Soundtracks for Computer Animation : Sound Rendering in Dynamic Environments with Occlusions

Nicolas Tsingos, Jean-Dominique Gascuel
iMAGIS[†] / GRAVIR-IMAG
BP 53, F-38041 Grenoble cedex 09, France

Phone: (+33)(0)4-76-63-57-94 Fax: (+33)(0)4-76-44-66-75
email: Nicolas.Tsingos@imag.fr, Jean-Dominique.Gascuel@imag.fr

Abstract

With the development of virtual reality systems and multi-modal simulations, soundtrack generation is becoming a significant issue in computer graphics. In the context of computer generated animation, many more parameters than the sole object geometry as well as specific events can be used to generate, control and render a soundtrack that fits the object motions.

Producing a convincing soundtrack involves the rendering of the interactions of sound with the dynamic environment : in particular sound reflections and sound absorption due to partial occlusions, usually implying an unacceptable computational cost.

We present an integrated approach to sound and image rendering in a computer animation context, which allows the animator to recreate the process of sound recording, while "physical effects" are automatically computed. Moreover, our sound rendering process efficiently combines a sound reflection model and an attenuation model due to scattering/diffraction by partial occluders, through the use of graphics hardware allowing for interactive computation rates.

Résumé

Avec le développement des systèmes de réalité virtuelle et des simulations multi-modales, générer une bande son devient un problème qui touche de plus en plus le milieu de l'informatique graphique. Dans le contexte d'une animation de synthèse, de nombreux paramètres et évène - ments peuvent être utilisés pour générer, contrôler et effectuer le rendu d'une bande son accompagnant l'action.

Produire une bande son convaincante implique toutefois le rendu des interactions du son avec l'environnement dynamique, en particulier ses multiples réflections et l'absorption due aux occlusions partielles, nécessitant habituellement des calculs couteux.

Nous présentons une approche pour le rendu simultané du son et de l'image dans un contexte d'animation de synthèse permettant à l'animateur de recréer un processus de prise de son virtuelle, les calculs d'effets «physiques» étant réalisés automatiquement. Notre processus de rendu permet de prendre en compte, en temps interactif, les réflections du son ainsi qu'un modèle de dispersion/diffraction due aux occlusions partielles en utilisant le hardware spécialisé des stations graphiques.

Keywords: animation, multi-modal simulation, virtual acoustics.

1 Introduction

Sound is essential to enhance any visual experience, and should fit the visual information. Otherwise, the effect produced on the listener/viewer will be negative. In a computer animation context many parameters such as object positions, collision events, etc. can be used in order to generate a proper soundtrack which reproduces the interaction of sound with the dynamic environment. In particular the rendering process should simulate reflections and attenuations due to occluders on emitter-to-receiver paths. Often used in room acoustics, these simulations usually involve significant computation that makes their use impracticable in case of fully dynamic environments, where emitters, receivers, occluders and reflectors are mobile.

With the development of the so-called "3D-sound", many approaches have been proposed to simulate an acoustic room response in real-time. Usually these methods only allow for positioning a sound source in 3D space relative to a listener and rarely include an environment simulation. Moreover, even if sound reflections are handled, they do not take into account partial occlusions, leading to unrealistic results even in the case of moderately complex scenes.

[†]iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.

We present a new sound rendering approach in a computer animation environment which allows fast simulation of mobile sound sources in a fully dynamic environment with occlusions. Our sound renderer efficiently combines an "image source" model for sound reflections and an attenuation model due to scattering/diffraction by occluders. We use graphics hardware to perform fast "sound visibility" calculations. Other effects such as Doppler shifting or atmospheric scattering are also taken into account. Our method allows on-line editing of the scene during calculations. Sound is processed by digital filtering while the animation is computed, thus no post-processing is needed and the soundtrack is directly synchronized with the video signal.

In the first part of this paper we discuss some of the previous work aimed at rendering sound in virtual environments. In the second part we present our approach and review our rendering process. In the third part we present some results, address some implementation problems and conclude.

