

COOLVR: Implementing Audio in a Virtual Environments Toolkit

Jarrell Pair and Rob Kooper
Virtual Environments Group
Graphics, Visualization and Usability Center
Georgia Institute of Technology
College of Computing
801 Atlantic Drive
Atlanta, Georgia 30332-0280
404-894-4488
jarrell@cc.gatech.edu, kooper@cc.gatech.edu

ABSTRACT

COOLVR (Complete Object Oriented Library for Virtual Reality) is a toolkit currently being developed at the Graphics, Visualization, and Usability Center (GVU) at Georgia Tech. The toolkit is written to allow programmers to easily create virtual environments (VE's) which will compile cross platform. Unlike most VE toolkits which focus effort primarily on the visual senses, COOLVR aims to equally engage both the sense of sight and the sense of hearing. One of the main design goals of the COOLVR toolkit is to give the programmer an intuitive method to enrich the virtual world with auditory cues. COOLVR uses a set of cross platform audio rendering modules to conduct real time sound processing. By providing potential designers with the capability of easily integrating spatial audio in a virtual world, a heightened level of immersivity or presence can be achieved in COOLVR environments.

INTRODUCTION

Designers of virtual environments seek to create a feeling of presence, or immersion for users. Primarily, this goal has been pursued by creating worlds with convincing 3D interactive graphics. This sense of presence is achieved by chiefly engaging the human sense of sight. Unlike sight, the sense of hearing is often neglected in the implementation of a virtual world. Recent work indicates that the integration of spatial audio in a virtual environment enhances a user's sense of presence [6]. Regardless of considerable evidence on its immersive potential, audio is often banished as the poor stepchild of virtual reality. The plight of audio in interface design is explained in [2]:

Audio alarms and signals have been with us since long before there were computers, but even though music and visual arts are considered sibling muses, a disparity exists between the exploitation of sound and graphics in interfaces. . . . For whatever reasons, the development of user interfaces has historically been focused more on visual modes than aural.

This trend is in part due to technical resource limitations of computer systems. Designers were forced to sacrifice audio quality for graphics performance. However, these restrictions no longer exist. In the past several years dedicated audio ASIC's (application specific integrated circuits) coupled with fast CPU's have made it feasible to implement high fidelity, immersive audio in graphically intensive virtual environments.

COOLVR (Complete Object Oriented Library for Virtual Reality) is a toolkit currently being developed at Georgia Tech's Graphics Visualization and Usability Center (GVU). It is intended to succeed the Simple Virtual Environments (SVE) toolkit which the GVU Virtual Environments Group has used since 1992. SVE was designed primarily for developing virtual environments for the Silicon Graphics (SGI) platform. COOLVR is intended to provide a set of authoring tools that will allow a designer to quickly build a virtual environment that will run on both SGI and PC workstations. A chief design goal is to create an array of functions that will empower users to create worlds which equally engage both the sense of hearing and the sense of sight.

IMPLEMENTATION DETAILS OF COOLVR

Until recently, complex virtual environments were developed almost exclusively for high performance SGI workstations. Only these machines could provide the performance required for VE applications. However, the Windows based PC has emerged as an increasingly powerful graphics platform. In order to exploit the price versus performance advantages of the PC while maintaining the ability to run VE applications on SGI's, COOLVR was designed to compile cross-platform. Another key design decision was to create a dual set of rendering modules. One module, the graphics renderer, handles the graphical components of the world. A separate sound renderer manages the environment's audio.

Details of the Audio Renderer

COOLVR (CVR) has been designed to support a set of modular objects (CVR objects) and renderers for both audio and graphics. The term "render" is traditionally used in reference to graphics, but the concept can also be applied to audio [8]. COOLVR uses an audio renderer to accomplish real time sound processing in a virtual environment. The audio renderer will function identically to the graphics renderer in that they both render objects. However, in this case the objects will encapsulate audio.

More formally, the audio renderer implements methods to render spatial (3D) attributes to instances of digital audio sample data. The sample data is read from files (currently only .AIFF or .WAV formats will be supported) and stored in audio VR objects. These objects can be positioned in the world as can any other type of CVR object. For instance an audio object can be associated with a graphics object to give the user the illusion that audio is attached to it. If the graphics object moves in virtual space, the audio will move with it. This audio renderer also provides non-rendered playback for the implementation of ambient soundscapes.

CVR_SetRenderMode() will select the audio rendering method. Currently the following modes will be supported. These modes are intended to give the designer a wide range of audio options to utilize depending on the needs of the environment, and the performance of the platform.

- **CVR_NOTRICKS** renders the audio sample data without spatial (3D) attributes. In other words, the sample data remains unmodified. If the sample data was preprocessed with spatial or other effects, these effects remain intact and static. This mode is intended for the playback of looped ambient audio.
- **CVR_DISTANCE** renders the audio sample data accounting for absolute distance from the listener. The sound is attenuated or amplified depending on the distance.
- **CVR_STEREO** renders audio accounting for the left-right position from the listener. This mode can be used when the platform lacks adequate resources to render spatialized audio.
- **CVR_STEREODISTANCE** renders audio accounting for the left-right position and the absolute distance from the listener. Again, this mode is a viable option for environments running on platforms with performance limitations.
- **CVR_SPATIAL** renders audio accounting for the X, Y, Z position of audio in respect to the listener. This mode includes the implementation of distance rolloff employed in CVR_DISTANCE and CVR_STEREODISTANCE.

