A framework for immersive spatial audio performance

Mike Wozniewski
McGill University
Centre for Intelligent Machines
3480 University Street
Montreal, Quebec
mikewoz@cim.mcgill.ca

Zack Settel
La Société Des Arts Technologiques
1195 Saint-Laurent boulevard
Montreal, Quebec
zack@sat.qc.ca

Jeremy R. Cooperstock
McGill University
Centre for Intelligent Machines
3480 University Street
Montreal, Quebec
jer@cim.mcgill.ca

ABSTRACT

Traditional uses of virtual audio environments tend to focus on perceptually accurate acoustic representations. Though spatialization of sound sources is important, it is necessary to leverage control of the sonic representation when considering musical applications. The proposed framework allows for the creation of perceptually immersive scenes that function as musical instruments. Loudspeakers and microphones are modeled within the scene along with the listener/performer, creating a navigable 3D sonic space where sound sources and sinks process audio according to user-defined spatial mappings.

Keywords

Control paradigms, 3D audio, spatialization, immersive audio environments, auditory display, acoustic modeling, spatial interfaces, virtual instrument design

1. INTRODUCTION

Virtual environments (VEs) have received considerable interest from researchers in fields as diverse as scientific visualization, cognitive perception, and medical therapy. Despite continued improvements in the quality of VE audio components, there has been comparatively little use of this technology in the areas of musical performance and interactive sonic display. This is perhaps due to the simplistic and impoverished audio representations, geared largely toward gaming applications on consumer-grade systems, that dominate the field. In these simplistic models, virtual sound sources are associated with fixed models of propagation, with audio rendered identically for every source. The demands of musical composition and performance, however, require far greater flexibility of control over sound propagation and mapping strategies. Musicians are typically less interested in having perceptually accurate acoustic models, and would rather have the power to manipulate or exaggerate those affects for artistic purposes. The research framework described here is among only a few (eg. [11, 12, 14]) that have explored VEs from the perspective of musical performance. It allows for the control of various features on the basis of individual sound sources and sinks, thereby facilitating the design of rich 3D audiovisual scenes that function as virtual musical instruments.

The framework allows a user to create a navigable 3D scene where his position and orientation are modeled in space, as well as the positions of the loudspeakers with which he will be rendering the scene. This allows the user to control the progression of a musical piece by traveling through virtual space, and to switch easily between different auditory display technologies. The content of the scene is composed of spatially positioned sound sources and sinks that process audio according to user-definable DSP routines. Any form of input device can be used and easily mapped to affect DSP or visualization parameters. Moreover, sonigenic mappings can be employed to visualize the sonic properties of various scene elements. The user's ability to control how and where audio propagates within the scene suggests a powerful control paradigm, building on the human strength of spatial understanding.

2. AUDIO IN VIRTUAL ENVIRONMENTS

By incorporating diegetic[1] audio into a VE, we provide users with additional information for localizing objects and events in the scene. This allows them to perform more parallel and simultaneous tasks, since the cognitive load imposed on the user is lessened. Furthermore, the psychological immersion that the user perceives parallels that of the real world, hopefully leading to a more pleasant experience.

2.1 Auditory Display Technologies

Research in the area of virtual audio environments is mainly concerned with the challenge of perceptually localizing sounds in 3D space. This is known alternatively as 'audio spatialization', 'audio rendering', or 'sound imaging'. Many techniques exist (see [4, 10, 13, 18] for an overview), but all these methods can be grouped into three main classes: binaural methods, amplitude differencing, and wave field synthesis (WFS).

Binaural methods use filters to simulate frequency-based modifications to the audio signals based on a head response transfer function (HRTF). The filter coefficients for sounds originating at various coordinates are acquired by empirically measuring the audio responses on a model human head [7]. This way, one can spatialize a sound by filtering two audio signals and rendering them through headphones, or through a stereo speaker setup. Headphones are obviously the best display method, since the audio signals are isolated to each ear and crosstalk cannot not occur. There are many methods for crosstalk cancellation [6], but the user is required to stay motionless, otherwise the perceptual effect is ruined.

