A SPATIAL INTERFACE FOR AUDIO AND MUSIC PRODUCTION

Mike Wozniewski, Zack Settel
La Société Des Arts Technologiques
Montréal, Québec, Canada
{mikewoz|zack}@sat.qc.ca

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

ABSTRACT

In an effort to find a better suited interface for musical performance, a novel approach has been discovered and developed. At the heart of this approach is the concept of physical interaction with sound in space, where sound processing occurs at various 3-D locations and sending sound signals from one area to another is based on physical models of sound propagation. The control is based on a gestural vocabulary that is familiar to users, involving natural spatial interaction such as translating, rotating, and pointing in 3-D. This research presents a framework to deal with real-time control of 3-D audio, and describes how to construct audio scenes to accomplish various musical tasks. The generality and effectiveness of this approach has enabled us to re-implement several conventional applications, with the benefit of a substantially more powerful interface, and has further led to the conceptualization of several novel applications.

1. INTRODUCTION

Even in mainstream applications, where MIDI keyboards provide control of digital synthesis and effects can be added in real-time, the interface between performer and computer remains a limiting factor. In experimental applications, where a performer playing on stage may want to accomplish more and more with the aid of computer technology, conventional audio techniques fail. Though the equipment (audio mixers, synthesizers, effects units) can perform the required signal processing, interaction with it is very poorly adapted to the performing musician who wishes to control multiple effects and processors while playing other instruments. Throwing additional knobs, pedals, buttons, or sliders at this problem only worsens the situation because the task is at variance with the interface. In our approach, we abandon conventional audio control devices, and focus rather on the natural gestures and movements a performer might make.

We seek to capitalize on the great understanding of physical phenomena that humans acquire when dealing with the natural environment. For example, kinesthetic feedback from muscles will inform a person about the locations of their limbs, and the innate understanding of sound propagation allows them to localize sounds and infer various spatial characteristics of the world. We propose an paradigm for interaction with sound that utilizes this understanding. In previous work [1], we have described a framework for immersive spatial audio performance, where a user's bodies are modelled within a virtual 3-D world, and the propagation of audio is computed based on acoustic physical modeling. This framework is among only a few (e.g. [2, 3]) that have explored virtual environments from the perspective of music or digital signal processing (DSP). Ultimately we have created a tool with which artists can create interactive virtual scenes that are controlled by regular spatial activity such as moving, turning, pointing, or grabbing. Thus, for example, a listener's experience of music in this environment will change as he/she moves or turns.

It is also possible to augment the perceptually accurate model of sound propagation to achieve results that are interesting for artistic or musical purposes. For example, effects of Distance or Doppler shift can be diminished or exaggerated and sound can be radiated or captured with a very narrow focus so that users can choose exactly how sound travels within their scenes.

Users can thus construct a *musical situation* (e.g. performing, listening, both) that takes the form of realistic scene, with similar geometry and behaviour to that of the real world. The ability to control signal processing with natural body motion and modified physical models obviously results in new metaphors for DSP design and the potential for novel sonic applications. In the following discussion, we will review our framework and discuss how to construct various audio scenes. Several sample applications are provided that illustrate the novel uses of this paradigm.

2. THE SPATIAL AUDIO FRAMEWORK

At the heart of our research is something we call the *soundEngine*, which handles the computation of virtual audio. This is a 3-D engine built using OpenSceneGraph to manage the arrangement of objects in space, and efficiently control various geometric transformations performed on the objects. PureData is used as the front-end interface and a method to manage signal processing and audio hardware. The engine allows for the dynamic creation, modification, and control of custom-made virtual audio scenes.

An audio scene for that matter, is composed of sound processing entities called *soundNodes*, which exist at some 3-D location and have various parameters to aid in DSP computation. A *soundConnection* is made between pairs of soundNodes, defining how sound propagates through the space between them. These soundNodes and soundConnections can respectively be thought of as the nodes and edges of a *DSP graph*, which is a convenient way to visualize the processing chain. We will expand on these topics in the following sections, though readers requiring further details can refer to our previous work [1].