2 Virtual acoustics and computer animation

Various methods have been developed to compute the impulse response of a given environment - that is the response obtained when sending a single "sound impulse" (Dirac signal) in the environment - using ray-tracing, beam-tracing, or image sources methods [Kut91, SG89, MMV93, Eme95]. The first family of methods uses ray/beam tracing to determine all sound paths from the sound source to a given listening position. This is achieved by firing rays in all directions from the sound source and constructing the successive reflected rays until they reach a given "counting volume". The image sources method consists in constructing for each surface in the scene (assumed to be a plane pure specular reflector) an image source by mirroring the original sound source at the plane of the reflecting wall. This quite generic process can be iterated to any order of reflection, however it leads to a quadratic number of sources. Once the impulse response is computed, sound rendering can be performed by convolving this filter with a "dry" (monaural reverberation-less) sound [RM95, FWT91]. This process is often referred to as *auralization* [KDS93]. Widely used in room acoustics simulation, for concert hall planning for example, these precise methods quickly become impracticable in case of fully dynamic environments due to their high computational cost. Fortunately, it has been shown that the most significant information is carried by the early sound reflections, particularly lateral reflections [Bar71, Bla83]. Thus, some real-time approaches propose to recompute the early reflections only at each change in the environment while late reverberation is kept constant or modeled as artificial reverberation using recursive filtering methods [FWT91, HKVH94]. Nevertheless, taking into account the influence of occluders is impossible, if interactive computational time is required.

Another approach to model real-time moving sound sources in a reverberant environment is to design real-time spatializing processors [Jot92, Sal95, JW95, Moo83, Cho71]. This kind of processors provides real-time signal processing but does not allow effects such as partial occlusions by moving objects to be taken into account.

Several approaches have also been presented to integrate sound calculations in an animation environment, in the computer graphics field. In 1992, Takala and Hahn proposed a general methodology to perform integrated rendering of sound and images in a computer animation production [TH92, HGL+95, HFGL95]. They model a sound scene by associating a prototype sound with each object in the scene. Sounds are represented by parameterized "timbre-trees", which are similar to shade-trees for textures. Thus, instantiating sounds in the animation means mapping animation parameters to sound trees parameters. Trees can also contain special environmental nodes to create environmental effects (sound reflections for example). Once created and instantiated, sound trees are post-rendered using a resampling algorithm in the microphone space but no on-line calculation is performed. Recently, Noser and Thalmann proposed to integrate sampled sounds in a virtual animation environment [NT95]. The goal of their approach was to add virtual audition to synthetic autonomous actors, thus their sound rendering process remained quite simple and did not include any environment simulation.

Thus, none of the previous approaches seem convenient to render moving sound sources in a dynamic environment with occlusions at interactive to real-time rates.

3 Sound scene modeling and scripting

3.1 Sound scene modeling

Our sound scenes are composed of four kinds of "primitives": sound sources, microphones, sound blockers and sound reflectors. Sound sources and microphones define the sound to be rendered and the way this will be done (i.e. the recording device). Sound blockers and sound reflectors will allow environmental effects to be taken into account, in particular sound reflections and partial absorption/diffraction due to occluding dynamic objects.

Sound emitters and receptors

Sound sources and microphones can be defined as new 3D objects and animated over time. Their animation can be performed either using a key-frame based approach or any physically-based approach. Sound sources and visual 3D models can be connected to share the same position over time, in order to "attach" sounds to moving objects. "Sound textures" can then be attributed to different sound sources. As in previous work [NT95, TH92], a sound texture is defined as an AIFF sampled prototype sound which will be *mapped* to microphone space during sound rendering. Sampled sound textures appear to be simpler to use than ones synthesized directly, allowing for richer and more realistic sounds to be produced. Moreover, sound engineers usually work with specific recorded sounds, or taken from a sampled sound library [Mye95]. Special *foley* sound sources can also be defined. We will discuss the interest of defining such sources in section 4.3. Virtual microphones can also be created and moved in the same way as sound sources. Directive filtering of the sources and microphones can be defined as frequency-dependent directional functions. Basic microphones are monaural but can be easily combined together to simulate complex recording devices such as widely used stereophonic "AB ORTF" antennas [HW94, MR92], or human ears if specific filters such as Head Related Transfer Functions (HRTFs) are used [Beg91, WK89, Mol92, Beg94, GM94]. These filters, measured on a dummy head or directly on the listenner's ears, allow to recreate the human spatial perception of sound [Bla83].