The consumer-grade systems that many gamers and cinemaphiles have deployed in their homes use several speakers placed equidis-

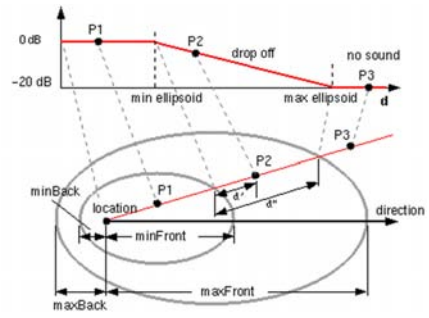
[1]The term *diegetic sound* is common in the film industry and describes a sound that is presumed to be present and localized in the scene. If a character in a movie plays an instrument, this is diegetic music, whereas the omni-present background music (that the character is unaware of) is non-diegetic.

tant from the one central point, often called the *sweet spot*. These systems typically use amplitude differences between loudspeakers to make sounds appear from a certain direction. The commercial-grade systems are however usually pantophonic (localize sounds only in a horizontal plane), and require very precise speaker arrangements. Some work has made periphonic (fully 3D) methods possible. Ambisonics [8] for example, encodes audio signals with directional components (x,y,z,w), and decoders exist that reproduce the necessary sound signals on multiple speakers to simulate sounds arriving at the user with apparent direction. Vector base amplitude panning (VBAP) [15], groups speakers into triplets, and a direction vector can be computed for any virtual sound source located within the loudspeaker triangle. Both ambisonics and VBAP support rendering of 3D sound with variable numbers of speakers, though ideally the speakers should be uniformly distributed and equidistant from the user to minimize error.

Contrary to amplitude panning techniques, where the sound field is accurately produced only in the sweet spot, WFS attempts to recreate the sound field throughout an entire volumetric space. This requires an array of many speakers, and much computational power, in order to compute the propagation of each boundary signal. The sound field may not be completely accurate in any one spot, but the error field is minimized and uniformly distributed, allowing the listener to move about the space without loss of perceptual auditory immersion.

2.2 3D Audio Scene Description

In the case of real-time interactive systems such as games and virtual reality systems, there have been many attempts at modeling virtual sound sources. Several toolkits available for developing such applications, including Microsoft DirectX, OpenAL, and X3D. Creative EAX can be added to these to simulate room reverberation effects complete with occlusions, exclusions, and obstructions. These toolkits allow developers to move sounds sources interactively around a virtual scene and have them properly rendered. However, support only exists for standard surround formats, so arbitrary speaker configurations for ambisonic or VBAP rendering are not yet possible.



1: The X3D sound node model [1].

However, the most limiting factor of these toolkits is the fact that they use very simplistic spatial models of audio sources. Most APIs have no method to specify directivity of sounds (i.e. all sounds are considered omni-directional), usually a maximum of one listener is supported, and very little power is provided to manage more complex sonic interactions between sound nodes. In fact, the only API that supports directivity of sound sources is X3D, which models directivity with two ellipsoids (as shown in Figure 1). The inner ellipsoid indicates the commencement of attenuation based on distance, which proceeds linearly until full attenuation occurs at the outer ellipsoid.

The problems with this method are abundant. First, there is no method to represent more complicated directivity patterns, such as that of traditional musical instruments. Similarly, directivity applies only to sound sources. Sound sinks (i.e. listeners) cannot be modeled with directional sensitivity, thus cardioid patterns of microphones cannot be modeled. Finally, the linear attenuation model does not accurately relate to real-world physics, where sound in fact decays exponentially.

There have been a few projects that have gone to much further extents in modeling sound source propagation for interactive systems. For example, the DIVA (Digital Interactive Virtual Acoustics) project [17] uses ray casting (similar to methods in 3D graphics) to compute reflection paths of audio signals. Additional virtual sources are created at the reflection points, and by modeling surfaces with absorption parameters, they achieve a much more realistic response of the environment. Tsingos et al [19] augment this technique, by also modeling the diffraction of sound around the edges of 3D objects using the uniform theory of diffraction (UTD). These methods still however simplify the directionality of sound sources in order to make their more complicated rendering approaches tractable in real-time.