2.1. The soundNode

The soundNode is the fundamental building block of a virtual audio scene, and can be either a *source* (which emits sound), a *sink* (which absorbs sound), or *both* of these at the same time. The case where a soundNode represents both is particularly important in the context of musical interaction since this is how signal processing is realized.

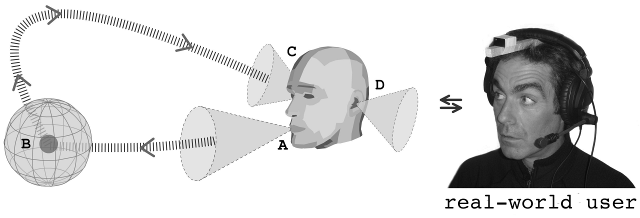

Figure 1: *Example of how to model a simple scene with soundNodes.*

This type of signal processing is illustrated in Figure 1. A source node A emits sound into the virtual world. Node B absorbs some of this sound, applies DSP, and emits the signal back into the scene, thus acting as both a sink and source. Nodes C and D are both simple sink nodes which collect the resulting signal and pass it to loudspeakers in the real world.

We note that source nodes in our representation are quite similar to those found in many other sound spatialization systems. They typically represent input from microphones, sound files, or sound synthesis. The concept of sink nodes is however a little more unique. Most interactive systems only render the audio scene for one listener at one position in space, called the "sweet spot". There has been some recent work to synthesize sound at arbitrary spatial positions using virtual microphones [4, 5], yet this tends to focus on resynthesis of pre-recorded materials and not real-time performance. In our representation, a listener is just a particular type of sink node (or group of sinks) at a given location in space, where the audio absorbed is written to hardware buffers. Sink nodes are however used for many other purposes as well. Most importantly, they allow for the spatially localized DSP processing described above, but in addition, they allow for the simultaneous rendering of an audio scene at various locations. This can be used to produce surround recordings, or for multiple listeners who wish to occupy the environment at the same time.

Each soundNode entity also has other interesting parameters that users might wish to control. It can be translated/rotated in space, and the directivity (or roll-off) can be adjusted in real time. The roll-off is a parametric function that describes the attenuation that should be applied to a signal at different angles of propagation. Arbitrary functions can be defined, including those that are commonly found in acoustics and sound recording equipment (e.g., cardioids). However, the ability to focus sound with a tighter or wider directivity can be far more interesting than those common cases. Such ability provides the user with highly accurate control over a particular sound signal, and where it can travel. In the end, users can create focused "beams" of sound that they can aim at various soundNodes, which become virtual effects units that process audio at some location in space. A user can thus mix among them by focusing sound in different directions.

2.2. The soundConnection

In our representation, each pair of soundNodes that can exchange audio, must have a *soundConnection* established between them. This is different to many conventional systems, where every sound source is rendered in the same fashion. The reason for this is that we are viewing the propagation of audio as a type of signal bus, similar perhaps to a patch cord that one would use when connecting audio equipment to accomplish various DSP tasks. Yet in addition to the logical connection made between the nodes, the soundConnection also describes *how* the sound should travel through the intermediate space. For example, whether it should be absorbed/scattered by air, delayed in time, have it's intensity decay with distance, be filtered based on angular incidence, or obey Doppler shift.

Once a connection is established, a user can specify the intensity and character of each of these propagation features, allowing for various interesting musical effects. For example, a user preparing a scene rich with harmonies might like to prevent frequency shifting and preserve the tonal content of the music. This can easily be accomplished by removing the effects of distance decay and Doppler from a connection, which effectively "teleports" a sound signal between two distant nodes.

One important result of the soundConnection paradigm is that mixing is not controlled with sliders and knobs, but is instead determined by the spatial position and orientation of the soundNodes involved. Users are free to move and manipulate these in real-time using a variety of sensors and tracking mechanisms. When the nodes are far apart, it means that the signal will be attenuated more and that there will be a greater delay. By bringing nodes closer together, we will increase the gain just like a slider on a mixing board would do, but with a much more natural conceptual model. Likewise, by orienting a node directly towards a source, we would imagine that the full spectrum of sound gets propagated. Then as the source rotates, pointing further away from the source, the incidence filter will attenuate higher frequencies, thus simulating the diffraction of sound waves.