Sound blockers and sound reflectors

Computing the propagation of sound in a complex environment with occlusions is a very difficult problem usually involving a significant computational cost. If the surface of the occluding object is flat, of the scale of the sound wavelength λ and its extent is large compared to the wavelength, then geometric optics can be used to compute sound reflections. This is typically the case for walls and ceilings. If the object is small compared with the wavelength then it will scatter the wave equally in all directions, the fractional intensity scattered being proportional to the sixth power of the diameter of the object. If the object sizes range from $\lambda/10$ to 10λ then scattering is complex even for simply shaped objects. There is similar complexity in the "sound shadows" cast by objects. If the object size is comparable to the wavelength, then diffraction will occur along edges, blur the edges of the shadow zone and will entirely eliminate the shadow at distances a few times the diameter of the object [FR91].

In the context of computer animation, we are more interested in producing a convincing soundtrack than a physically precise one. Nevertheless, effects produced should be realistic and take into account the influence of the environment. Thus, we introduce two types of "sound objects": sound blockers and sound reflectors that will render the influence of the 3D objects in the scene on the sound. Sound reflectors geometry (usually polygon sets) will be used to generate image sound sources to create some ambiance effects. Thus they should be used for large objects such as walls or ceilings. Sound blockers will render a sound attenuation due to a diffraction/scattering behaviour of the occluding objects on a source to microphone path. An object can of course be defined as a sound reflector and a sound blocker at the same time, and different geometries may be used for each of the two purposes. It is left to the animator to define what kind of effects he/she wants to generate according to the size of the objects in the scene and of course the scenario of his/her animation. We will detail in section 4.2 how we perform the combined rendering of sound blockers and sound reflectors.

3.2 Synchronization and Scripting

Synchronization of sound with the video signal is achieved by an on-line approach to sound rendering. Figure 1 shows how the simulation process can be structured to achieve this goal. Assuming that motion of all objects is recomputed at a fixed time-step, dt_{frame} (25Hz for example[1]), sound rendering is then performed at usual sampling rates (from 8kHz to 48kHz) during this time-step. Thus, no post-rendering is needed. This scheme also allows for pipelining of motion calculation with parallel processing of image and sound rendering.

Generating a sound means activating the appropriate sound source. Finally sound recorded through various microphones can be mixed using virtual mixing consoles to an AIFF file output or directly to the audio hardware. Activation and deactivation of sound sources can be defined by scripting over time in a sequencer based approach. Special events such as collision or friction, which are likely to generate sounds, can also be handled automatically by using specific callback functions. These functions are executed each time a specific event occurs and can be used to activate and control the intensity and the pitch of a sound texture by accessing various simulation parameters (kinetic energy of a contact for example). Since these scripts are interpreted, they can be modified and merged on-line during simulation, providing a very flexible way to control the scenario of an animation. New callback functions can easily be created to handle other specific situations like object creation or destruction, etc.

4 Sound rendering

In this section we detail our sound rendering pipeline, in particular how we combine our sound blockers and sound reflectors to efficiently take into account sound reflections and partial attenuation by occluders.

[1] even if motion must be integrated at finer time-steps, in case of physically-based systems.

