USER-SPECIFIC AUDIO RENDERING AND STEERABLE SOUND FOR DISTRIBUTED VIRTUAL ENVIRONMENTS

Mike Wozniewski & Zack Settel

La Société Des Arts Technologiques
Montréal, Québec, Canada
{mike/zack}@sat.qc.ca

Jeremy R. Cooperstock

McGill University Centre for Intelligent Machines
Montréal, Québec, Canada
jer@cim.mcgill.ca

ABSTRACT

We present a method for user-specific audio rendering of a virtual environment that is shared by multiple participants. The technique differs from methods such as amplitude differencing, HRTF filtering, and wave field synthesis. Instead we model virtual microphones within the 3-D scene, each of which captures audio to be rendered to a loudspeaker. Spatialization of sound sources is accomplished via acoustic physical modelling, yet our approach also allows for localized signal processing within the scene. In order to control the flow of sound within the scene, the user has the ability to steer audio in specific directions. This paradigm leads to many novel applications where groups of individuals can share one continuous interactive sonic space.

[Keywords: multi-user, spatialization, 3-D arrangement of DSP, steerable audio]

1. INTRODUCTION

Most APIs and toolkits for managing audio in virtual environments (e.g. OpenAL, DirectX, X3D) are purely concerned with the spatialization of sounds located in the scene. As a result, the audio experience is focused around one user who indeed is immersed in 3-D sound, yet lacks control mechanisms that would allow for greater interactive sonic activity. This of course is understandable, since the aim of these toolkits is to create realistic simulations for gaming and 3-D visualization, which only require the spatialization of external audio streams (sound files, line-in, etc.). In contrast, our audio architecture is based on nodes that behave simultaneously as *sound sources*, which emit audio into the scene, and *sound sinks*, collecting audio at a particular location. The nodes can therefore be used as signal processing units that perform a modification to a sound signal at some 3-D location, and emit the result back into the scene.

This type of approach towards audio processing and the representation of sound objects has several important ramifications. In particular, user-specific audio rendering of a virtual scene can easily be attained. Rather than applying amplitude differences among loudspeakers or filtering signals with a head-related transfer function (HRTF), an audio rendering is accomplished with a collection of sound sinks that each correspond to a loudspeaker. This can be thought of as a *virtual microphone* technique, where the number of mics is equivalent to the number of channels that comprise the audio display. Furthermore, this technique is not limited to standard speaker configurations, since an arbitrary number of virtual microphones can be defined. It is thus possible to render a scene for multiple participants, which leads to a number of exciting new applications that deal with group activity. Examples include virtual audio installations, collaborative musical performances, and remote gatherings in shared virtual environments.

2. A FRAMEWORK FOR MODELLING VIRTUAL AUDIO

In our framework [1, 2], a virtual scene is defined by a number of *soundNodes* arranged in a scene graph. This is a tree-shaped data structure which allows spatial transformations applied on a node to be propagated automatically to all child elements. Nodes can thus be grouped together so that they move as one through a scene, following the virtual representation of a participant. In Figure 1 for example, the user is modelled with three soundNodes that correspond to his ears and mouth. If these nodes are grouped together in a scene graph, one only needs to translate or rotate the parent node to affect the whole ensemble.

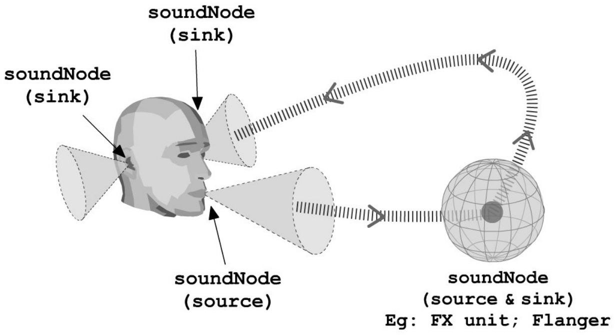

Figure 1: A simple scene modelled with four soundNodes.

This figure also illustrates the important fact that soundNodes are more than just sound sources, since the fourth node in the scene behaves as *both* a source and sink. The node is located at a particular 3-D location, and performs audio processing on any captured audio from other virtual sound sources. The modified result is then re-emitted back into the scene.

