HOW FAR AWAY IS PLUG 'N' PLAY?
Assessing the Near-Term Potential of Sonification and Auditory Display

Robin Bargar
Virtual Environment Group
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

INTRODUCTION

Sound is gradually making its way into virtual environments (VE). This presentation addresses the state of the sonic arts in scientific computing and VE, analyzes research challenges facing sound computation, and offers suggestions regarding tools we might expect to become available during the next few years. Sound immerses us in an acoustic world of rhythmic and melodic messages and environmental and spatial cues. Remove the sounds in our real world and we will be less certain where we are. When sound is included in VE, users begin to rely upon it for similar environmental orientation.

Since VE's are predominantly graphical display environments, we include discussion of sound relative to the computation and display of visual information. For many of us, the cinema provides formative experiences of sound in visual environments. The cinematic model creates strong expectations regarding the roles sound plays and the places we will be able to hear it. Sounds in VE fill many cinematic roles, giving an environment a more continuous sense of presence and providing information to enhance or reinforce visual display.

A list of classes of audio functionality in VE includes *sonification* - the use of sound to represent data from numerical models; 3D auditory display (*spatialization* and *localization,* also called *externalization*); navigation cues for positional orientation and for finding items or regions inside large spaces; voice recognition for controlling the computer; external communications between users in different spaces; and feedback to the user concerning his own actions or the state of the application interface.

To effectively convey this considerable variety of signals, we apply principles of acoustic design to ensure the messages are neither confusing nor competing. Acoustic design requires the talents of musicians and composers to ensure a listener does not experience auditory fatigue. At NCSA we approach the design of auditory experience through a comprehensive structure for messages, and message interplay we refer to as an *Automated Sound Environment.* We implement classes of auditory messages as high-level functions in a software environment for rendering sounds. Our research addresses four engineering and communication challenges: real-time sound synthesis, real-time signal processing and localization, interactive control of high-dimensional systems, and synchronization of sound and graphics. Each of these represents a set of hardware-software engines needed by the general VE community in order to effectively use sound. Such engines are not at this time commercially available. In the following pages we discuss some of the principles involved in these tools, practical issues surrounding their implementation, and examples of their application in working VE systems.

AN ACOUSTIC OBSERVATION-FEEDBACK CYCLE

Observation in VE depends upon interaction between an observer and a computational model. Numerical computation is reflected upon through the cognitive process of an observer. To achieve a reflection we need an auditory display interface and a control input from the observer to the computation. Observation includes the control gestures input from an observer investigating the system and the cognitive processing of acoustic feedback generated by the resulting state of the computational model. "Sounds" and "events" indicate the acoustic signals from the interface have been transformed into auditory signals by a listener. The terms *qualitative* and *quantitative* denote this transformation. In distinction to a practice of referring to the "qualities" of numerical data, our proposition is that numerical models have intrinsic properties, but these properties do not have "qualities" until they are perceived through the actions of an observer and an interface [1].

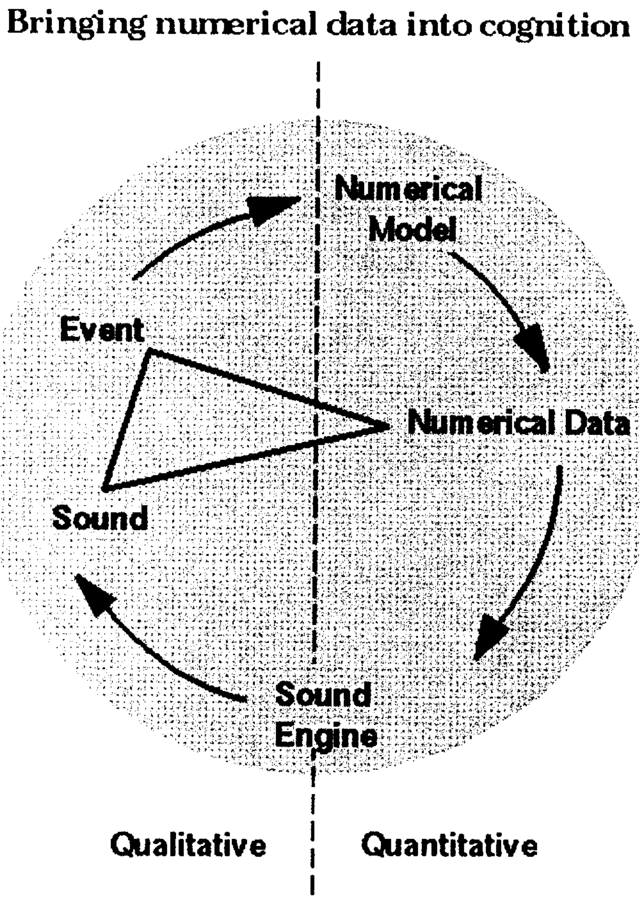


Figure 1

CONTROL FLOW ARCHITECTURE

Sonification is the rendering in sound of scientific data from a numerical model. It is one of the least explored functions of auditory display. Computer-synthesized sound is controlled by numerical data so it is possible to construct a control flow from a scientific model to a sound synthesis engine. However, there is no guarantee the scientific data will produce intelligible auditory information. A sound designer determines an appropriate mapping between the two systems. The diagram below accounts for two design stages: (1) the creation of a sound synthesis engine capable of producing a known and controllable range of sounds, and (2) the creation of an expressive relationship between the sound synthesis capability and the characteristics of the scientific data.

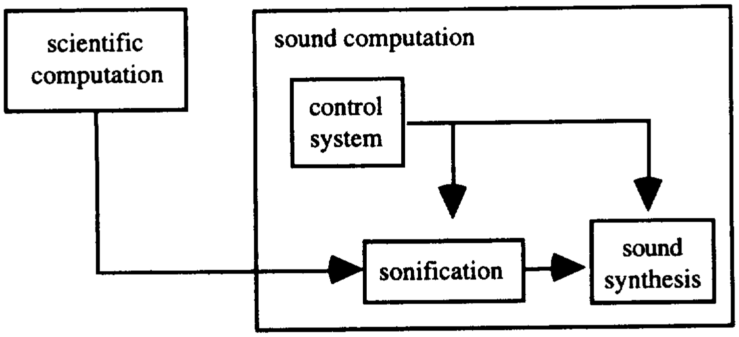


Figure 2

WHAT WE HEAR IN AN ACOUSTIC SIGNAL

Sound is characterized by energy distribution in the frequency domain and by rapid changes in the time domain. Sampling rates of 48 kHz or better are needed to encode a signal compatible to the perceptual range of the human ear. To observe the structure of sound we can decompose a signal into a series of discrete short-time Fourier transforms. Our ears are remarkably sensitive to small changes in energy in the frequency domain, over time. The diagram below shows the structure of a small sample of the steady-state portion of a tone played by a trumpet. The physical structure of the trumpet provides a resonating column of air; its resonant characteristics can be seen in the regular distribution of energy peaks in the frequency domain. These energy peaks are called *partials* or *harmonics*. They determine the tone quality of a sound. Even distributions present a listener with tonal attributes such as pitch; irregular distributions create noise-like characteristics. In the figure note the complexity of the energy peaks, highly structured but irregular; also note the amount of acoustic information discarded as the same signal is reproduced at lower sampling rates. These features describe the two most elusive objectives of real-time sound synthesis: (1) to generate complex harmonic structures, (2) at high sampling rates.