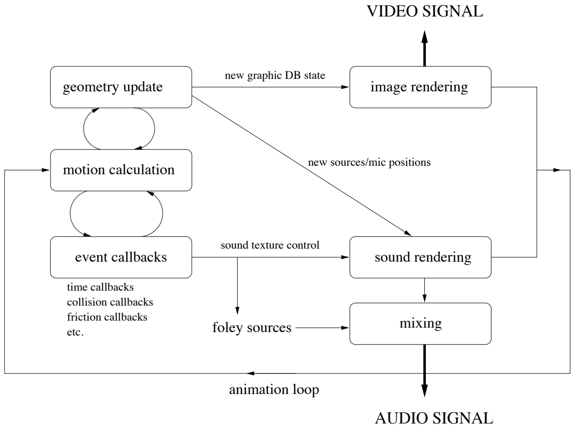


Figure 1: Overview of the animation process.

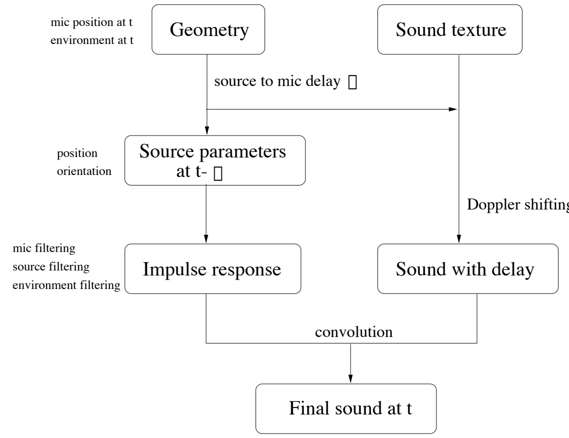


Figure 2: Sound rendering pipeline.

4.1 The rendering pipeline

To render the sound from a microphone we must determine the signal that reaches the microphone at each time t. As in previous approaches [TH92, NT95], the rendering algorithm needs to resample the original prototype sound texture. This resampling requires the knowledge of the position of sources in the past, i.e. at time $t-\tau$, τ being the sound propagation delay [DW83]. This can be maintained on-line during motion calculation as a list of source previous positions in key-frames. The position of a source at $t-\tau$ can be retrieved by interpolating between the appropriate key-frames. Sound propagation delay is calculated using an iterative process assuming that $\tau = c/d(mic_t, source_{t-\tau})$ where c is the sound propagation speed and d the Euclidean distance in 3D. Delay is recomputed for each sample frame, i.e. 44100 times per second for CD sound quality. Hence, Doppler shifting (i.e. shifting of sound frequency due to relative source to microphone motion) is automatically taken into account. Once the sound propagation delay is computed, sound texture is “back-mapped” to microphone space and convolved with a Finite Impulse Response (FIR) combining medium, source, microphone and environment filtering (Figure 2). We will describe now how we compute this FIR.

4.2 Combining sound blockers and image sources

At each time step dt_{frame}, image sources are created for each sound reflector. Contributions of each sound source (i.e. real sources and valid image sources) to each microphone are then computed and summed-up. This process involves the calculation of a FIR that reproduces sound filtering along source to microphone path. This FIR is the convolution of source filtering (and reflection filtering for image sources), microphone filtering, medium filtering and blockers filtering. In order to take into account indirect occlusion, image sources are tested against image blockers (Figure 3). To determine the fraction of energy absorbed by blockers we compute a frequency dependent geometric attenuation term. Evaluating this term for each octave band leads us to a filter that we will use to alter the original signal.

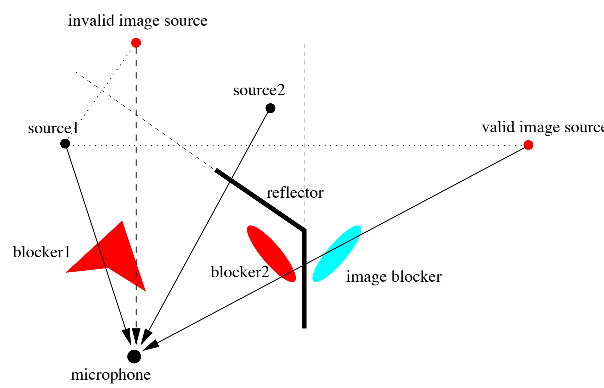


Figure 3: Combining sound blockers and image sources.