Sound effects nodes can thus be created, such as the harmonizers, flangers, or any other type of DSP that might be found in the traditional studio. However, instead of using patch cables or wires to connect different DSP units, our framework uses 3-D space as the medium for signal flow. There are no logical connections between different components, and no knobs or sliders to control levels. Rather, the audio that travels between soundNodes is processed (attenuated, delayed and filtered) according to the laws of physics. Manipulating the relative distance and orientation of soundNodes becomes the method of controlling audio processing. Thus, the principal operators for mixing (e.g. bussing, panning, filtering) become spatial in nature (e.g. translation, rotation, scaling).

2.1. Steerable Audio

With continuous 3-D space as the medium for signal processing, it is imperative that the path in which an audio signal travels is controllable. This is accomplished by specifying two steering vectors that respectively define the direction of radiation and/or sensitivity, depending on whether the node is acting as a source and/or a sink. Having two independent orientations allows a node to accept sound from one direction and radiate in a completely different direction, providing great flexibility for the organization of the audio scene.

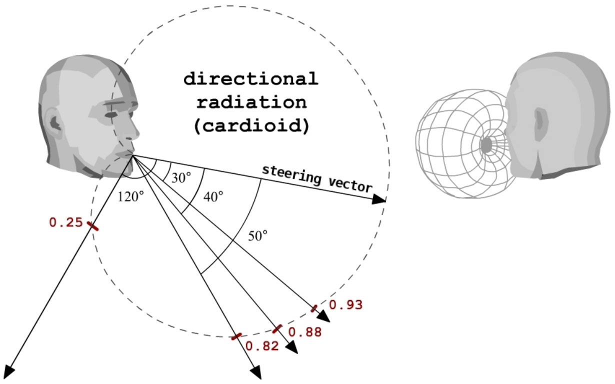

Figure 2: The directionality of a soundNode is defined by a steering vector and a rolloff function.

Associated with each steering vector is a parametric diffusion pattern, which specifies how sound intensity decays as one orients away from that direction. To illustrate, Figure 2 shows a source node and its radiation. The sound signal will radiate with unity gain along the steering vector; on either side, the gain value attenuates according to a *rolloff function*, based on the *angle of incidence*, α, to the steering vector. In the case of Figure 2, the rolloff function is a cardioid, defined by the following equation[1]:

$$\mathrm{f}\left(\alpha\right) = \left[\frac{\left(1+\cos\left(\alpha\right)\right)}{2}\right]^{\gamma} \qquad (1)$$

2.2. Physical Simulation & Bending the Rules of Physics

As sound travels between different nodes in our scene, we apply a fairly realistic simulation of audio propagation. This includes the modelling of sound decay and absorption of higher frequencies with distance, diffraction around volumes, reflections off of surfaces, and Doppler shift delays [3]. These phenomena are familiar to users since they occur in the real world, and thus offer some natural mappings. For example, the gain of a transmitted signal is proportional to the distance between nodes, and low-pass filtering can be accomplished by steering a node away from a source.

However, certain aspects of our control paradigm *deliberately* violate the rules of physics. The fact that audio can be tightly focused and travel only along a narrow pathway is obviously unnatural. Furthermore, for the purpose of musical creation and performance, users may not desire correct models for natural sound propagation. Rather, they may wish to exaggerate or diminish certain acoustic properties for artistic purposes. For example, Doppler shift is often emphasized in sound tracks for cinema because it adds dramatic effect. Also, to conserve timing and intonation in musical pieces, it may be desirable to eliminate distance-based delay of sound. Allowing for such rule-bending is a useful feature of our system.

2.3. Sound Spaces & Acoustics

In nature, the size, shape, and the types of objects within an environment will affect the propagation of sound. Our brains are remarkably adept at uncovering this hidden information and inferring properties about our surroundings. Users will expect these acoustic cues when they travel through virtual spaces, hence we provide a mechanism for defining enclosed areas within a scene. These *soundSpaces* are similar to soundNodes, yet rather than capturing and emitting audio at a fixed point in space, they operate on a volume defined by an arbitrarily shaped 3-D model (defined in 3D Studio Max, Maya, Blender, etc.). Signals from soundNodes within the volume will be captured with unity gain, while signals from nodes outside the volume will be captured and attenuated according to a soundNode's distance from the boundary and the absorption coefficient of the volume's surface.