3. A SPATIAL AUDIO FRAMEWORK FOR MUSIC & SOUND APPLICATIONS

Although the traditional methods presented earlier do offer control mechanisms for interactivity, these often pertain to spatial rather than sonic arrangement. In traditional methods, sounds can be moved individually in real-time, but their acoustic parameters (directivity pattern, roll-off, etc.) cannot be manipulated dynamically. Rather, these parameters remain fixed and are usually defined for the entire scene rather than an individual node. For example, it is not possible to include some sources in the computation of reverberation, while excluding others. Furthermore, the virtual sound sources tend to map to external audio streams (sound files, networked audio, etc.) rather than signal processing objects that can interact with other nodes according to their spatial arrangement.

For the purpose of musical creation and performance, perceptually accurate models of sound propagation may not always be desired. The user (or rather, musician) may desire to exaggerate certain acoustic effects for artistic purposes, and use the 3D propagation of audio as a medium for musical expression. The audio scene representation that we propose allows for this level of control by making *sound nodes* much more generic entities, and explicitly requiring a *sound connection* to be made between nodes. With proper management of these nodes and connections, one can create a navigable 3D space, that functions as a control interface for an immersive virtual instrument.

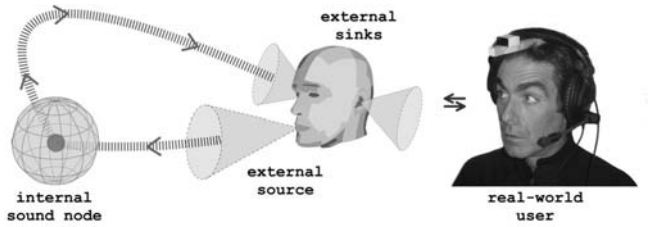
3.1 The Sound Node

Sound nodes in our representation can be either *sources* (emitting sound), *sinks* (absorbing sound), or both. The latter case is particularly important in the context of musical performance since we can allow a node to absorb sound, apply DSP, and then emit the result back into the scene. By doing so, we create *internal* nodes, which become virtual effects units that process audio at some location in space. Each node is also represented with a parametric directivity pattern, giving the user highly accurate control over objects with which the node can interact.

Notice also that the concept of a listener within the scene is somewhat more abstract; While traditional systems allow for only one listener, our representation supports many. A listener is just a particular type of sink node (or group of sinks), where the audio absorbed by the sink is written to hardware buffers (eg. for headphones and speakers). We call this an *external* sink, since it relates to external or real-world entities. External sources also exist; they represent input from microphones, sound files, or other

audio streams.

Figure 2 illustrates a simple example of how internal and external sound nodes function together. The audio output of the scene is delivered to the user through two external sink nodes, which correspond to the real world headphones being worn. These are positioned in space relative to his virtual head position. When a user sings, their voice is propagated directionally into the scene and interacts with any internal nodes within range. Those nodes process the audio and emit the result back into the scene. Methods for tracking the user's head orientation and position relative to that of the virtual world are discussed in Sections 5.2 and 5.3.



2: Simple example: A singer's voice propagates directionally into the scene, gets processed, then propagates back toward the singer.

In addition to the advantage of supporting multiple simultaneous listeners, it is also possible to easily change the type of listening hardware by creating the appropriate nodes. This would be done for example, when the user switches from headphones to a multi-channel speaker configuration.

3.2 The Sound Connection

As mentioned earlier, a musical performance context benefits from the addition of various parameters to more finely control the audio propagation between nodes. Rather than having an identical propagation model for every node, we construct *sound connections* between every source-sink pair in the scene. This allows users the ability to enable and disable various connection features.

This is important for several reasons. The fact that a node has the ability to process audio and re-emit the result means that there is no guarantee of diminishing sound energy as audio makes its way through the scene. To prevent the occurrence of intensifying feedback loops, the the user must ensure that only acyclic connection chains exist. Conversely, feedback may be desirable from an artistic perspective, so the user may wish to create such loops among only certain nodes.

