" sound while slightly decreasing intelligibility, although many low-cost pitch shifters also introduce additional undesirable artifacts.* Peaking works well with speech when used to boost amplitude in the 1-kilohertz range (where a good deal of speech phoneme information is carried), but is otherwise limited. Distancing gives a fuller, mystical sound but reduces intelligibility. Muffling creates impressions of confinement or distance but with greatly reduced intelligibility.

Informal experiments and our previous experience with these techniques suggest we can use self-animation, thickening, and peaking to highlight or emphasize an audio source, making it more prominent than an unprocessed source. Similarly, we believe we can use muffling and distancing to deemphasize or "background" a source, making it less prominent than an unprocessed source. We believe muffling can help to acoustically denote a metaphorical "grabbing" of an audio source (similar to "grabbing" a visual icon with a mouse in a visual system).

We chose the following steps for creating hierarchies. The highest level of enhancement uses a thickening operation piped into a self-animation operation. The next level uses only self-animation. The lowest level presents the native, unprocessed signal. These choices work fairly well given the quality of the audio products we used, although the difference in emphasis between the highest two levels was far less than between the lowest two. Thickening alone compared with self-animation alone does not create a well-defined hierarchical relation. However, the fact that the slightly trained ear can recognize thickening and self-animation as separate operations suggests possibilities for orderings with two or more indices. (Note that a hierarchy has only one index, which is the level occupied within the hierarchy.) We did not explore this, however, with any serious effort. We reserved muffling and thinning for use with the window manager to provide in-band audio feedback when an input device (such as a mouse) singles out or grabs a source.

One-, two-, and three-dimensional source distributions. Human hearing has remarkable capabilities for extracting subtle information from audio sources at the same time it extracts directional information.[3-5] For example, the same amplitude and pitch pattern can be filtered by a speaker's mouth and throat to say either "You went home?" or "He flew where?" A listener not only can recognize these questions effortlessly but also can tell roughly where they are coming from. The directional perception of human hearing appears to have much to do with binaural differences in delay and frequency filtering as produced by environmental and outer-ear acoustics.** Further, human hearing can use its ability to discriminate source direction to sort simultaneous audio sources and focus on a specific source, as at a cocktail party.

In our early work[10] we used audio mixers with stereo outputs in an attempt to create a one-dimensional spatial distribution for sounds presented via stereo speakers or headphones. Stereo-output mixers use differences in amplitude only and ignore the problem of correctly synthesizing natural delay and filtering attributes. Amplitude differences work well for correlated multiple sources (such as music) transmitted through speakers or headphones and for a small number of uncorrelated sources when used with spatially separated speakers. However, the amplitude difference effect creates cacophony under more general conditions. We experimented with recording studio techniques to make simultaneous sources sound more

* An arbitrary audio signal segment can be pitch shifted via Doppler effects created when a signal buffer is read out at a different rate than it was loaded. Underflow and overflows naturally result, so pitch shifters employ time-staggered multiple buffers whose outputs are sequentially directed to the output port in some form of fixed-duration cycle. Early low-cost pitch shifters periodically switched between the buffers, resulting in glitches. Modern low-cost pitch shifters periodically pan between the buffers, resulting in flanging effects because of the time staggering. The best approach would intelligently splice waveforms from a finishing buffer to those of a starting buffer. In any case, the pitch-shifted signal can then be mixed with the original signal. If the pitch shift is small, say 30 cents, the result is a thick chorused sound. Intelligibility can be improved by delaying the original signal by an amount roughly equal to the mean buffer delay incurred by the pitch-shifted signal.

** Binaural differences are those between what the left and right eardrums are presented from the end of the ear canal. Recognition of different filtering patterns created by outer-ear acoustics implies that the hearing system works from a set of sound templates from which it "deconvolves" information about the outer-ear filtering.