Trumpet tone at decreasing Sample Rates

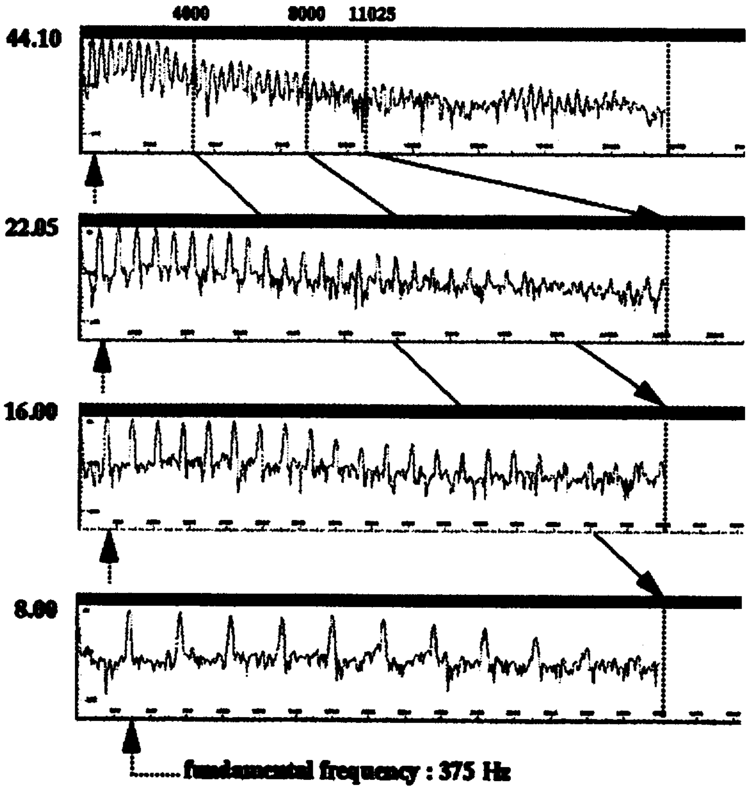
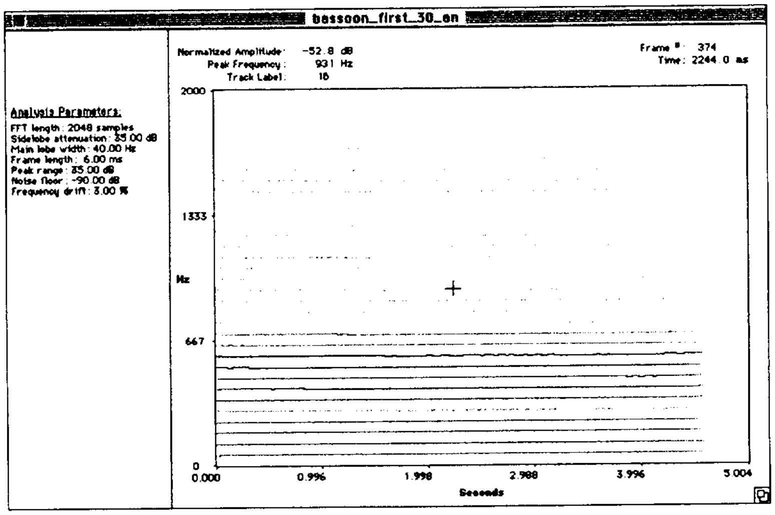


Figure 3

SOUND EVOLVING IN TIME

In natural sounds, frequency-domain structures evolve in complex ways in the time domain. This upper figure shows the energy peaks in a single bassoon tone: frequency is depicted on the vertical axis, time on the horizontal axis and amplitude by greyscale. The lower figure provides a better view of the amplitude evolution over time. The regular distribution in frequency and stability of peak locations in time indicates that the tone is quite harmonic (having a well-tuned pitch). Note even with this regularity the high degree of complex variation in local structure. The human ear is very good at comparing one such structure with another. The potential to hear distinguishing features in resonating systems at this level of acoustic structure encourages the pursuit of sonification tools for studying high-dimensional data that may have hard-to-detect regularities.



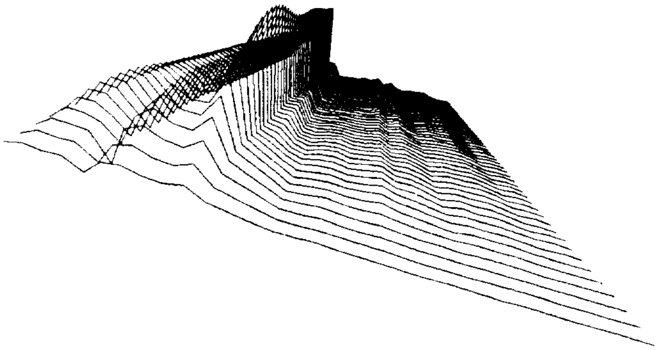
bassoon_first_30_on

Normalized Amplitude: -52.8 dB
Peak Frequency: 931 Hz
Track Label: 16

Frame #: 374
Time: 2244.0 ms

Analysis Parameters:
FFT length: 2048 samples
Sidelobe attenuation: 85.00 dB
Main lobe width: 40.00 Hz
Frame length: 6.00 ms
Peak range: 85.00 dB
Noise floor: -90.00 dB
Frequency drift: 3.00 %

Spectral analysis of a bassoon tone. Time on the X axis, frequency on the Y axis, amplitude indicated by darkness of lines.



The same bassoon tone analysis viewed as a spectral surface, with frequency on the X axis, amplitude on the Y axis, and time receding along the Z axis.

Figure 4

SOUND SYNTHESIS ENGINES

Two taxonomies of sound synthesis methods account for the range of solutions to the problem of generating complex signals. Dodge [2] describes three broad classes: additive accumulation of simple waveforms; modulation of one waveform by another to produce sidebands; and filtering of a broadband (noisy) signal to obtain desired energy peaks. Each of these produces a steady-state waveform with controllable harmonic and noise characteristics. A waveform may be generated by continuous functions or lookup tables with a corresponding tradeoff between flexibility and computational efficiency. To obtain waveforms varying in time, additional control signals are applied to the amplitudes and frequencies of the source signals during the course of a synthesized sound. The problem of organizing the control signals in efficient and structured ways remains unsolved. Smith [3] provides a classification of synthesis strategies organized by models. The models provide varying degrees of criteria for time-domain evolution of the signal. Digitized sounds are already complex signals; it is difficult to manipulate them to produce different sounds. Spectral models organize the trajectories of energy peaks in a sound over time; analyses of natural sounds may be used to obtain guidelines for the time-based control signals that are required. Physically-based models describe coupled excitor-resonator systems with sets of ordinary differential equations. These provide efficient time and frequency descriptions; however, they are difficult to control and offer many unpredictable solutions. Smith's last category is a catch-all for systems that do not follow models based upon the reproduction of natural sounds.