Computing a sound geometric attenuation term

We compute the fraction of sound which will be blocked by obstacles between an emitter and a receiver using *Fresnel zones* (Figure 4). *Fresnel zones* are volumes enclosed between ellipsoids E_k defined by:

$$E_k = \{M \in \mathbb{R}^3 / EM + MR = ER + \frac{k\lambda}{2}\}$$

where E is the emitter at time $t-\tau$, R the receiver at time t and λ the wavelength of the sound wave. We consider only the first Fresnel zone (i.e. for $k=1$) which is an ellipsoid.

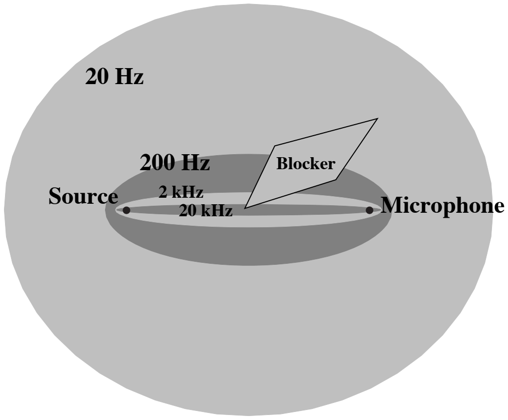


Figure 4: Fresnel zones in 2D. Source to microphone distance is 10 meters.

As we assume that most of the energy will propagate in these zones, we define a visibility criteria for sound between a source and a microphone at a given frequency by computing the amount of the volume of the appropriate Fresnel zone blocked by occluders [Vau94]. Since evaluating this volume is quite difficult in a general case, we use the graphics hardware to compute the corresponding visibility term. To compute this term we perform a rendering of all occluders from the microphone point of view using an appropriate viewing system (Figure 5). Our visibility term is obtained by counting all pixels that are set (i.e. not background) in the rendered images. Figure 6 shows an example of this process. Notice that grey level can be used to specify the transparency for blockers' materials at a given frequency. In order to take into account dissymmetric configurations we average this visibility information with a symmetric information computed from the sound source point of view to obtain the final attenuation term $V_{Fresnel_{\square}}$.

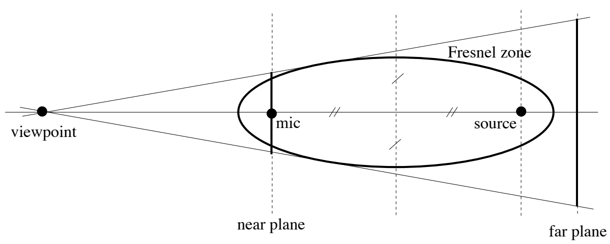


Figure 5: Viewing frustum used for sound visibility calculations.

We assume that objects' speeds are far smaller than sound speed. Hence, we can simplify this computation and evaluate visibility at time t using blockers' positions at t. Figures 7 and 8 show results of visibility calculations for two simple cases.

Finally, sound pressure attenuation along a sound path is given by:

$$A_{\square} = \frac{1}{d} \cdot \square_{\square}^{d} \cdot V_{Fresnel_{\square}}$$

where d is the Euclidean distance between the microphone and the sound source and $\square_{\square}$ is the medium scattering coefficient per meter. Atmospheric scattering is computed according to the *ISO 9613-1:1993* definition.