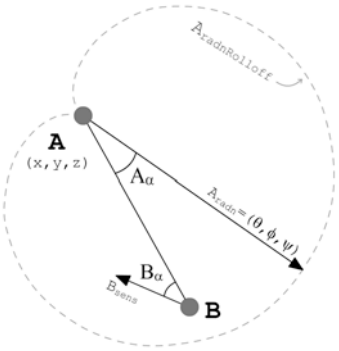
DISTANCE DECAY	The attenuation of the sound signal with distance
ANGULAR ROLL-OFF	The amount of decay relative to the angle of propagation
DOPPLER EFFECT	A variable delay as a function of distance
PROXIMITY EFFECT	A low-shelf filter for small distances (close proximity)
ABSORPTION FILTER	A low-pass filter simulating air absorption as a function of distance
INCIDENCE FILTER	A low-pass filter to simulate frequency-dependent attenuation with angle
REVERB	A filter with an impulse response representative of the current scene size

1: Features of a sound connection.

The *connection features* are described in Table 1. The user can specify the intensity of each feature in a given connection. This allows for some interesting special cases; for example, the user could "teleport" a sound signal between two very distant nodes by removing the effects of distance decay and Doppler. Additionally, diminishing the Doppler effect can be useful to prevent detuning of pitched material in musical passages.

3.3 Computation of audio propagation

Once a sound connection is established between two nodes and the various connection features have been specified, the computation of audio propagation is quite simple. Though the virtual sound nodes we saw in Figure 2 provided cone-shaped patterns of propagation, this is a simplification of more the elaborate mechanism shown in Figure 3.



3: A virtual sound node in more detail.

A source node $\mathbf{A}$, with a more complicated cardioid[2] directivity pattern has several properties, including a 3D position (x, y, z) and the orientation with which it *radiates* sound: $\mathbf{A}_{\text{radn}} = (\theta, \phi, \psi)$. It also has an angular roll-off, $A_{\mathbf{radnRolloff}}$, that describes how the radiation of sound is affected at various angles of propagation. The sink node $\mathbf{B}$, has similar properties though it's directivity is called $\mathbf{B}_{\text{sens}}$ and it's rolloff (not shown) is called $B_{\mathbf{sensRolloff}}$ to imply *sensitivity* rather than radiation.

3.3.1 Distance Decay

Sound decays exponentially according to the inverse square law [16]. We compute the distance gain, G_{dist}, that the signal should be multiplied by to simulate this decay:

$$G_{\text{dist}} = \frac{1}{(1 + |\mathbf{B} - \mathbf{A}|)^{\beta}} \tag{1}$$

The additional control parameter, β, is used to control the steepness of the exponential decay. When $\beta = 2$, this model is identical to natural sound decay.

3.3.2 Angular Roll-off

The angle of incidence between a node's orientation and the vector connecting the source and sink can be computed with basic vector mathematics. For example, A_{α} is found by:

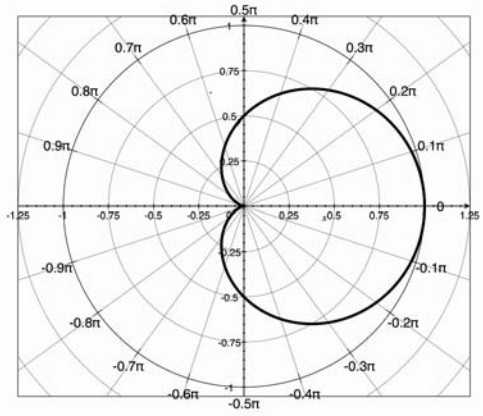
$$A_{\alpha} = \cos^{-1}\left(\frac{\mathbf{A}_{\text{radn}} \bullet (\mathbf{B} - \mathbf{A})}{|\mathbf{A}_{\text{radn}}|\,|\mathbf{B} - \mathbf{A}|}\right) \tag{2}$$

B_{α} is computed in the same fashion. Then we compute a gain value that simulates the sensitivity of the sink and radiation of the source with:

$$G_{\text{rolloff}} = A_{\mathbf{radnRolloff}}(A_{\alpha})\, B_{\mathbf{sensRolloff}}(B_{\alpha}) \tag{3}$$

[2]It's worthwhile to note that such directivity patterns are commonly found in the field of acoustics. Most directional microphones exhibit sound sensitivity similar to cardioids, and many traditional musical instruments have lobed radiation patterns [16].