Dodge:

Additive   Distortion   Subtractive

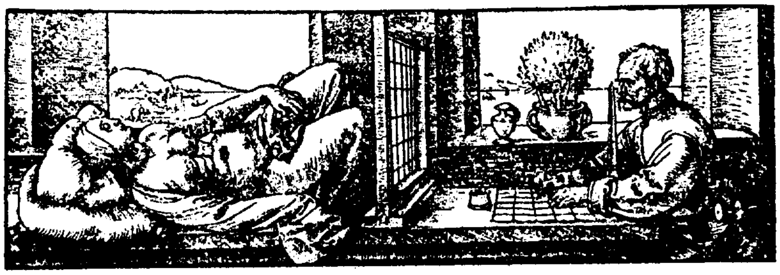
Smith:

Processed Recording   Spectral Model   Physical Model   Abstract Algorithm

Figure 5

AN EARLY VIRTUAL ENVIRONMENT

In the early 16th century Albrecht Dürer recorded the research efforts of visual artists to harness the principles of linear perspective. Historically it is noteworthy that the artists' efforts pre-date those of geometers to understand Euclidean projection in graphical terms [4]. This etching portrays mechanisms that also operate in a VE system, particularly knowledge of the user's position and orientation with respect to other objects, and the capability to render visual information accordingly. Consider what might be the acoustic analogy to the visual systems depicted here. One analogy is *localization*, the presentation of sounds from various positions and distances measured with respect to the user. Visually there are a number of relations the artist can obtain by the use of perspective, in addition to the representation of distance. The position of the frame provides a particular discourse concerning the positions of the objects that are framed. The frame not only defines an observer's relation to the scene, it defines the relations of the objects within the scene to one another. We may ask, in sound do we have analogies to these visual *frames of reference*? One analogy is the implicit need for a model of the space shared by the listener and sound sources, a space in which the sound reverberates. Unlike light, we attend to sound simultaneously in all directions. Another analogy is the need to compare sounds with one another, to arrive at complex relations and subtle meanings such as relative degrees of importance or degrees of similarity and differences among objects which the sounds represent. Research by musicians and composers will be of great benefit to creating acoustic frames of reference in VE systems.



Albrecht Dürer: *Man Drawing Reclining Woman.* Engraving from *Measuring Instructions*, printed in Nuremberg, 1525. Kupferstichkabinett, Berlin.

Figure 6

VISUAL FRAMES OF REFERENCE

Theses images demonstrate the influence of neighboring patterns upon the perception of a whole. Curvilinear groups convey different messages depending upon their re-contextualization by other groups. None of the groups below have a strong representational function in isolation. Together, converging lines become a road, vertical lines become poles and curves become a mountainous horizon. Can we say absolutely that these figures do or do not convey these meanings? Together, they convey *my intention* to convey these meanings, an intention in which you participate if you also see the contents I enumerate. Again inviting analogy to sound, we wish in VE to assemble acoustic signals to convey meaningful inter-relations rather than abstract figures. Let us also understand that acoustic or visual messages do not emerge from scientific or engineering data without the presence of intentional designs to enable the assemblage of meaningful relations according to principles of perception and cognition.

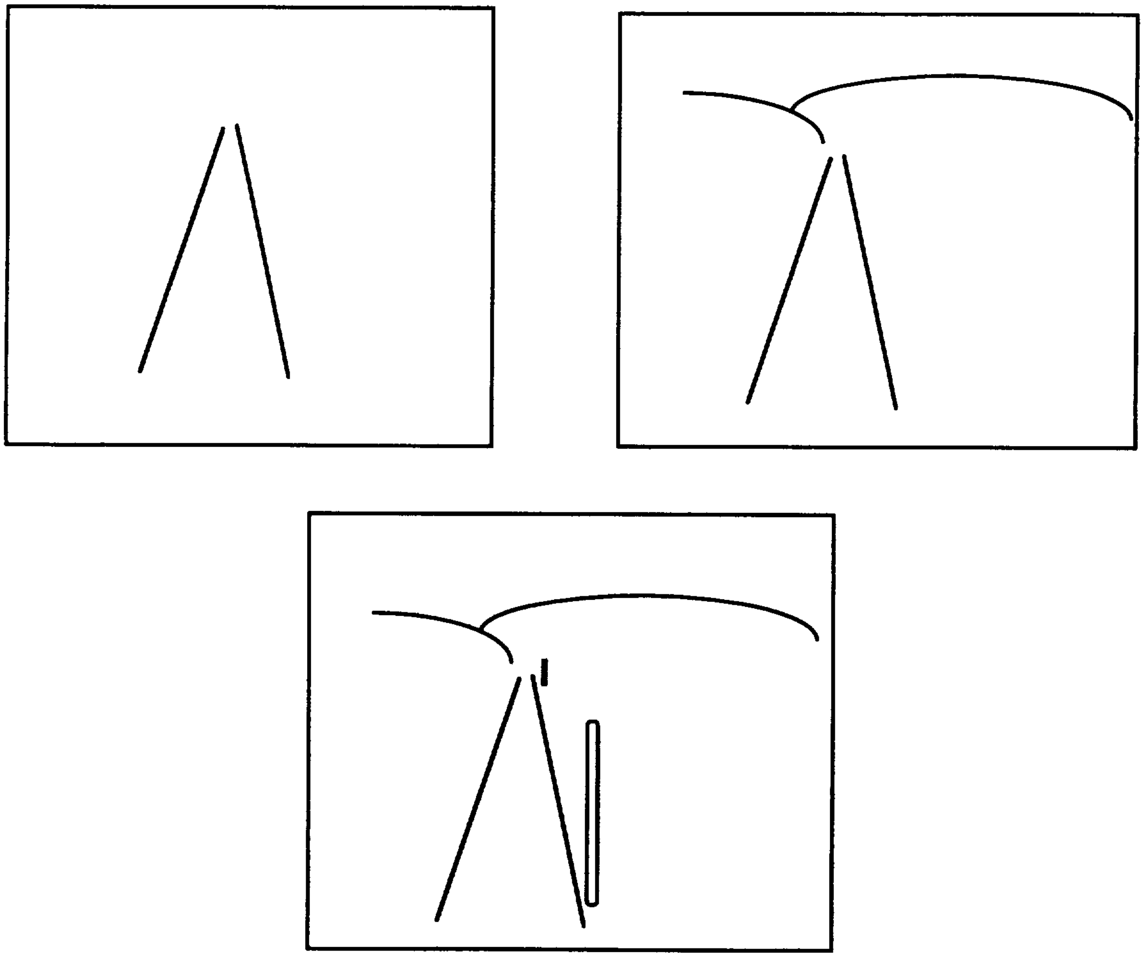


Figure 7

ACOUSTIC FRAMES OF REFERENCE

An excerpt from a string quartet from Haydn [5] provides examples of acoustic frames of reference constructed from abstract figures in sound. The musical staff orders the instruments by ascending frequency range, 'cello, viola, second violin, first violin. Vertical lines across all four parts indicate the time passing in measures. Vertical coincidence of notes indicates simultaneity. Throughout most of this example the first violin has a more active part, supported by the others making more regular sounds that change more slowly. A discourse is established in reference to a small collection of musical patterns, which may be shared among the players. Significant changes are perceived not on a note-by-note basis, but across the discourse of patterns.