4.3 Mixing control

Sound mixing is essential if a complex soundtrack is to be generated. Examples are virtual reality applications where real-time audio output is needed because no post-production is available [PST+96]. Thus, several virtual mixing consoles can be defined in order to generate the various parts of the soundtrack. Each console can combine several virtual microphone entries to several output channels. *Foley* sound sources can be directly integrated in the mixing process without any treatment in order to provide ambient sounds or music. The mixing consoles may also be used to post-process the signal in order to perform specific encoding such as "Dolby surround"-like 4-channel format. This kind of format, allowing for the recreation of some spatial effects, is probably the most useful for animated movies soundtrack generation because it is standardized and now available at low cost for domestic applications [Dre88]. Moreover, this format is encoded on stereo tracks so it can be recorded on almost every support.

5 Implementation details

5.1 About filters implementation

Our filters are implemented as short 256 points FIRs. Material filtering associated with image sources and sound blockers is defined as octave-band filters. We use a cubic spline interpolation between octave-band values in order to deduce a minimal phase FIR. Directive filters are given for sampled directions in space. The appropriate filter is chosen by taking the one closest to the desired direction. In order to avoid clicks in the signal when changes occur along sound paths, final signal is computed by convolving the signal with the filters corresponding to the old and new paths and then crossfading the two resulting signals (Figure 9). Directional filters are usually given in octave bands except for HRTFs used for binaural processing which are derived from HRTF narrow band data measured with a KEMAR dummy head at the M.I.T. Media-Lab [GM94].

5.2 Time vs quality rendering

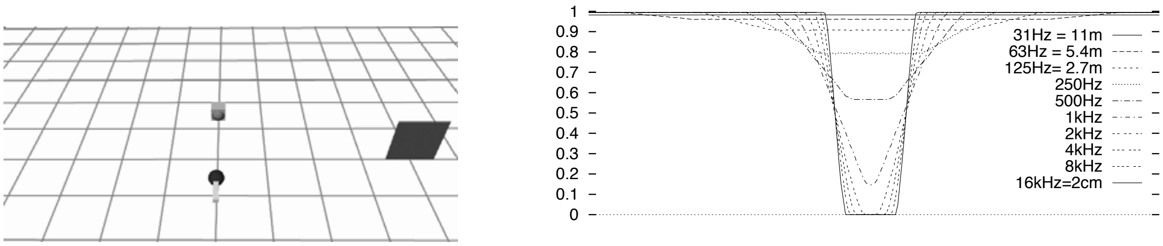
Our current implementation, on an Indigo2 Impact 200 MHz graphic workstation allows already near realtime

3D view showing microphone, source, occluders and Fresnel zones for 400 and 4000 Hz.

Visibility from microphone at 400Hz.

Visibility from microphone at 4000Hz.

Figure 6: Using graphic hardware for “sound visibility” calculations.



Experimental setup.

Visibility term in octave bands.

Figure 7: Visibility through time when an occluding square moving from right to left passes between a sound source and a microphone (The size of each square in the grid is $1m^2$).

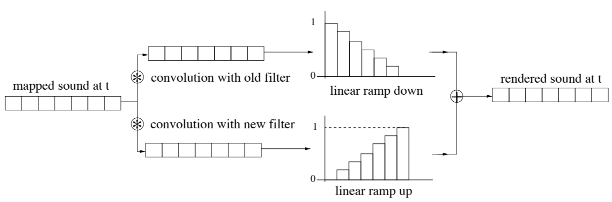


Figure 9: Crossfading sound in “image-space”.

rendering of sound for simple environments and first-order image sources. As we do not use any “convolution engine” to perform convolutions, signal processing time is not negligible. However, the image source model keeps being the main bottleneck of the method. So we plan to use our method to update precisely the first reflections while using an artificial reverberation with recursive filtering, sending the outputs of direct sound and image sources to the late reverberation filterer [HKVH94]. Computation time could be improved by reducing the number of frequencies used for visibility calculations, which are at the moment performed for ten octave bands. Another improvement could be to recompute the propagation delay at fixed time steps (i.e. 25Hz) and then perform some interpolation to get the delay for each sound frame (i.e. 44100 Hz).