While angular roll-off can be described by a mathematical equation, we choose to represent this with a lookup table for greater flexibility to experiment with custom roll-off functions. The cardioid pattern from Figure 3 is more nicely depicted in Figure 4. For each angle of incidence (i.e., A_α), the gain value is shown that indicates the level of attenuation based on incidence. At an angle of zero, the entry in the table returns 1.0. As the incidence increases to 90°, the gain drops to 0.5 until it finally reaches 0.0 at an angle 180°.



4: A rolloff function for cardioid sensitivity/radiation.

3.3.3 Doppler

Doppler is the apparent frequency shifting of a sound as a function of changing distance. This is accomplished using a variable delay embedded in each connection. The delay time corresponds to the amount of time it would take a sound to travel the distance between the connection's source and sink.

3.3.4 Other spatial filters

We attempt to model frequency-specific audio attenuation effects based on distance and angle. For example, high frequencies are quite directional and usually much quieter behind the source while low frequencies tend to wrap around and are heard equally in all directions. We simulate this with the use of a low-pass filter (which we call an *incidence filter*), whose cutoff frequency increases with the angle of incidence. High frequencies are also lost due to air absorption as the distance between source and sink increases. This too is modeled with a low-pass filter, called an *absorption filter*.

A common effect known to sound recording engineers, the *proximity effect*, models the boost of low frequency energy when the distance between source and sink is very small. This is accomplished with the use of a low-shelf filter, but only when when a threshold closeness is attained. Though this is a side effect of directional microphones in the real world, it serves as a good cue for spatial proximity. Furthermore, it is a familiar effect that most people will recognize, particularly performing musicians.

Another spatial effect that we consider is a simulation of reverberation by filtering with an impulse response function. While this impulse function should ideally be computed directly from the scene geometry, this has not yet been implemented in an automatic fashion. For now, the user must specify which type of room is closest to the desired effect, similar to Creative EAX filter choices such as hallway, arena, or bathroom. In future work, we plan to examine ray casting tecnhiques similar to the DIVA project [17] mentioned earlier, where additional sound sources would be automatically created at important reflection points. The balance of direct sound, reflections, and diffuse/reverberant sound will greatly add to the user's spatial awareness. The fact that our framework allows the user to tune each of these components is quite interesting. For example, a user could diminish directivity effects (angular roll-off and incidence filtering) when the distance between source and sink is larger than the reverberation radius. The reverberant sound would thus dominate and the sense of directivity would decrease, which is a common psychoacoustic effect. On the other hand, reverb may be heavily exaggerated or made to contradict directivity for an interesting artistic effect.

4. THE ROLE OF GRAPHICS

A further important distinction of our approach from other projects is the rich visual component. In our framework, graphics are not only used as an artistic medium, but are essential components of the editing and authoring environment. When dealing with large amounts of data and many control parameters, it is important to have suitable feedback. For example, the designer of a complex scene can visually perceive the spatial arrangement of sound nodes, when doing so by ear might be difficult or impossible. In performance, graphical feedback provides a high degree of precision when directing a sound towards a particular sink, especially when either are in motion.

4.1 Graphical Engine

Since users of the system are creating artistic content, we would like to provide a graphical engine that supports beautiful and convincingly believable content. This means that our engine should allow designers to develop 3D graphics with the most cutting edge tools on the market. Ideally, we would like to allow graphics development with any commercial application that designers might be comfortable with, and then provide a simple mechanism to incorporate these graphics into the virtual world.

The scene that is displayed will typically be composed of complicated, independently designed 3D models such as: people, musical instruments, audio equipment, and architectural elements such as furniture, walls, and floors. The spatial interactions between these elements will likely be high-level involving simple movements like translation, rotation, or scaling. We anticipate that the need to control these models at the vertex level will be rare, although the ability to animate various components of these models will be required - for example, articulating the body parts of a human avatar. With these constraints in mind, we have adopted a scene-graph representation for the virtual world.