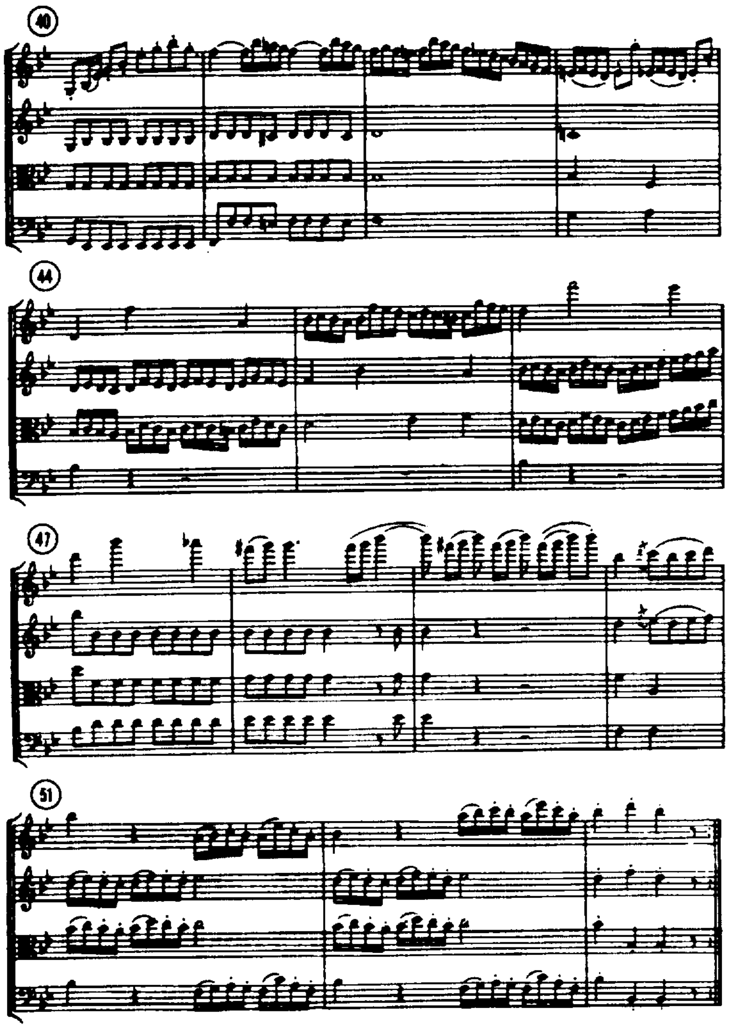


Figure 8

For example, at measure 40 violin 1 ascends in an ornamented passage while the others play together in a steady pulse; at m. 42 the lower three instruments sustain single tones while violin 1 descends through the acoustic space opened up in the previous two measures. In music this *solo-accompaniment* relation is similar to visual *figure-ground* systems. A conversation begins in m. 44 as violin 2 and viola trade patterns with violin 1. The 'cello rests in mm. 45 and 46, providing a silence in the lowest frequency range. One function of silence is to emphasize a sound upon its return, such as the return in m. 47 of the lower the instruments' accompaniment role against a loftier violin 1. Another role of silence is to emphasize a sound by isolation, as violin 1 solo reaches a peak in m. 49 while the others rest. In mm. 50-53 the conversation and rests are redoubled and shared by all players, reaching a temporary conclusion and punctuation when all play and rest together. The terminal symbol on the musical staff indicates the passage will be repeated. The composer chooses repetition for structural emphasis before going on to new material.

CONSTRUCTING A MULTI-MODAL EXPERIENCE

The cinema is the dominant paradigm for audio-visual messages. This figure represents essential cinematic features: images in discrete frames that hold the screen unchanged when they are displayed, while sounds accompany the images in a continuous signal, having no notion of "frame." The dichotomy, motionless image - frameless sound, carries over into digital media, and with it come a host of complications regarding the conjunction of sound and image. Many of these complications have never been resolved in the cinema; instead, the industry adopted work-arounds that are now communications conventions. Computer-based media are capable of finding new technical solutions to image-sound incompatibilities; in so doing we may challenge existing communications conventions. Issues that arise in delivering a real-time audio-visual message stream include time-critical computing in UNIX, negotiated graceful degradation when processes overtax the CPU, separation of VE applications from an application framework, and locating outside of the application program specialized engines such as physics modules. We have already touched upon basic sound modeling; next we will discuss rendering, synchronization and display.

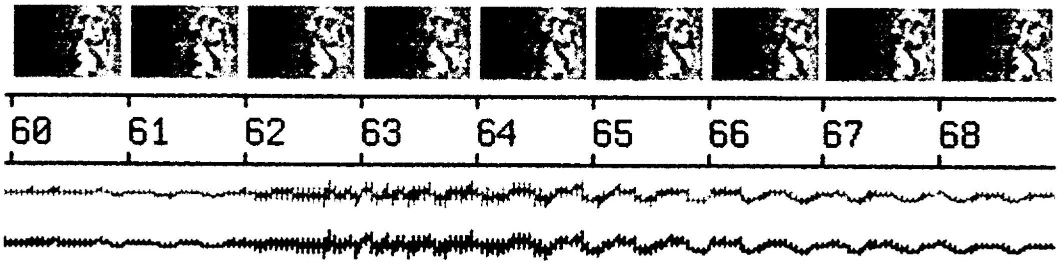
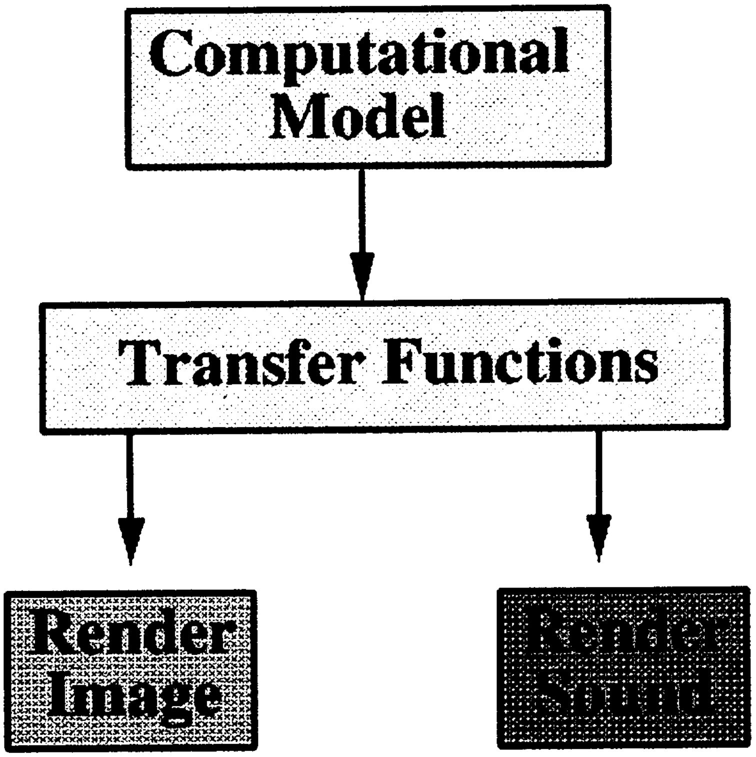


Figure 9

PARALLEL RENDERING PIPELINES

We propose an alternative to the cinematic model. In the cinema, sounds and images may be captured from anywhere and placed together on the film. In our alternative system, sounds and images come from the dynamics of a single numerical model. This rigorous restriction defines new boundaries for audio-visual communications. The knowledge that sound and image are both originating in a single source model allows experimental observations about the state of the underlying model. This sort of observation is not part of the conventional cinematic experience. Using parallel rendering pipelines we may be able to represent experimental data with cinema-like, naturalistic display strategies. Research is needed to investigate and design transfer functions for extracting control signals for image and sound rendering.