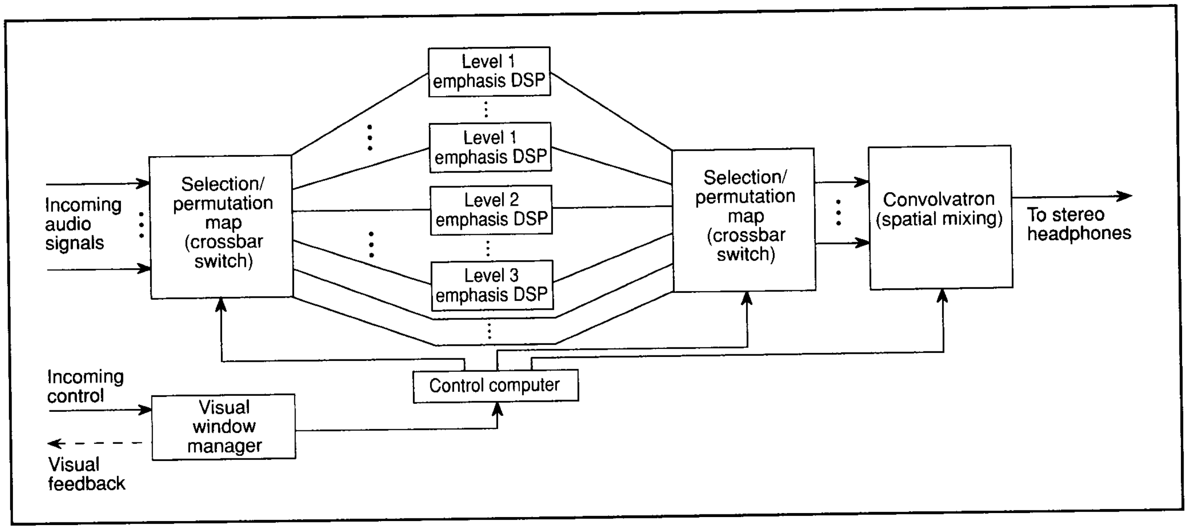


Figure 3. A prototype audio window system.

separated by changing equalization to impose a distinct spectral filtering signature on each source. Our experiments suggested that what was actually being exploited here was the hearing system's ability to sort and focus based on its ability to distinguish source location. Although unorganized filtering and left-right amplitude differences can be combined to create crude but usable artificial separations,[10] we decided to explore properly synthesized spatial sound. One reason for this decision was the strong metaphors made possible by synthesizing the effect of sound sources distributed in three space.

Three-dimensional spatial sound synthesis has been explored for many years, and a variety of techniques and systems are available. Spatial auditory cues are produced in nature by reflections and absorptions about the head, torso, and outer ear, and they have been extensively catalogued as a function of three-space location with respect to the head. Spatial sound synthesis uses linear filtering techniques to impose (stereo) pairs of transfer functions (one for the headphone's left speaker, one for the right) whose amplitude and delay response reproduce those of known spatial cues. From a signal processing standpoint, each transfer function can be imposed by continuously "shaping" the source audio signal with the impulse response corresponding to the desired transfer function. Algorithmically, this shaping is done via an operation called convolution. For a discrete time signal (such as sampled digital audio), a convolution first computes at each instant scaled versions of current and previous samples of the audio signal. Each sample is scaled by multiplying it by a specific coefficient from the collection of coefficients that represent the discrete time impulse response. The scaled samples are then added to produce the output sample for that particular instant. In general, impulse responses can be infinite (involving an infinite number of multiply and add functions realized by recursion) or finite (a finite collection of multiply and add functions that can be pipelined in parallel).

The spatial sound system we are using is based on the Crystal River Convolvatron, a high-throughput (300 million instructions per second) audio digital-signal-processor convolution engine fitted with four analog audio input channels and a stereo audio output (see Figure 2). This system time-interleaves eight separate 128-stage finite impulse response filters in 24-bit arithmetic on a single parallel pipeline created around a 128-stage delay-line/multiplier engine. An additional processor selects, retrieves, and interpolates coefficient arrays used by the pipeline to create the desired transfer functions. The outputs of all left- and right-ear filters are presented through stereo headphones (which this approach requires). The system can also perform real-time coordinate transformations, allowing not only the source locations but also the observer location to be adjusted in real-time.