A scene-graph is simply a data structure that organizes the logical and spatial elements of a 3D scene. Each element is represented as a node in a tree structure, such that any spatial transformations performed on a parent node will be propagated automatically to all children nodes. Scene-graphs are particularly useful for representing rigid structures and managing hierarchical scenes where groups of objects are animated together. By applying operations to the parent node, we can affect all other grouped elements without specifically indicating them. This holds not only for spatial commands, but also for logical commands such as hiding a group, pruning it from the scene, and propagating other higher-level parameters.

A custom 3D engine has been developed using the OpenSceneGraph (OSG) [2] graphics toolkit, that allows interactive control of various 3D models, with several audio-centric graphical elements made available to the user. The user is also provided with tools to distribute the rendering over several screens, so that a truly immersive environment can be created.

4.2 Sonigenic Graphical Display

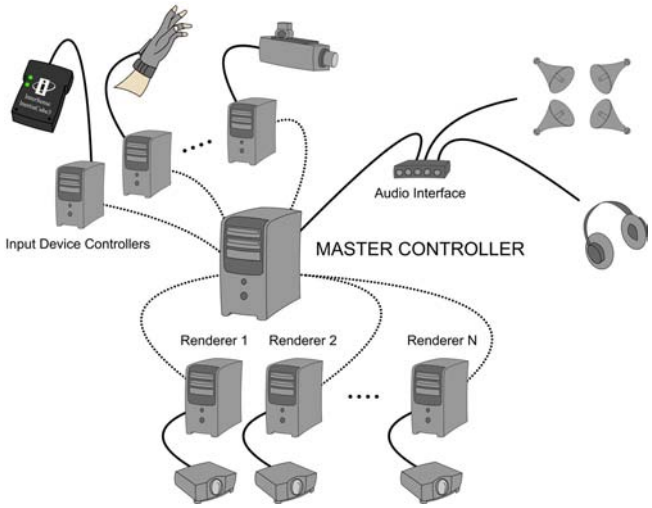
When dealing with audio propagation on such a controllable level, it may become difficult to conceptualize what processes

are taking place. The possibility of meaningful visualization is thus highly appealing. We provide users the ability to visualize the directivity of their sound nodes with a 'laser beam', wireframe mesh, or lighting affects. Each approach has its respective benefits and drawbacks. The laser beam only shows the direction vector of a sound node, while the wireframe mesh shows the radiation or sensitivity pattern as well. Lighting, though computationally expensive, can act as an appealing analog to the behavior of sound as it propagates. Light has similar decay properties (in terms of both distance and angle), but can only be used to visualize monotonically decreasing roll-offs.

For each sound node, an animated intensity meter is also available which displays the current sound energy contained within the node, similar to a home stereo VU meter. Other custom animations are possible using keyframe indexing of 3D models. The animations can be created using any 3D modeling application such as Maya or 3D Studio Max, and then assigned a control signal that determines the time index. This provides a powerful mechanism for sonigenic display, where audio parameters such as energy, envelope, or even beat detection can be rendered visually. Further details of this process can be found in Section 5.3.

5. CONTROL ENVIRONMENT

This system is developed with artists and designers in mind, and thus we aim for simple control mechanisms to allow such users to modify and extend the functionality. This is primarily accomplished with the inclusion of the PureData (Pd) [3] patcher programming language. Pd has a large community of music-focused programmers, who are continually developing new DSP and control methods with this language. Hence, we believe that it is the ideal candidate to avail powerful control in an extensible fashion.



5: Overview of how the 3D engine might be deployed in hardware. Note solid lines represent a physical connection while dotted lines might represent a network connection via TCP/UDP.

Figure 5 illustrates a sample hardware architecture, including various sensors and rendering technologies. This is similar to a CAVE-like [5] immersive environment.