The reverberation model for these soundSpaces allows for the specification of delay times (for direct sound, and 1st & 2nd order reflections), a reverberation time, and a filter to simulate frequency-dependent damping over time. However, if the user wishes, this model can be replaced with any type of DSP unit such as a flanger or harmonizer. Thus, rather than walking into an enclosed space and hearing the reverbaration of your voice or instrument, you would hear a flanged or harmonized version of your sound signal instead. This offers many interesting possibilities for artistic creation. For example, a virtual scene can be subdivided into various zones, each of which correspond to a different movement in a musical score. The progression of a work is thus achieved via spatial movement of performers in the virtual scene during performance.

3. USER-SPECIFIC AUDIO RENDERING

In the context of audio for virtual environments, the general goal is to present sounds to a listener so that they may localize the sources in 3-D space. This problem, known as 'audio spatialization', can be solved in many ways depending on the audio display being used (i.e. the number of loudspeakers available and their positions relative to the listener).

Binaural methods, for example, aim to reproduce the interaural timing and intensity differences heard by human ears. By filtering a signal with an HRTF, signals for each ear can be obtained that simulate sound sources originating from arbitrary directions [4]. This approach is typically used if a headphone-based display is being used. If instead the scene is rendered on a loudspeaker array, amplitude (and sometimes timing) differences between loudspeakers can be used to affect the apparent location of a sound source. Ambisonics [5] for example, encodes audio signals with directional components (x, y, z, w), where w is the gain in the direction specified by the vector (x, y, z). Decoders exist that

[1] Note that $\gamma = 1.0$ produces a normal cardioid, $\gamma = 0$ flattens the shape resulting in omni-directional radiation, and $\gamma > 1.0$ results in a hyper-cardioid. This is therefore useful to a performer who wishes to change from omni-directional to tightly-focused with just one parameter.

reproduce the appropriate loudspeaker signals by taking a linear combination of these four channels. Vector base amplitude panning (VBAP) [6] is a similar method, in which the speakers are grouped into triplets and each sound source is rendered using only three loudspeakers rather than the whole array.

Both ambisonics and VBAP are meant to be deployed on loudspeaker arrangments that are uniformly spaced around the user, providing accurate spatialization at the convergence point, or *sweet spot*. Wave Field Synthesis (WFS) on the other hand attempts to create a sound field throughout an entire volumetric space. This requires a large array of small speakers and is computationally expensive compared with the aforementioned techniques. The concept is based on Huygen's principle, which states that a wavefront can be seen as the composition of elementary waves, each of which propagate with the same velocity and wavelength as the original wave. In WFS, each loudspeaker is responsible for generating one of these elementary waves, which combines with others to re-create the true wavefront of a particular sound experience.

There have been a few implementations that render sound at arbitrary spatial positions using a virtual microphone approach [7, 8]. Mainly based in audio engineering, these implementations tend to focus on acoustic space simulation and typically use prerecorded material with an acoustical model of the recording space to resynthesize the signal heard at any given position. We are however unaware of any projects that use this technique for real-time sound rendering in arbitrary virtual environments.

Our method of audio display differs from the above methods, taking advantage of our dual-purpose sound nodes, and their customizable diffusion patterns. A listener is simply represented by a collection of sink nodes, corresponding to the real-world loudspeaker array that renders their sound. Each sink can be thought of as a virtual microphone that collects sound from a specific direction. For example, a 5.1 channel audio display uses an array of six virtual microphones, where each microphone corresponds to a particular loudspeaker in the listeners physical space. The centre microphone would have 0° rotation, collecting sound directly in front of the user. The right microhone would be oriented 30° to the right, pointing to an appropriately positioned real-world loudspeaker. The right rear microphone would have 110° rotation, and so on. Each microphone then buffers its collected signal to the appropriate output channel of a soundcard so that it may be heard correctly. Furthermore, each virtual microphones has an appropriately defined rolloff function that allows for an equal energy distribution when a source pans across adjacent microphones.