Ideal: Parallel Rendering Pipelines



Computational Model

Transfer Functions

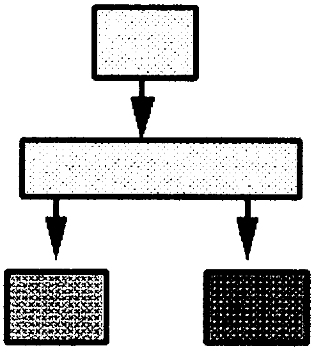
Render Image

Render Sound

Figure 10

PARALLEL RENDERING PIPELINES: FEATURES

The capability to generate both sounds and images from a single apparatus, the computer, offers desirable features for developing robust audio-visual correlations for making experimental observations.

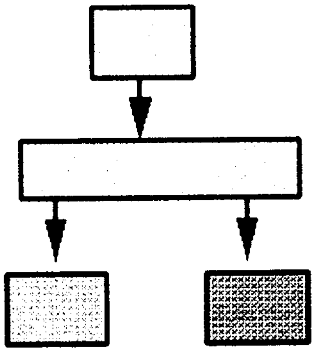


- Single Hardware Platform
- Single OS and File System
- Single Programming Language
- (Eventually: Single Frame Rate)
- Timing controlled at top or bottom

Figure 11

PARALLEL RENDERING PIPELINES: GRAPHICS

Hardware manufacturers of advanced graphics systems provide sophisticated hardware and software rendering pipelines. Many of these operations are available by simple function calls in high-level programming languages. Graphical scenes operating according to complex real-time dynamics may be rapidly prototyped.

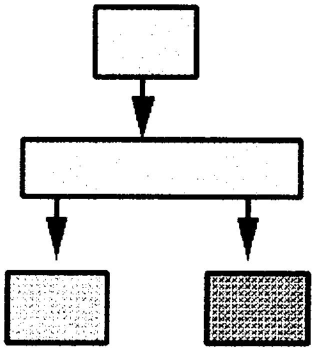


Atmospheres
Textures
Lighting
Color
Clipping
Shading
Matrix operations
2D and 3D Primitives
Pixels

Figure 12

PARALLEL RENDERING PIPELINES: SOUNDS

If we look for hardware and software support of sound rendering on general-purpose computing platforms, we find no such architecture in existing commercial systems. High-fidelity sound rendering requires fast floating-point computation, a D/A converter and drivers, and an audio sample-buffer and scheduler protected from system interrupts. Multi-media systems on the market do not address general-purpose high-fidelity sound rendering. Multi-media systems are currently geared toward low-power desktop machines with special hardware support devices, and offer linear reproduction of sound and image sequences that were created on non-real-time platforms and are primarily non-interactive. High-level computing platforms which have the power to render sound in real-time have so far not been targeted for development of the necessary converters, drivers and libraries. Considering the capability of sound to assist in the interpretation of computations performed on powerful platforms, the lack of support for audio takes on the appearance of an oversight, or at best a lack of imagination.



Atmospheres	
Textures	**?**
Lighting	
Color	
Clipping	
Shading	
Matrix operations............	*"play soundfile"*
2D and 3D Primitives......	*soundfiles*
Pixels...............................	*sound samples*

Figure 13

THE NCSA SOUND SERVER

The NCSA Audio Development Group conducts research and provides software prototypes to address the need for a real-time interactive sound rendering system to function in parallel with graphical systems. We created the NCSA Sound Server to explore the capability for sound rendering in a general-purpose computing environment [6]. The Sound Server is written in C++ and runs in UNIX, with a scheduler (HTM) optimized for high-level communications to a D/A converter architecture in real-time [7]. The Server includes libraries for sound synthesis and signal processing (VSS), and high-level "Actors" containing networks of transfer functions for translating numerical signals into intelligible acoustic patterns. Communications protocols allow our libraries to be controlled from client applications. Client and server may run on separate machines, passing messages using the serial udp protocol. An interface configuration file format allows the control of the mapping between client and server at run time. This is critical for practical purposes as it allows sound design to be located outside of the client application, increasing the likelihood of immediate interactive testing using the client as a sound controller to provide actual data conditions.

CLIENT - SERVER ARCHITECTURE
NCSA SOUND SERVER

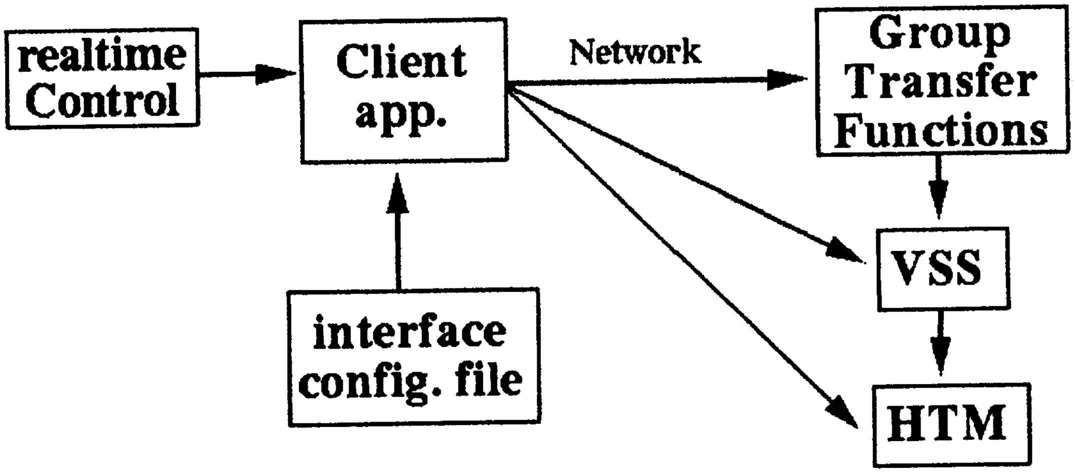


Figure 14

THE NCSA SOUND SERVER: FEATURES

Advantages typically associated with client-server architectures provide a favorable media development environment for applying sound to scientific computation.

Client-Server Advantages

- **Less code to merge - prototypes easily**
- **Audio code remains independent and stable**
- **VE client becomes synthesis interface**
- **Clients run on platforms other than SGI**
- **Sound synthesis in real-time in UNIX**

Figure 15

PARALLEL RENDERING PIPELINES IN THE CAVE

The EVL-NCSA CAVE© provides a testbed for applying sound rendering in parallel to graphics environments. Most CAVE applications include a computational engine that models the environment as well as the graphical rendering functions. CAVE clients link to audio libraries at compile time. The client application typically runs on a dedicated multiprocessor machine, while the sound server requires an interrupt-protected CPU and usually runs on a dedicated machine, receiving messages via serial communications.