5.1 Digital Signal Processing (DSP)

Any type of complex DSP can occur within a sound node. In fact, every node has a Pd abstraction with an incoming and outgoing signal, where users can insert their own DSP. For example, a user can filter the signal, add echo or delay, or process the sound according to some physical model. A node's spatial parameters (eg. position or orientation) can be used in the DSP computation, increasing the audio-visual coherence.

Some special DSP cases are worth pointing out. For example, if the user chooses not to send a signal back out, and rather writes the incoming signal to soundcard buffers, then they have created the external sink that was mentioned in Section 3.1. This is how real-world loudspeakers are modeled. On the other hand, if the user ignores the incoming signal and instead generates only an outgoing signal (perhaps by playing a sound file, or reading from the soundcard's input buffer), then they have created an external source. Input from the real world is modeled this way.

We should note that sound nodes are mono-channel audio objects. Multi-channel audio effects may be created with the 'grouping' mechanism, where sound nodes are arranged together in a scene-graph. As a result, any operation applied to the parent will propagate to the entire group. This applies not only to typical spatial transformations (such as translation or rotation), but also to sonic operations (such as establishing connections or distributing audio input). If we consider again the example in Figure 2, we note that the user's headphones and microphone should be grouped together, as children of a 3D head model. By translating the head, the source and sinks will automatically translate with it.

5.2 Input Daemons

Creating virtual musical instruments is of little value unless they can also be played. We believe that an excellent input mechanism is natural body motion. The human body is directly modeled in 3D and various control mappings can be implied from body posture or motion (gestures). Such control makes direct use of what we have already learned through daily practice with our bodies. In particular, kinesthetic feedback informs us of our body's orientation in space. This provides additional information in another modality channel, which again decreases the cognitive load imposed on the user, and allows for natural and expressive control [20].

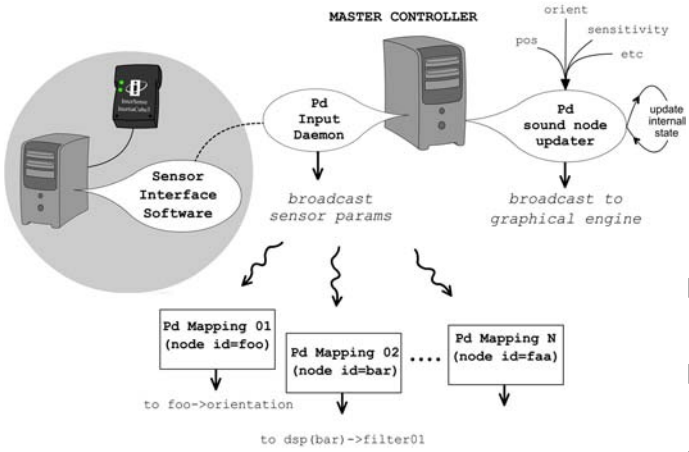
Currently, body tracking solutions are being investigated using computer-vision based approaches, although we note that performance of these methods remains disappointing due to a lack of both temporal and spatial precision. It has been demonstrated several times that musical systems are particularly demanding of low-latency control [21]. Thus we must include the possibility of using other input devices with lower latency and greater fidelity. Examples include data gloves, orientation sensors, accelerometers, and other such devices. Each device is interpreted by a 'Pd input daemon', which formats input signals into a standard message format (in the range of [0,1] for each parameter). These messages are available for control of DSP or more complicated mappings, as described in the following sections.

5.3 Mapping Considerations

One of the trickiest aspects of creating powerful musical interactions is the mapping of control parameters to musical effect. Many of today's so-called computer music interfaces either end up being too simple to achieve virtuosic playability in the long term, or too complicated to learn in the short term. Wessel & Wright [21] have made this point clear, and suggest that controllers have mappings to higher-level musical behaviors such as navigation through timbre space, or any multidimensional synthesis control. Hunt & Kirk [9] likewise suggest that 'holistic control' of a parameter space is better than sequential control using one-to-one mappings. They also mention the need for a 'performance mode', where the system "stays constant whilst the focus is on the improvement of the player". Our system takes inspiration from these ideas by offering a spatially immersive control paradigm.