Audio Distance Cutoff

In a virtual environment, graphic objects are rendered only if they can be seen by the user. This task is accomplished through the process of visible surface determination algorithms such as z-buffering [1]. For illustration purposes, this graphics rendering technique is analogous to the audio cutoff scheme employed in COOLVR. Each audio object is assigned a maximum cutoff distance and a minimum cutoff distance. If the audio object is spatially located at a position outside the bounds defined by the cutoff distances, the sound file is not played. Because the files are not played back, computational resources are conserved. This scheme differs from the distance attenuation algorithms in CVR_DISTANCE and CVR_STEREODISTANCE. These rendering modes guarantee audio file playback with the volume level varying as a function of the distance. Audio distance cutoff makes it possible for specified audio objects to be audible in an area relatively close to the user. For example, a whispering voice would be assigned cutoff bounds spanning only a few feet. However, the

sound of a train engine would be assigned cutoffs allowing it to be played over long distances. If employed properly, this cutoff method can enhance the user's impression of realism in the environment.

The initial outline of COOLVR audio distance cutoff functions are described below. These two functions essentially define the playback bounds of an audio object.

- **CVR_SetMaxDistance()** sets the distance from the user beyond which an audio object's sample data will not be played back.

CVR_AudioSetMinDistance() sets the shortest distance from the user at which an audio object will play back.

CROSS PLATFORM CAPABILITY

COOLVR can be used to develop environments that can run on SGI's, PC's, and future platforms. The audio and graphics renderers will be written specifically for each platform. On the PC, the DirectX API (Application Programming Interface), and the OpenGL graphics library are being used to implement the graphics rendering module. PC spatial audio is rendered through calls to Microsoft's DirectSound3D API. The use of this API facilitates the use of spatial audio acceleration hardware such as Diamond Multimedia's Monster Sound. The SGI graphics rendering module is built upon OpenGL. The audio module for SGI machines will be based on head related transfer function (HRTF) convolution techniques similar to those discussed in [4] utilizing the KEMAR dummy HRTF data set [5].

Modular Approach

To be able to implement the different renderers on all platforms while maintaining a consistent interface, COOLVR is being built as a set of modules. These modules will provide a flexible, upgradable degree of functionality. The modules can be replaced with updated versions that exploit new audio hardware and libraries that become available. Similarly, several variants of a rendering module can exist for a user to interchange depending on the needs of an application.

FUTURE DIRECTIONS

The version of COOLVR currently in development is being implemented with the audio functionality demanded by the head mounted display (HMD) based virtual environments utilized at Georgia Tech's GVU Center. The audio rendering techniques implemented are intended for headphone (binaural) playback. Once the initial iteration of COOLVR is complete we plan to include functions to support speaker playback and simple voice recognition. These audio features are well suited to spatially immersive displays (SID's) such as the CAVE (Cave Automatic Virtual Environment) [3] and table projection systems similar to the responsive workbench [7].

CONCLUSIONS

Virtual environments have been primarily visually oriented. With the introduction of new audio API's and recent perceptual research, VE developers have begun to realize the indispensability of auditory cues in virtual environments. The COOLVR toolkit will enable users to intuitively implement spatial audio in both PC and SGI based virtual environments. Resultingly, COOLVR environments can be authored to achieve a high sense of presence for users.

REFERENCES

1. Catmull, E., *A Subdivision Algorithm for Computer Display of Curved Surfaces*, Ph.D. Thesis, Report UTEC-CSc-74-133, Computer Science Department, University of Utah, Salt Lake City, UT, December 1974.
2. Cohen, M., and Wenzel, E. M. The Design of Multidimensional Sound Interfaces. In W. Barfield and T. Furness III, editors, *Virtual Environments and Advanced Interface Design*, pages 291-346. Oxford University Press, New York, New York, 1995.
3. Cruz-Neira, C., Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE, *Computer Graphics Proceedings, Annual Conference Series*, 1993, 135-142.
4. Gardner, W.G., Transaural 3-D audio. MIT Media Lab Perceptual Computing Technical Report #342, July 1995.
5. Gardner, W.G., and Martin, K.D. HRTF Measurements of a KEMAR Dummy Head Microphone. MIT Media Lab Perceptual Computing Technical Report #280, May 1994.
6. Hendrix, C., and Barfield, W. The Sense of Presence Within Virtual Environments. *Presence: Teleoperators and Virtual Environments* 5, 3 (Summer 1996), 290-301.

7. Krüger, W., Bohn, C-A., Fröhlich, B., Schüth, H., Strauss, W., and Wesche, G. The Responsive Workbench: A Virtual Work Environment. *Computer* 28, 7 (July 1995), 42-47.
8. Takala, T., and Hahn, J. Sound Rendering. *Computer Graphics (Proceedings of SIGGRAPH '92)* 26, 2(July 1992), 211-220.