CAVE Client – Sound Server Architecture

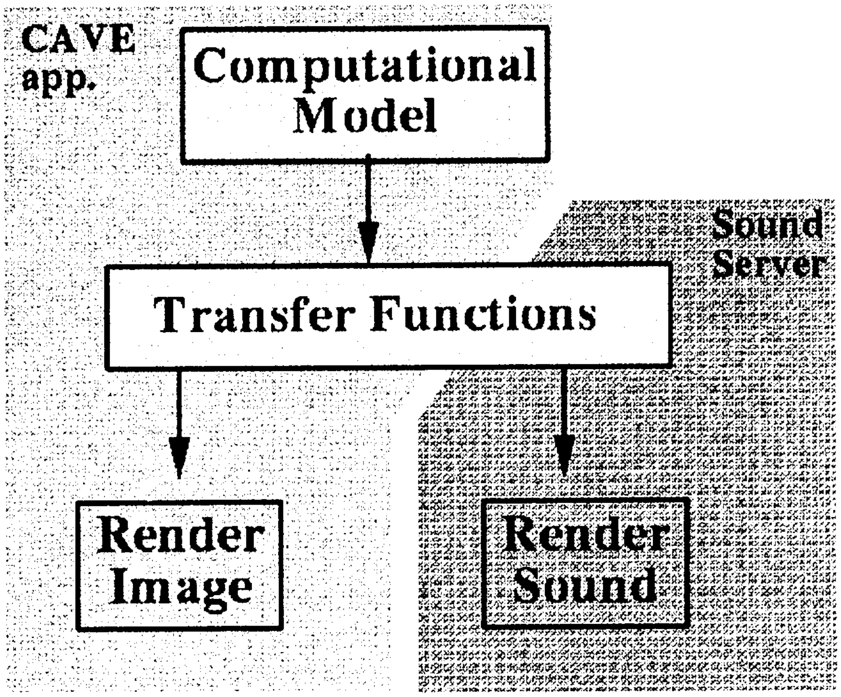
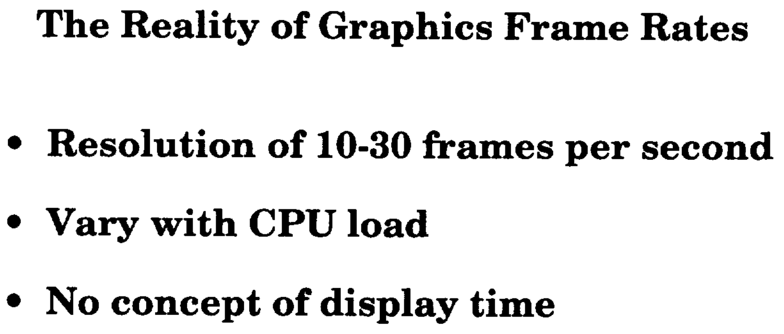


Figure 16

SOUND-IMAGE SYNCHRONIZATION: THREE HEADACHES

Three attributes of standard graphical rendering architecture contradict the needs of sound rendering systems. First, high-fidelity sound requires a sample-loop execution 48,000 per second. Graphical frame rendering loops perform at much slower rates. Second, the display rate of rendered frames is allowed to vary radically, whereas sound needs to be displayed at a constant uninterrupted sample rate. Pauses as short as two samples in duration will create noticeable discontinuities in the form of bothersome clicks in an audible signal. Third, graphical rendering pipelines have no concept of scheduling other than "next in line" and "as soon as possible." Even if visual and audible samples are rendered at the same time in their respective pipelines, there is no way to guarantee with existing hardware that the results will reach the display devices at the same time.



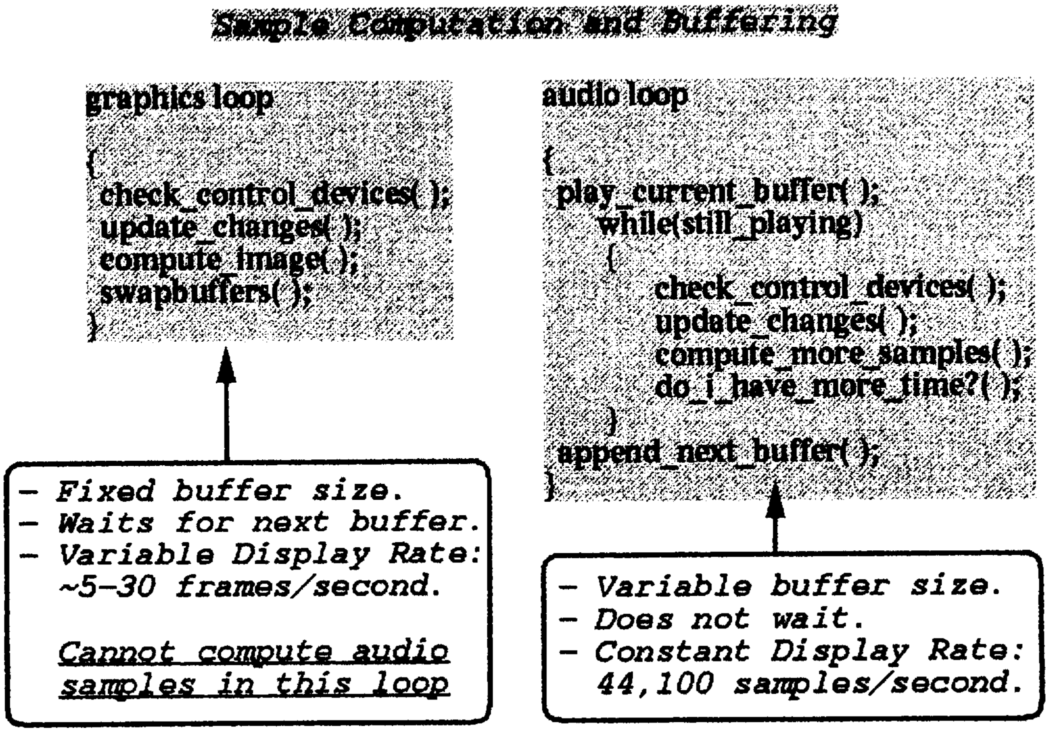
The Reality of Graphics Frame Rates

- **Resolution of 10-30 frames per second**
- **Vary with CPU load**
- **No concept of display time**

Figure 17

SOUND-IMAGE SYNCHRONIZATION: COMPETING DEPENDENCIES

In many virtual environments the update of the entire system is determined by the frame rate of the graphical display. This presents a problem for sound if it is to be synthesized within a graphics loop. Comparing the computation loops of graphics and sound samples we notice they operate on incompatible concepts of time. Graphical display is dependent upon upcoming events: the current frame remains on screen until the next frame is finished rendering. Display time varies accordingly. Auditory display is dependent upon passing events: the current sample buffer is displayed at a fixed sample rate and as soon as it is completed the next buffer must begin its display in order to avoid interruptions in the signal. Audio is computed in variable buffer-lengths to compensate for the fixed display rate. Human sensitivity to time discontinuities appears to be lower for visual signals than for audio signals: a noticeable variation in visual frame rate does not prohibit the interpretation of visual form and motion, whereas human perception of audio signals cannot tolerate a comparable degree of discontinuity in time without disrupting the cognitive imaging of a signal as the product of a sounding body in a real world.