Since users are both geometrically and perceptually encom-

passed within the instrument, they have the ability to explore complicated parameter configurations in a holistic manner. Multimodal feedback is provided so that users can easily learn how the system reacts to certain actions, and there are typically many methods to achieve similar results. We offer a 'performance mode', though the distinction between this and the 'editing mode' is fuzzy. An initial configuration of nodes can be created during an authoring phase, but this configuration can change in realtime during performance. For example, a user may choose to 'grab' a node and move it to a new location, or even animate the node in an orbiting fashion. Thus mappings are dynamic and manipulable, providing users with a dynamic mapping interface rather than a static performance system.



6: How mappings can be defined.

Figure 6 shows how Pd can help to realize more complex mappings. The node updater on the right receives parameters relating to a specific node id. The input daemon formats sensor readings into a standard format and broadcasts that information to any Pd mapping abstraction that a user might create. For example, one mapping might receive values from an orientation sensor and control the 3D orientation of a node while a different mapping uses those values to compute coefficients for a DSP filter.

6. CONCLUSION

We have presented an initial research framework for creating 3D scenes that function as spatial musical instruments. In addition to a more thorough representation of the 3D audio scene, we have introduced the concept of spatial navigation through a musical 'piece'. Powerful control mechanisms are provided that allow users to map parameters (either from input devices or generated by sonic events) to meaningful audio-visual outcomes. We believe that natural spatial mappings that humans typically encounter in the real world will make this paradigm particularly effective for musical expression.

7. 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.

8. REFERENCES

[1] Extensible 3d (x3d) iso standard (iso/iec 19775:2004). http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/.

[2] Openscenegraph. www.openscenegraph.org.

[3] Puredata. www.puredata.info.

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

[5] C. Cruz-Neira, D.J.Sandin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of SIGGRAPH '93*, pages 135–142. ACM Press, 1993.

[6] W. G. Gardner. *3-D Audio Using Loudspeakers*. Kluwer Academic Publishers, Norwell, MA, 1998.

[7] W. G. Gardner and K. D. Martin. Hrtf measurements of a kemar. *J. Acoustics Soc. of America*, 97(6):3907–3908, 1995.

[8] M. Gerzon. Periphony: With-height sound reproduction. *J. Audio Eng. Soc.*, 21(1):2–10, 1973.

[9] A. Hunt and R. Kirk. Mapping strategies for musical performance. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. Ircam - Centre Pompidou, 2000.

[10] J. M. Jot. Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia Systems*, 7(1):55–69, 1999.

[11] T. Maki-Patola, J. Laitinen, A. Kanerva, and T. Takala. Experiments with virtual reality instruments. In *Proceedings of NIME*, pages 11–16, 2005.

[12] A. Mulder, S. S. Fels, and K. Mase. Mapping virtual object manipulation to sound variation. In *Proceedings of the IPSJ-SIGMUS*, pages 63–68, Tokyo, Japan, Dec 1997.

[13] M. Naef, O. Staadt, and M. Gross. Spatialized audio rendering for immersive virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 65–72. ACM Press, 2002.

[14] J. Pressing. Some perspectives on performed sound and music in virtual environments. *Presence*, 6(4):482–503, 1997.

[15] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. *J. Audio Eng. Soc.*, 45(6):456–466, 1997.

[16] T. D. Rossing. *The Science of Sound (2nd edition)*. Addison-Wesley, 1990.

[17] L. Savioja, J. Huopaniemi, T. Lokki, and R. Vaananen. Creating interactive virtual acoustic environments. *J. Audio Eng. Soc.*, 47(9):675–705, 1999.

[18] G. Steinke. Surround sound - the next phase : An overview. *J. Audio Eng. Soc.*, 44(1):651.

[19] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of SIGGRAPH '01*, pages 545–552. ACM Press, 2001.

[20] R. Vertegaal, T. Ungvary, and M. Kieslinger. Towards a musician's cockpit: Transducers, feedback and musical function. In *Proceedings of ICMC*, 1996.

[21] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *Workshop at NIME*, 2001.