A McDonnell Douglas Polhemus Isotrak gives the headphones' three-space coordinates (x, y, z; roll; pitch; and yaw) to the Convolvatron's coordinate-transformation calculation processor. The resulting dynamically selected filters use stereo spatial cues to create the psychoacoustic imagery of four monaural sound sources distributed in three dimensions. The sources' perceived position with respect to the user incorporates the user's head position and orientation at each moment (the position can be recalculated in 33 milliseconds). Further, the perceived position of each sound source can be stationary or moving (using periodic coordinate updates). This system is the work of Foster, with assistance from Wenzel and Wightman.[11] Although the head-tracking corrections might seem frivolous at first, they greatly help system effectiveness.

The Convolvatron can distribute sound sources in one and two dimensions as special cases. Even in the 1D case, the Convolvatron lets the user more clearly distinguish and selectively focus on two or more uncorrelated sources. Further, the 1D array of sources can appear in front of the user, to the side, etc., as opposed to the "inside the head" effect created by traditional stereo amplitude pans. The 2D case has obvious utility for PCs and workstations, as sounds can be spatially associated with specific application windows on the screen, while 3D arrays can extend the workspace to include regions beyond the screen area.

When headphones can be used, the use of 3D audio imaging to create realistic 2D and 3D arrangements of sound sources is far superior to conventional stereo output

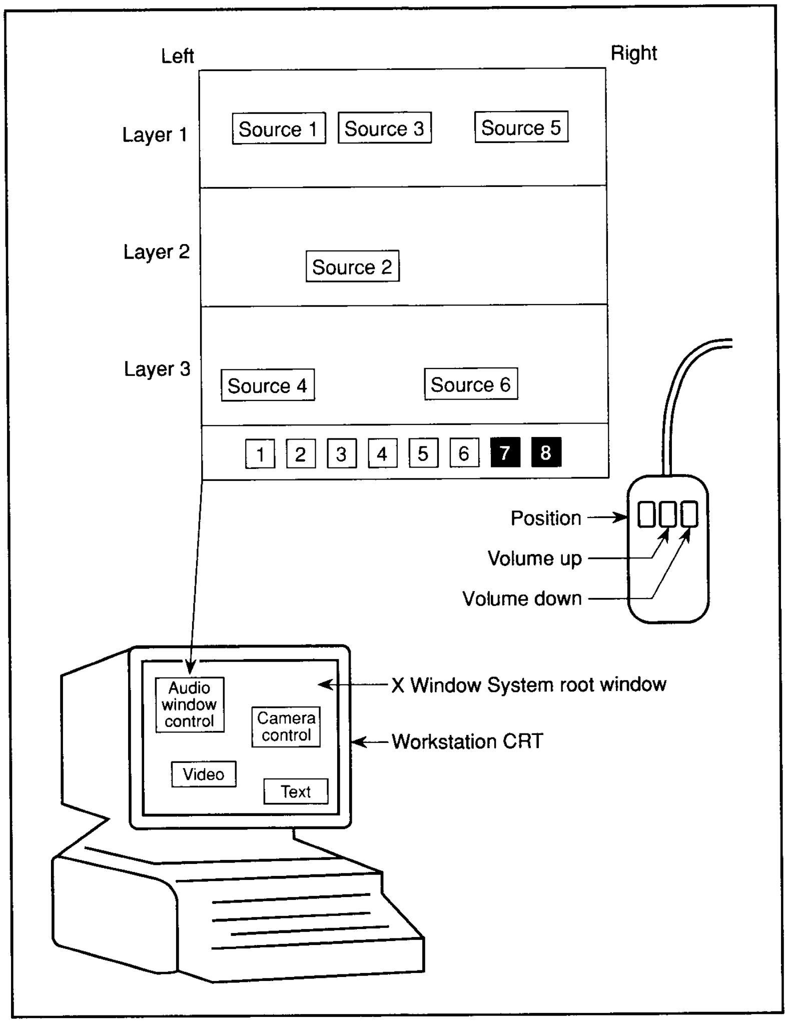


Figure 4. A graphical user interface, based on the X Window System, for the audio window system.

mixing. When stereo output mixing is used, intuitive graphical interfaces can control the organization of the sound sources. If movable audio sources are physically associated with the screen locations of visual windows, audio window management can track the effects induced by a visual window manager (we have not tried to implement this).