2.3. Audio Scene Description

The soundConnections allow a user to organize soundNodes into groups which accomplish a specific DSP task. These groups can be visualized as *DSP graphs*, where soundNodes correspond to the nodes of a graph and soundConnections form the edges. These graphs are directed, always starting from a source node and terminating at one or more sink nodes. They are also potentially cyclic, meaning that recursive filtering can easily be developed.

We note that a single soundNode only has the ability to provide monophonic signal processing. Polyphony is achieved by interconnecting several soundNodes, and copying signals between them. Mixing and real-time control over the DSP graph is accomplished by adjusting the parameters of all the various soundConnections and soundNodes involved. This may seem like a difficult task as graphs get larger and larger, yet recall that parameter adjustment is accomplished by regular spatial activity such as rotating and translating. Furthermore, we employ another graph structure to help propagate spatial updates to many nodes at once.

In addition to organizing soundNodes in a DSP graph, those same nodes are also held in a *scene graph*, which is a concept borrowed from the field of 3-D graphics. A scene graph is a tree-structured graph where spatial transformations applied to a node are automatically propagated to all children. Thus geometric relationships between soundNodes can easily be described. For example, if we wanted several nodes to move together then we would attach them all to some parent node that would act as a frame of reference. Whenever the parent is moved, all the children would move accordingly. If the parent rotates, then the children also rotate, but relative to the coordinate system of the parent.

The use of scene graphs can be illustrated with Figure 1 that

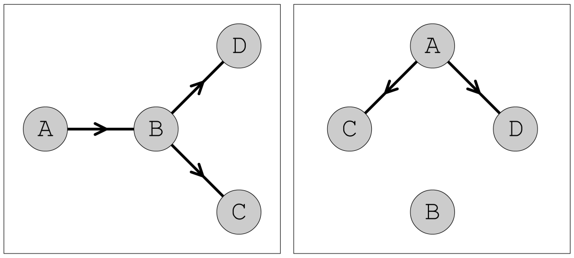

Figure 2: *The DSP graph (left) and scene graph (right) associated with Figure 1.*

we saw earlier. We see that nodes A, C, and D all share a common geometric reference. If the user's head moves or rotates, then all three of these nodes should move and rotate relative to the central head position. Figure 2 shows both the DSP graph and scene graph associated with the simple example presented in Figure 1. If the user rotates his head to speak in a different direction, the location of the sink nodes representing his ears ears will rotate accordingly. However, node B is not affected since it has no geometric relation to the user's head. Readers should understand that scene graphs have no bearing on the DSP chain. They are simply structures that organize the geometry of the scene.

3. APPLICATIONS

Our ability to simulate not only sound sources, but sound sinks, opens the door to a host of novel applications for directing and manipulating sound in 3-D. The flexibility of our propagation model provides us with a point of departure *away* from perceptually accurate models of 3-D audio. We may exaggerate or diminish certain acoustic properties to create novel experiences of sound / music for practical or artistic purposes. Of course, our approach lends well to creating applications that involve 3-D content. However, an obvious appeal of working with our framework is that some tasks that were difficult or impossible to do in a conventional audio production environment become easily accomplished in ours, allowing us to revisit conventional applications and to discover new ones. Below we describe several applications that have been built with our framework, and the new paradigms of interaction with sound that we have discovered along the way.