Sample Computation and Buffering

```
graphics loop
{
 check_control_devices( );
 update_changes( );
 compute_image( );
 swapbuffers( );
}
```

```
audio loop
{
 play_current_buffer( );
   while(still_playing)
   {
       check_control_devices( );
       update_changes( );
       compute_more_samples( );
       do_i_have_more_time?( );
   }
 append_next_buffer( );
}
```

- *Fixed buffer size.*
- *Waits for next buffer.*
- *Variable Display Rate: ~5-30 frames/second.*

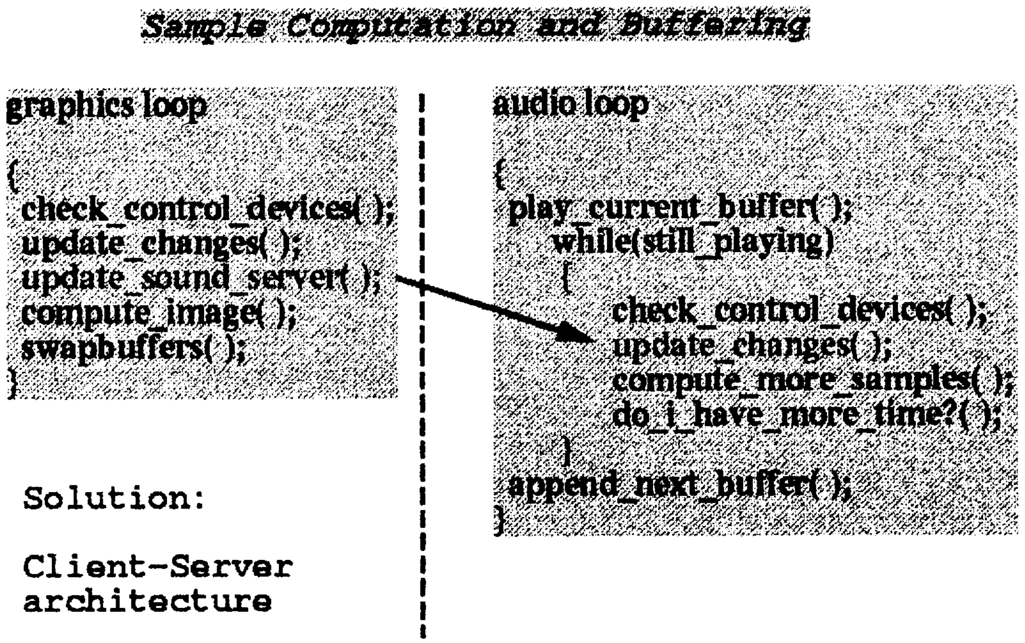
Cannot compute audio samples in this loop

- *Variable buffer size.*
- *Does not wait.*
- *Constant Display Rate: 44,100 samples/second.*

Figure 18

SOUND-IMAGE SYNCHRONIZATION: CLIENT-SERVER SOLUTIONS

A client-server paradigm permits two sample computation strategies to occur in parallel without conflicting dependencies. Sound and graphics do not share a computation loop, instead they are coordinated at two different locations: first, by sharing common data at their source; second, by high-level (but not low-level) time coordination of events. Each engine can run at its optimal rate and update under separate conditions. This requires the sound models to have sufficient intelligence to compute waveform trajectories independent of visual-based control information. Sounds update independently and receive high-level control signals from the graphical and interactive environment. These controls are generated no faster than the graphical frame rate, a good rate for phrase-level audio events as long as the integrity of the waveform evolution at 48 kHz is not interrupted. "Phrase level" events occur in sound at rates of roughly slower than 20 Hz, the rate at which a stream of changes in sound pressure level can be perceived as a steady tone.



Sample Computation and Buffering

```
graphics loop

{
 check_control_devices( );
 update_changes( );
 update_sound_server( );
 compute_image( );
 swapbuffers( );
}
```

```
audio loop

{
 play_current_buffer( );
    while(still_playing)
    {
       check_control_devices( );
       update_changes( );
       compute_more_samples( );
       do_I_have_more_time?( );
    }
 append_next_buffer( );
}
```

Solution:

Client-Server architecture

Figure 19

THE NCSA SOUND SERVER IN THE CAVE

CAVE clients run on a multiprocessor computer with special graphics hardware, while the Sound Server (VSS) runs on a separate dedicated platform with the necessary D/A conversion hardware. Downstream from the server the audio signal is multiplied in a signal matrix and signal processing is applied. In this way multiple sounds are independently localized in a 2D or 3D distribution of speakers, and distance cues *(externalization)* are applied. Positional values and moving sound sources are controlled from the CAVE application. The CPU-intensive nature of simulated localization and externalization requires dedicated hardware. This hardware is controlled from the Sound Server using the MIDI (Music Instrument Digital Interface) serial communications protocol.