Prototype audio window system

Figure 3 shows our prototype audio window system combining hierarchical and spatial processing functions with a computer controlled switch, software, and human input devices. This system uses the Convolvatron as a multi-input, stereo-output audio "mixer." We can combine two unmodified Convolvatrons to channel eight inputs into a single stereo output. Source position and operator pipeline control are under computer control in the form of the Convolvatron and a crossbar switch, respectively.

Incoming audio signals are presented to the switch, where they are either ignored, routed without processing to the spatial mixing stage (the Convolvatron), or routed to hierarchical audio processing modules. A first crossbar-switch stage performs concentration and permutation functions for the hierarchical modules. The resulting outputs are then presented to a second crossbar-switch stage, where the outputs and unprocessed signals are either ignored or routed to specific inputs of the spatial mixer. The second stage performs concentration and permutation functions for the spatial mixer. The system also includes a computer-controlled stereo mixer to provide parallel output for comparisons.

We are studying two ways to control this hardware arrangement using metaphor-based "window managers." The first approach uses a graphical user interface (X Window System) controlled by a mouse (see Figure 4). This interface currently focuses on 1D spatial control and emphasizes the control of hierarchy. The four stacked boxes contain icons representing audio sources, which the user manipulates with the mouse. The bottom box always shows icons for all the audio channels, which the user can turn on or off with a simple mouse click. When a channel is on, another icon for it is displayed in one of the three upper boxes. When a channel is turned on after having been off, its icon is displayed in the same box where it was last displayed. The three upper boxes each denote a distinct type of hierarchical processing effect. When an icon is in one of the upper boxes, the corresponding channel is processed by the hierarchical effect corresponding to that box. The left-to-right position of the icon corresponds to the audio channel's 1D spatial position through the headphones. The mouse adjusts the icon's location, and hence its audio position and level of hierarchical effect, as well as the volume.

Our second approach focuses on 2D and 3D audio windowing using a VPL Research Dataglove instead of a mouse and employing audio cues for user feedback. The Dataglove is a flexible fabric glove fitted with optical strain gauges for measuring hand postures and an Isotrak for measuring hand position in three space. Our Dataglove approach uses both pointing and hand-gesture recognition.[12] Pointing in the perceived direction of an audio source singles it out to be turned on or off, moved, or changed in the hierarchy. Grasping and releasing motions move an audio source from one perceived location to another. As feedback, audio signal processing "thins" a source when the user points at it, and muffles a source when the user "grabs" it. Holding out one, two, or three fingers changes the source's hierarchy level.

Envisioned implementations

Terminal-based audio window systems let the user manage audio sources locally. This ability is particularly appropriate if many of the applications are run entirely on the terminal. Publicly shared network-based audio window servers, on the other hand, also have some advantages: They can be structured to give users "virtual" audio window hardware whose complexity varies as needed, they work together with telecommunications services that produce or carry audio, and they share the cost of a high-quality system among a potentially large community. There could be a significant need for both architectures and many cases in which they must work together.

Our preliminary work suggests that an effective audio window system needs much less complexity and fewer levels of digital-signal-processing precision than our current prototype. With appropriate metaphors, low-cost techniques can handle human input to audio window managers. Our work plan focused on techniques first, applications next, and simplifications last.

Terminal-based system. In this approach, all needed digital signal processing would be done in the terminal, PC, or workstation. Such a system would use an architecture like that in Figure 3 and could use Vox.[2] A terminal-based approach gives users access to audio windowing without connecting to a network.