3.1. Active Listening

Since our framework intrinsically provides a model for "hearing", possibilities for *listening* are quite abundant. In the real world, sounds arrive at our ears from all directions, and it is up to the human perceptual system to differentiate and localize them. We are extremely good at this task, even being able to isolate a single conversation in a room full of people. However, the fact that our soundEngine framework allows us to bend the laws of physics means that we can narrowly "focus" hearing in a particular direction when needed: any sound that is not directly ahead would be severely attenuated. Additionally, the user could diminish the effect of distance attenuation so that distant sounds are heard more clearly. This ability to finely direct one's listening can lead to various novel applications, which we refer to these as *Active Listening* applications since they are based on the notion of user-directed listening. It is interesting to note that, given a high level of active listening in a musically rich field of sound sources, notions of authorship and creation can enter into the picture when considering the listener's particular experience of the given soundscape: who is the composer?

Our active listening application provides a user with the ability to navigate within a space containing source-type soundNodes, each one radiating one track of a multi-track recording of an African percussion ensemble. As the listener moves and turns in the space, his/her musical experience constantly changes based on the proximity and orientation to the various soundNodes. By "rolling" his/her head the listening focus may be narrowed or widened, thus adding an additional element of listening control. Given the polyrhythmic nature of the music, the range of potential listener experiences is quite large. When listeners' sessions are recorded, great differences can be noted among the recordings, pointing to a potential authorship tool for creating remixes.

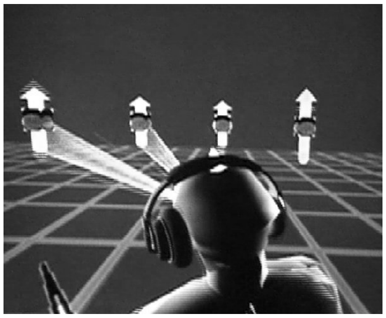

Figure 3: *A screenshot from the Active Listening application. A graphical avatar moves and rotates according to a sensor worn by the user, thus controlling two tightly-focused sound sinks which can be aimed at various sources in 3-D space. The arrows attached to each loop indicate sound level, something like a VU meter.*

3.2. Multichannel Audio Displays & Mix-downs

The ability to locate listeners in organized "fields" of sound sources lends extremely well to applications for audio mixing (mix-down). Unlike traditional interfaces for audio mixing, which usually involve panning sounds in 2D with some type of joystick interface, our interface intrinsically provides for mix-down using 3D spatial arrangement. In this case, the listener can be thought of as a *virtual microphone*, capturing sound sources in its proximity, and spooling them to disk. Thus, the quality of the mix is achieved by the organization of source sounds (tracks), and the placement of virtual microphones among them.

The particular format of the mix (5.1, stereo, etc.) is determined by the number of virtual microphones. For example, a 5.1 mix-down uses an array of six virtual microphones that collect the audio scene at some location in space. These microphones (sink soundNodes) are oriented to capture sound in their respective directions (e.g. the centre microphone would have 0° rotation, the right microhone would have 30°, the right rear would have 110°, etc.). It is worth noting that unlike conventional surround mixing environments, which localize sound in a horizontal plane (pantophonic), our environment's virtual microphones filter sounds based on their angle of incidence, offering fully 3D (periphonic) audio reproduction. Additionally, a given virtual microphone array can be itinerant, and move dynamically though the "recording space" during mix-down, thus opening up additional artistic possibilities to the mixing engineer.

3.3. 3D Audio Dipping

The technique of creating music using samples and loops of existing material has become quite popular. It usually involves gear such as turntables, samplers, and sequencers that are triggered using keys, knobs, or sliders. This control paradigm is quite effective, allowing a single performer to produce very intricate musical material. The user simply ensures that all of the devices are synchronized (perhaps by some external clock signal or by simply ensuring that all material has the same tempo). Then using sliders for example, he or she can fade in one or more of the signals.

This style of control can also apply to material generated by computer music processes or synthesizers. Wessel & Wright [6] have defined this technique as “Dipping”, where the musical processes are silent by default, but the performer can “dip” in with some controller to make the processes heard.

Similar to Active Listening, the 3D Dipping application is particularly successful deployed in our framework. We start by placing source soundNodes around the user, ideally so that each node is equidistant. The sound contained in the nodes could be a number of phase-locked loops to ensure synchronization. The performer then focuses a sink soundNode in the direction of the sound he/she wishes to dip into (capture). For example, a 6 DOF sensor, mapped to the position and orientation of a sink node could be coupled to each of the performers hands. Using an additional parameter such as the roll (or twist) of the hand, the user could also tighten/spread the focus so that fewer/more nodes can be dipped into.