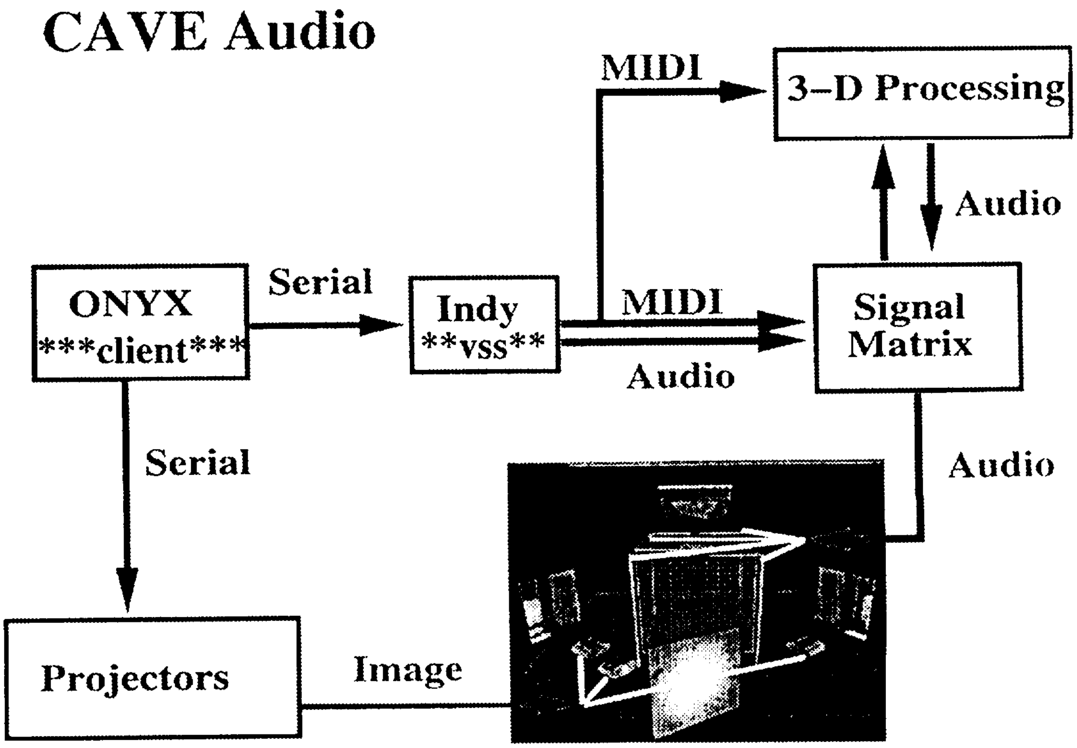


Figure 20

AN EXAMPLE APPLICATION: THE SOUND OF CHAOS

We have explored the sound of signals from the Chua's circuit, an experimental electronic circuit designed for the study of chaos [8]. In the CAVE we control a numerical simulation of the Chua's circuit with a *manifold interface* designed to allow gesture-based control of high-dimensional systems [9]. We display a graphical surface representing a control region and a cursor for navigating the surface using a gesture-based control device such as a 3D mouse or wand. In the same visual space we superimpose a phase portrait of the output signal of the three ODE's that simulate the Chua's circuit [10]. To obtain sound from the simulation the samples from one of the ODE's are sent to the Sound Server scheduler and converted directly into an audio signal. The sound changes radically during bifurcation scenarios from steady, pitched tones to regular and irregular rhythmic pulses, and then to bandpass-like noise as the state of the system moves from periodic to intermittent and chaotic regions. We cannot pass the sound samples from the CAVE client to the Sound Server in real-time at a 48 kHz rate, so we run the ODE's both in the client to obtain a visualization of the signal, and in the Sound Server (at a higher sample rate) to obtain the audible signal. The two sets of Chua's equations remain in very similar states because both are controlled in real-time by gestures from the manifold interface.

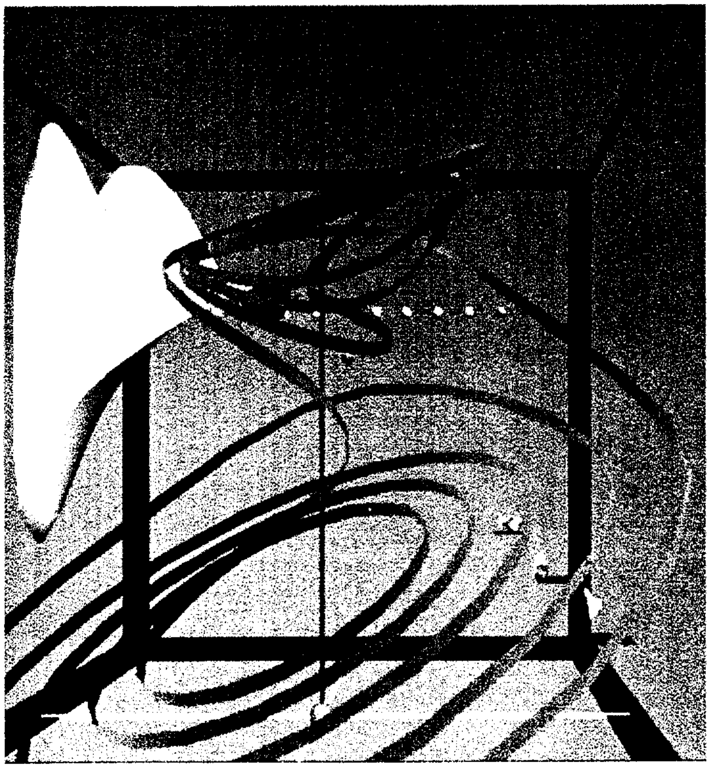


Figure 21

CONCLUSIONS

The commercial music industry offers a broad range of "plug 'n' play" hardware and software scaled to music professionals and scaled to a broad consumer market. The principles of sound synthesis utilized in these products are relevant to application in VE. However, the closed architectures used in commercial music synthesizers are prohibitive to low-level control during real-time rendering, and the algorithms and sounds themselves are not standardized from product to product. Thus a given control signal produces different results on different synthesizers. To bring sound into VE requires a new generation of open architectures designed for human-controlled performance from interfaces embedded in immersive environments.

The implementation of interactive sound synthesis in a general computing environment is a step toward "Plug 'n' Play" audio functionality in VE. Both the graphical computing and digital audio communities are just beginning to awaken to the potential needs of researchers and artists for these types of integrated tools. The NCSA Audio Group is developing high-level libraries that can be called from client applications to create well-structured audio environments. These respond to the states of a client application with special sound signals or subtle changes to the acoustic ambiance in a VE display. We desire to keep our functionality in software as much we can, with obvious tradeoffs between low-level control and speed of execution. In software we have the greatest chances of developing a uniform set of protocols to be used and upgraded by the scientific computing community. Hardware manufacturers need to be encouraged to include audio hardware, device drivers and synthesis strategies as part of the standard tool set provided for scientific computing environments.

For further information regarding the NCSA Audio Development Group please visit our web page at *http://www.ncsa.uiuc.edu/VEG/audio* .

REFERENCES

1. Choi, I., "Sound Synthesis and Composition Applying Timescaling Techniques for Observing Chaotic Systems," *Proceedings of the Second International Conference on Auditory Display, ICAD '94*, G. Kramer (ed.), Sante Fe Institute, Santa Fe, 1995, pp. 79-107.
2. Dodge, C. and Jerse, T., *Computer Music: Synthesis, Composition and Performance*, Schirmer Books, NY, 1985.
3. Smith, J. O., "Viewpoints on the History of Digital Synthesis," *Proceedings of the International Computer Music Conference*, Montreal, CA, 1991, pp. 1-10.
4. Panofsky, E., "Dürer as a Mathematician," *The World of Mathematics*, J. R. Newman (ed.), Redmond, WA, Tempus, 1988, pp. 595-612; reprint of *The World of Mathematics*, Simon and Shuster, NY, 1956; original from Panofsky, E., *Albrecht Dürer*, Princeton University Press, 1943.
5. Haydn, F. J., "Finale," String Quartet Op. 74, No. 3, (Hob. III/74), 1793, *Anthology for Musical Analysis*, C. Burkhart (ed.), Holt, Rinehart and Winston, NY, 1986, pp. 153-165.
6. Bargar, R., Choi, I., Das, S. and Goudeseune, C., "Model-Based Interactive Sound for an Immersive Virtual Environment," *Proceedings of the International Computer Music Conference,* Aarhus, Denmark, 1994, pp. 471-474.
7. Freed, A., "Tools for Rapid Prototyping of Music Sound Synthesis Algorhythms," *Proceedings of ICMC*, San Jose, Oct. 1992.
8. Chua, L. O., Wu, C. W., Huang, A. and Zhong, G. Q., "A Universal Circuit for Studying and Generating Chaos, Part I: Route to Chaos," *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 40, No. 10, Oct. 1993.
9. Choi, I., "Interactive Exploration of a Chaotic Oscillator for Generating Musical Signals in Real-Time Concert Performance," *Special Issue: Chaos and Nonlinear Dynamics, Journal of the Franklin Institute,* June 1995.
10. Kennedy, P. M., "Three Steps to Chaos, Part II: A Chua's Circuit Primer," *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 40, No. 10, Oct. 1993, pp. 657-674.