With this approach, multiple listeners with their own 'private' audio rendering can occupy the environment at the same time. All that needs to be done is to define additional virtual microphones for each participant, and the audio scene can be simultaneously rendered at several locations. Furthermore, these microphone arrays can be re-arranged on the fly. A user can begin exploring the environment with headphones and switch to a surround speaker display by simply swapping one array with another.

4. APPLICATIONS

The framework that we have described provides a novel approach to managing audio in virtual environments. There are many applications that can be developed with these features in mind. We have begun to explore the following applications, and plan to develop these further in the future.

4.1. Next-generation Video Conferencing

One of the most exciting application domains of our framework is the state-of-the-art in videoconferencing environments. In an increasingly connected world, individuals are communicating over greater distances and are looking for technological tools to facilitate the process. The current state of most videoconferencing technology is, however, still impoverished compared to real-life communication. Even ignoring problems of latency and the low resolution of audio and video associated with consumer-grade systems, support for anything other than one-to-one conversation is poor. As a result of these defficiencies, interaction is typically limited to an unnatural 'turn-taking' type of conversation, with no affordances for sidebar or private interactions between distributed participants.

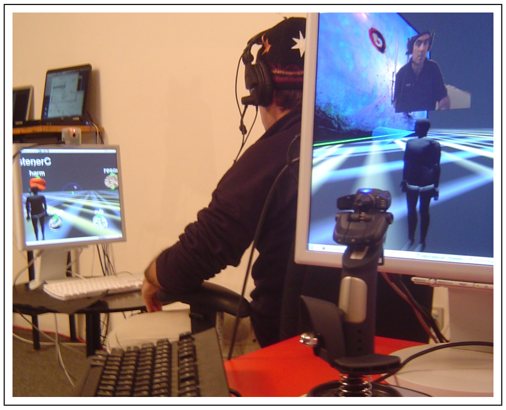

Figure 3: A prototypical deployment, where a remote user is seen as a virtual avatar with a video stream above his head. Though difficult to see in this figure, the avatar has a diffusion cone emanating from his head that indicates the direction of the emitting sound.

In contrast, our framework provides the ability for users to steer audio or focus listening in specific directions. In a videoconferencing scenario, this allows some users to engage in private discourse (sidebar conversation) while others in the vicinity are unaware of what is being discussed. We regularly employ private discourse in day-to-day interactions, and this is perhaps essential in negotiation activities or other business transactions. Somewhat surprisingly, this feature is absent from all existing videoconferencing systems.

4.2. 3-D Audio Mixing

Unlike conventional interfaces for mixing, which usually involve sliders and panning sounds in 2-D with some type of joystick interface, our interface intrinsically provides the ability to mix-down recordings using 3-D spatial arrangement. The format of the mix is achieved by the spatial organization of source sounds (tracks), and the placement of virtual microphones among them.

Virtual microphone arrays can be swapped in a particular scene to create mixdowns for different types of media. For example, a 5.1 channel mix can easily be changed to a stereo mix by simply switching from an array with six virtual mics to an array with two. It is also worth noting that unlike conventional surround mixing (pantophonic), which localizes sounds on a horizontal plane, our virtual microphones capture sound based on a 3-D angle of incidence, offering fully three dimensional (periphonic) audio reproduction. Finally, as mentioned earlier, a virtual microphone array can be itinerant, and move dynamically though the playback space during mix-down, thus opening up additional artistic possibilities to the mixing engineer.

4.3. Musical Performance

This architecture also allows for many interesting musical scenarios, where musicians can gather and share an interactive virtual space. There are obviously possibilities for remote performance where users can jam together, yet the fact that performers can share resources in the virtual scene, such as effects processing nodes, significantly extends notions of *collaboration*. Most remote performance systems (e.g. AccessGrid [9]) simply allow musicians to play together, exchanging remote audio signals. There is little influence that one user can have over another's production of sound. Steerable audio provides a novel and essential mechanism that enables the use of virtual space as an interface and medium for musical collaboration and interaction.