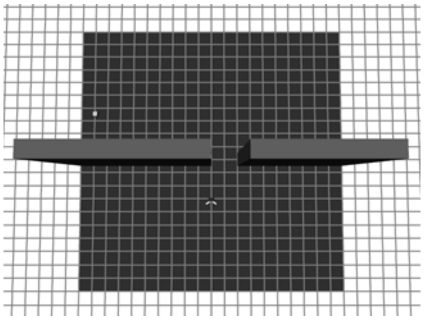
6 Conclusion

We have presented a new approach to sound rendering in a computer animation environment. We use combined image sources and frequency dependent “sound visibility” calculations to integrate effects of scattering/diffraction due to partial occlusions in our sound propagation model. This approach leads to realistic sound recording situations which, to our knowledge, are not correctly handled by previous approaches. Unlike common ray-tracing based methods, we take advantage of graphics hardware to achieve interactive computation rates for fully dynamic complex environments.

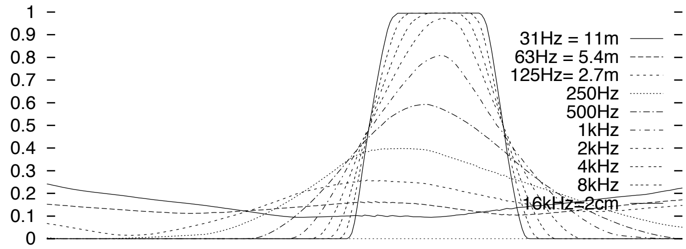
We plan to implement our method on a multi-processor graphic platform in order to obtain realtime performance for virtual reality applications.

Acknowledgements

We would like to thank all members of the iMAGIS project for their moral and technical support. Special thanks go to Jacques Martin and Eric Dudouet from C.S.T.B. Grenoble, “Room Acoustics Division”, for interesting and helpful chats, bibliography elements and various other data.



Experimental setup.



Visibility term in octave bands.

Figure 8: Visibility through time when a sound source passes in front of a "door" in a wall. Microphone and sound source are disposed respectively in front and behind the wall. Sound source is moving from left to right. Since obstacles are limited in extend, low frequencies tend to "wrap around" walls and floor (The size of each square in the grid is $1m^2$).

References

[Bar71] M. Barron. The subjective effects of first reflections in concert halls - the need for lateral reflections. *Journal of sound and vibration*, 15(4), 1971.

[Beg91] D.R. Begault. Challenges to the successful implementation of 3d sound. *Journal of the audio engineering society*, 39(11), November 1991.

[Beg94] Durand R. Begault. *3D Sound for Virtual Reality and Multimedia*. Academic Press Professional, 1994.

[Bla83] J. Blauert. *Spatial Hearing : The Psychophysics of Human Sound Localization*. M.I.T. Press, Cambridge, MA, 1983.

[Cho71] J.M. Chowning. The simulation of moving sound sources. *Journal of the Audio Engineering Society*, (19), 1971.

[Dre88] R. Dressler. Dolby prologic surround decoder principles of operation. Technical Report S89/8540/8624, Dolby laboratories Inc., 1988.

[DW83] A.P. Dowling and J.E. Ffowcs Williams. *Sound and Sources of Sound*. Ellis Horwood publishers, 1983.

[Eme95] Marc Emerit. *Simulation Binaurale de l'Acoustique de Salles de Concert*. PhD thesis, Centre Scientifique et Technique du Batiment, 1995.

[FR91] N.H. Fletcher and T.D. Rossing. *The Physics of Musical Instruments*. Springer Verlag, 1991.

[FWT91] S.H. Foster, E.M. Wenzel, and R.M. Taylor. Realtime synthesis of complex environments. *proceedings of the ASSP (IEEE) Workshop on Application of Signal Processing to Audio and Acoustics*, 1991.

[GM94] Bill Gardner and Keith Martin. Hrtf measurments of a kemar dummy-head microphone. Technical Report 280, M.I.T. Media Lab Perceptual Computing, may 1994.