It is unclear if digital signal processing and switching for audio windowing could be folded into future desktop computing products, but we expect it to contribute at least some additional cost. This cost, plus the other functional needs of network-based systems, motivate the network-based server design.

Network-based server. Figure 5 shows an architecture for a network-based audio window server. Digital signal processing and configuration switching resources are allocated as needed from a shared pool. This arrangement allows systems to be created on demand with widely variable numbers of channels. It also takes advantage of per-resource utilization gains inherent in shared-resource systems. Another advantage is the sharing of common signals in group applications. For example, in teleconferencing, users might have all processing parameters in common except their individual head positions, in which case the middle switch in Figure 5 simply fans out the same signals from the previous stages to multiple instances of spatial processing. In integrated-service networks, the network can provide audio window functions to manage the multiple audio streams that would be sent to the user. This reduces the number of audio channels sent to each user to two and gives terminals with limited audio features (such as telephones and current PCs with audio capabilities) access to audio windowing.

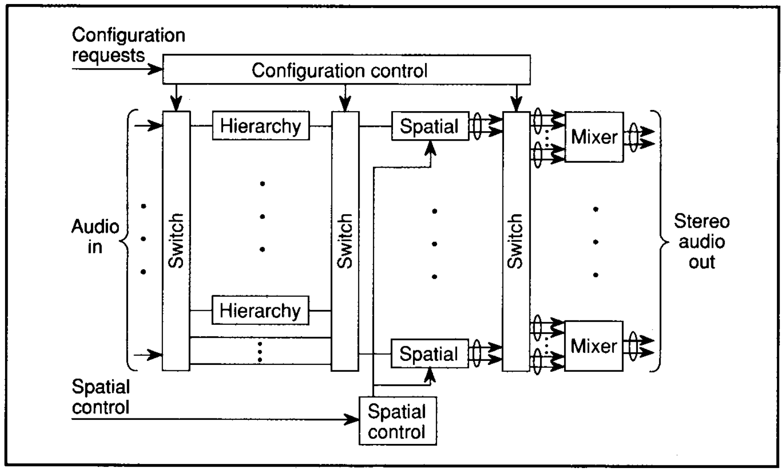


Figure 5. A network-based audio window server.

Working together. There may be cases where both methods must be used simultaneously. For example, a user might want to impose local audio sources atop the signals provided by a network-based audio window server.

Our system lets terminal and network-based methods work together by exploiting points of linearity in the audio windowing processing pipeline and by exploiting points between processing stages. Since the final stage involves a linear mix of the outputs from several spatial processing sections, the final mixing stage can be split into a network-server part and a terminal part. The outputs from the network-server part are transmitted through the network to the terminal, where they become inputs to a final mixer in the terminal. This mixer can also receive the outputs of local spatial-processing stages in the terminal. To exploit points in the pipeline, hierarchical processing can be invoked separately from the spatial processing. (Note that the order of hierarchical and spatial processing is most likely not interchangeable except with careful design.) The development of protocols to let audio window systems work together at control level remains an unexplored area.

The possibilities and potential for audio windowing create many needs and opportunities for future work. For example, we have developed but not tested an audio cursor, modes that have sources in motion, audio icons, very distant sources, and multi-index extensions of the one-index hierarchy. A good deal of formal subjective testing is needed to study such issues as user fatigue and the effectiveness of techniques and metaphors. A psychologist has worked out a subjective testing form and procedure, but we have not been able to perform the testing yet. Audio window systems invite hardware development as well. Better pitch shifters (without glitches or flanges) are needed, while very large scale integrated circuits could make audio windowing accessible to terminal designers and value-added network-service providers.