Additionally, by performing an editing gesture (via glove controller, modal button, etc.), the performer could relocate and swap soundNodes during performance, thus extending musical range. Such ability could prove quite useful, since the performer could group certain nodes together in space so that they are always activated together. Nodes could also be moved closer or further away which effectively controls the gain, since their emitted sound decays with distance.

3.4. Spatial Routing to DSP Effects

Effects routing is well-known in live electro-acoustic performance circles, and particularly well-suited to re-implementation in our framework. Unlike the above applications, this one focuses on the performer as a modeled human source node surrounded by many sink nodes present in the scene that collect sound from him/her. To make them easy to target, sink soundNodes can be placed in a configuration which corresponds to locations of real objects in the performance space, such as a corner, clock, pillar, person etc.. Thus, we may equip a saxophone with an orientation/position sensor and a wireless microphone. We then define a certain number of sink nodes, each with a particular effects processing unit, such as reverb, delay, flanging, ring modulation etc.. When the sax player points the instrument in a certain direction, its sound is proportionally radiated to one or more effects nodes in space according to the propagation model between them. By “rolling” the horn, its radiation is altered, fanning out to reach more effects nodes, or the inverse. Of particular interest to performers of live electronics, *dangerous* dsp, such as recursive flangers, can be included among the effects nodes, since feedback can only occur when the input is aimed directly at it. In fact, the incredibly high degree of control that the performer has in sending his/her signal to a node allows for riding an extremely fine edge between stable and unstable DSP. Live sound technicians are particularly appreciative of this capability.

4. CONCLUSIONS

The organization of DSP processors in 3-D space and using physically modelled sound propagation as a signal bus results in a new approach to musical applications. We have described a framework for controlling 3-D audio by creating scenes composed of soundNodes and soundConnections. These scenes can further be analyzed in terms of their DSP graphs and scene graphs, which provide convenient tools for conceptualizing and manipulating parameters. We have shown how several conventional and new applications can be conceived using this paradigm, and how novel approaches can be taken to provide existing tasks with more natural control. In the end, we believe that the conceptual model provided to the user will lighten the cognitive load required to deal with complicated musical situations. The fact that we build on the existing knowledge that humans have of spatial relationships gives us reason to believe that this interaction paradigm has great potential.

5. 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 La Société Des Arts Technologiques and to the Banff Centre for the Arts for offering a productive and stimulating environment in which several of the applications described here were prototyped.

6. REFERENCES

[1] M. Wozniewski, Z. Settel, and J. Cooperstock, “A framework for immersive spatial audio performance,” in *Int. Conf. New Interf. for Musical Expr. (NIME’06),* Paris, France, 2006, pp. 11–16.

[2] J. Pressing, “Some perspectives on performed sound and music in virtual environments,” *Presence*, vol. 6, no. 4, pp. 482–503, 1997.

[3] T. Maki-Patola, J. Laitinen, A. Kanerva, and T. Takala, “Experiments with virtual reality instruments,” in *Int. Conf. New Interf. for Musical Expr. (NIME’05),* Vancouver, Canada, 2005, pp. 11–16.

[4] J. Braasch, “A loudspeaker-based 3d sound projection using virtual microphone control (ViMiC),” in *118th Conv. Audio Eng. Soc.,* Barcelona, Spain, 2005, pp. 968–979.

[5] A. Mouchtaris, S. S. Narayanan, and C. Kyriakakis, “Virtual microphones for multichannel audio resynthesis,” *EURASIP J. on Applied Sig. Proc.*, vol. 2003, pp. 968–979, 2003.

[6] D. Wessel and M. Wright, “Problems and prospects for intimate musical control of computers,” *Computer Music J.*, vol. 26, no. 3, pp. 11–22, 2002. [Online]. Available: citeseer.ist.psu.edu/wessel00problems.html