[HFGL95] J. Hahn, H. Fouad, L. Gritz, and J. Wong Lee. Integrating sounds and motions in virtual environments. *Sound for Animation and Virtual Reality, Siggraph'95 Course #10 Notes*, August 1995.

[HGL+95] J. Hahn, J. Geigel, J. Wong Lee, L. Gritz, T. Takala, and S. Mishra. An integrated approach to motion and sound. *Journal of Visualisation and Computer Animation*, 28(2), July 1995.

[HKVH94] J. Huipaniemi, M. Karjalainen, V. Valimaki, and T. Huotilainen. Virtual instruments in virtual rooms - a realtime binaural room simulation environment for physical models of musical instruments. *Proceedings of ICMC*, 1994.

[HW94] C. Hugonnet and P. Walder. *Theorie et Pratique de la Prise de Son Stereophonique*. Eyrolles ed., 1994.

[Jot92] J.M. Jot. An analysis/synthesis approach to real-time artificial reverberation. *Proceedings of ICASSP*, 1992.

[JW95] J.M. Jot and O. Warusfel. Le spatialisateur. *Proceedings of Rencontres Musicales Pluridisciplinaires, Le Son & L'espace, Lyon*, March 1995.

[KDS93] M. Kleiner, B. Dalenbâck, and P. Svensson. Auralization - an overview. *Journal of the Audio Engineering Society*, 41(11), November 1993.

[Kut91] Heinrich Kuttruff. *Room Acoustics (3rd edition)*. Elseiver applied science, 1991.

[MMV93] J. Martin, D. Van Maercke, and J.P. Vian. Binaural simulation of concert halls : A new approach for the binaural reverberation process. *Journal Of The Acoustical Society Of America*, 6, December 1993.

[Mol92] H. Moller. Fundamentals of binaural technology. *Applied acoustics*, 36, 1992.

[Moo83] F.R. Moore. A general model for spatial processing of sounds. *Computer Music Journal*, 7(3), Fall 1983.

[MR92] T. McCormick and F. Rumsey. *Sound and Recording : An Introduction*. Butterworth-Heinemann, 1992.

[Mye95] T. Myers. Sound in live action and animation : A comparative look. *Sound for Animation and Virtual Reality, Siggraph'95 Course #10 Notes*, August 1995.

[NT95] H. Noser and D. Thalmann. Synthetic vision and audition for digital actors. *Proceedings of EUROGRAPHICS'95*, 1995.

[PST+96] R. Pausch, J. Snoddy, R. Taylor, S. Watson, and E. Haseltine. Disney's aladdin: 1st steps toward storytelling in virtual reality. *Proceedings of SIGGRAPH'96 conference. Orlando. Florida.*, August 1996.

[RM95] A. Reilly and D. McGrath. Convolution processing for realistic reverberation. *98th AES Convention Proceedings, Paris*, February 1995.

[Sal95] C. Salès. Processeur d'espace sonore roland rss-10. *Proceedings of Rencontres Musicales Pluridisciplinaires, Le Son & L'espace, Lyon*, March 1995.

[SG89] Adam Stettner and Donald P. Greenberg. Computer graphics vizualisation for acoustic simulation. *ACM Computer Graphics, SIGGRAPH'89 Proceedings*, 23(3), July 1989.

[TH92] Tapio Takala and James Hahn. Sound rendering. *ACM Computer Graphics, SIGGRAPH'92 Proceedings*, 28(2), July 1992.

[Vau94] R. Vauzelle. *Un Modèle de Diffraction en 3D dans le Premier Ellipsoide de Fresnel*. PhD thesis, Université de Poitiers - Laboratoire de signaux et communication, 1994.

[WK89] F. Wightman and D.J. Kistler. Headphone simulation of free-field listening (part 1 and 2). *Journal of the acoustical society of America*, 85(2), February 1989.