An important area still unexplored is the treatment of bursty (sporadically active) audio sources. With or without a complementary graphical display, some method is often needed to establish the position of these sources when they are dormant. One approach might deliberately introduce low-level background noise as a continuous reminder of the bursty source's existence. Rare audio events signaling some condition that disappears when acknowledged (such as an alarm or a phone ring) could be treated as a pop-up audio window. ■

Acknowledgments

We thank Ying Wu for computer interface assistance (hardware and drivers), Scott Foster for Convolvatron support, Renee Doctor for subjective testing methods, and Mark Levine, Damon Antos, and Sarah Martin for X.11 help. We also thank the editors and Mary Lea Crawley for their extensive editorial assistance.

References

1. "Multimedia Stirs a Revolution in Corporate US: Apple, IBM, Others Rush into New Market," *PC Week*, Vol. 6, No. 35, Sept. 4, 1989, p. 109.

2. B. Arons et al., "The Vox Audio Server," *Multimedia 89*, IEEE.

3. W. Slawson, *Sound Color*, University of California Press, Berkeley, 1985.

4. J. Blauert, *Spatial Hearing*, MIT Press, Cambridge Mass., 1983.

5. G. Kendall and W. Martens, "Simulating the Cues of Spatial Hearing in Natural Environments," *Proc. 1984 Int'l Computer Music Conf.*, Paris, pp. 111-125.

6. R.A. Bolt, *The Human Interface*, Van Nostrand Reinhold, New York, 1984.

7. I. Greif, *Computer-Supported Cooperative Work*, Morgan Kaufmann, San Mateo, Calif., 1988.

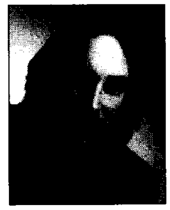
8. L. Ludwig and D. Dunn, "Laboratory for Emulation and Study of Integrated and Coordinated Media Communication," *Proc. ACM SIGComm*, 1987.

9. L. Pate and R. Lake, "A Network Environment for Studying Multimedia Network Architecture and Control," *Proc. Globecom 89*, IEEE, pp. 1,232-1,236.

10. L. Ludwig and N. Pincever, "Audio Windowing Realization," tech. report, to be submitted to *IEEE Acoustics, Speech, and Signal Processing* (contact Ludwig at the address below).

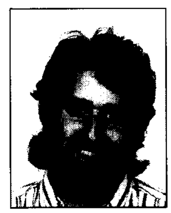
11. E. Wenzel, F. Wightman, and S. Foster, "A Virtual Display System for Conveying Three-Dimensional Acoustic Information," *32nd Human Factors Soc. Proc.*, 1988, pp. 86-90.

12. M. Cohen and L. Ludwig, "Multidimensional Audio Window Management for Spatial Sound," to be published in *Int'l J. Man Machine Studies*.



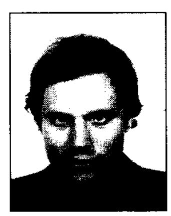
Lester Ludwig is principal scientist at Vicor, Inc. Previously, he established Bell Communications Research's Integrated Media Architecture Laboratory in 1986. His interests include multimedia telecommunications, stochastic control of resource allocation, scientific visualization, signal processing, human interface systems, and wide-area wide-band networking.

Ludwig holds a PhD in electrical engineering and computer science from the University of California at Berkeley, and MS and BS degrees in electrical engineering from Cornell University.



Natalio Pincever is working on his MS degree in media technology at the Massachussetts Institute of Technology's Media Laboratory. His research interests include digital audio processing, desktop audio, human interfaces, and multimedia communications.

Pincever holds a BS degree in electrical engineering from the University of Florida. He is a member of the IEEE Computer Society.



Michael Cohen is a PhD student in the Electrical Engineering and Computer Science Department at Northwestern University, where he is researching spatial sound with the Computer Music Group. His interests are in telecommunications semiotics, including stereotelephonics, digital typography, and hypermedia.

Cohen received his BS in electrical engineering from Brown University and his MS in computer science from the University of Washington. He is a member of ACM.

Readers can contact Lester Ludwig at Vicor, 525 University Ave., Suite 1307, Palo Alto, CA 94301.