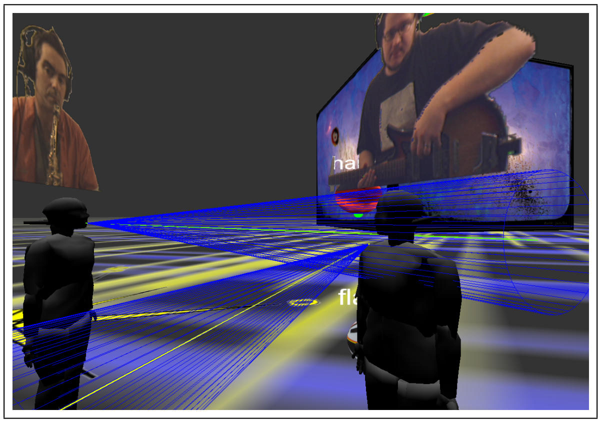

Figure 4: A screenshot showing two performers sharing a virtual environment.

Our framework provides for the positioning of effects processing units in space that may be controlled by any participant. A performer can thus apply harmonizers, flangers, delays, or any other processing to their sound simply by steering it towards the appropriate soundNode in the virtual world. Meanwhile, other users in the space can reposition or rotate various soundNodes. This means that the audio scene can change while performers are steering their sounds through it. The interaction is thus highly collaborative, where individuals may work together to modify scene and create musical material in real time.

Furthermore, users are free to move to different parts of the world, entering and exiting various soundSpaces, and interacting with different sets of sound processing objects. It is even possible to have several simultaneous jam sessions in the same virtual environment, each occuring at a different location in the space. Audience members are free to navigate around the world, visiting different venues where other participants perform. Thus, a user with a suitable network connection and hardware configuration (i.e. a computer, stereo headset, and a joystick/mouse) may experience live 3-D audio performances from the comfort of their own home. In effect, the environment becomes one large, distributed, multi-user performance venue where a number of possible hats can be worn.

5. CONCLUSION

We present a system that allows multiple participants, each with user-specific 3-D audio rendering, to share a common virtual world. Users can be distributed geographically, leading to interesting applications in the domains of videoconferencing, distributed performance, and remote collaborative interaction. The use of a spatial audio steering interface lends quite well to implementations in virtual spaces. This approach has already shown great promise when applied to conventional applications such as audio mixing. However, we are particularly interested in its potential for uses in artistic applications where new forms and modes of interaction can be explored.

6. ACKNOWLEDGEMENTS

The authors wish to acknowledge the generous support of NSERC and Canada Council for the Arts, which have funded the research and artistic development described in this paper through their New Media Initiative. In addition, the authors are most grateful for the resources provided to the project by McGill University's Centre For Intelligent Machines, La Société Des Arts Technologiques, and The Banff Centre for the Arts. These research centres have provided productive and stimulating environments allowing for the development and prototyping of this project.

7. REFERENCES

[1] Mike Wozniewski, Zack Settel, and Jeremy R. Cooperstock, "A framework for immersive spatial audio performance," in *New Interfaces for Musical Expression (NIME), Paris*, 2006, pp. 144–149.

[2] Mike Wozniewski, Zack Settel, and Jeremy R. Cooperstock, "A paradigm for physical interaction with sound in 3-D audio space," in *Proceedings of International Computer Music Conference (ICMC)*, 2006.

[3] Durand R. Begault, *3-D sound for virtual reality and multimedia*, Academic Press Professional Inc., 1994.

[4] W. G. Gardner and K. D. Martin, "HRTF measurements of a KEMAR," *J. Acoustics Soc. of America*, vol. 97, no. 6, pp. 3907–3908, 1995.

[5] M.A. Gerzon, "Periphony: With-height sound reproduction," *J. Audio Eng. Soc.*, vol. 21, no. 1, pp. 2–10, 1973.

[6] Ville Pulkki, "Virtual sound source positioning using vector base amplitude panning," *J. Audio Eng. Soc.*, vol. 45, no. 6, pp. 456–466, 1997.

[7] Athanasios Mouchtaris, Shrikanth S. Narayanan, and Chris Kyriakakis, "Virtual microphones for multichannel audio resynthesis," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 968–979, 2003.

[8] Jonas Braasch, "A loudspeaker-based 3D sound projection using Virtual Microphone Control (ViMiC)," *Convention of the Audio Eng. Soc. 118*, pp. 968–979, 2005.

[9] "AccessGrid," www.accessgrid.org.