
INTERACTIVE GEOMETRIC SOUND PROPAGATION AND RENDERING

AUTHORS: MICAH TAYLOR, ANISH CHANDAK, LAKULISH ANTANI, DINESH MANOCHA

ABSTRACT:

We describe a novel algorithm and system for sound propagation and rendering in virtual environments and media applications. Our approach uses geometric propagation techniques for fast computation of propagation paths from a source to a listener and takes into account specular reflections, diffuse reflections, and edge diffraction. In order to perform fast path computation, we use a unified ray-based representation to efficiently trace discrete rays as well as volumetric ray-frusta. Furthermore, our propagation algorithm scales well with the number of cores, and uses interactive audio rendering technique to generate spatialized audio signals. The overall approach can render sound in dynamic scenes, allowing source, listener, and obstacle motion, and we show its performance on game-like and architectural environments. To the best of our knowledge, this is the first interactive sound rendering system that can perform plausible sound propagation and rendering in dynamic virtual environments.

DESCRIPTION OF TECHNIQUES UTILIZED:

INTRODUCTION

Realistic sound rendering can directly impact the perceived realism of users of interactive media applications. An accurate acoustic response for a virtual environment is attuned according to the geometric representation of the environment. This response can convey important details about the environment, such as the location and motion of objects. The most common approach to sound rendering is a two-stage process:

- Sound propagation: the computation of impulse responses (IRs) that represent an acoustic space.
- Audio rendering: the generation of spatialized audio signal from the impulse responses and dry (anechoically recorded or synthetically generated) source signals.

Sound propagation from a source to a listener conveys information about the size of the space surrounding the sound source, and conveys the source position to the listener even when the source is not directly visible. This considerably improves the immersion in virtual environments. For instance, in a first-person shooter game scenario, the cries of a monster or the steps of an opponent approaching can alert the player to danger. Sound propagation is also used for acoustic prototyping for computer games, architectural buildings, and urban scenes. Audio rendering also provides sound cues which give directional information about the 3D position of the sound source relative to a listener. For example, in a VR combat simulation it is critical to simulate the 3D sounds of machine guns, bombs, and missiles. Another application of 3D audio is user interface design, where sound cues are used to search for data on a multi-window screen.

Main Results: We present an algorithm and system for interactive sound rendering in complex and dynamic virtual environments. Our approach is based on high-frequency approximations of sound waves. Many such algorithms have been proposed for interactive sound propagation [6, 12, 23]. However, they are either limited to static virtual environments or can only handle propagation paths corresponding to specular reflections.

In order to perform interactive sound rendering, we use methods that can be parallelized across many processors. Our propagation algorithms use a hybrid ray-based representation that traces discrete rays [14] and ray-frusta [18]. Discrete ray tracing is used for diffuse reflections, and frustum tracing is used to compute the propagation paths for specular reflections and edge diffraction. We fill in the late reverberations using statistical methods. We also describe an audio rendering pipeline combining specular reflections, diffuse reflections, diffraction, 3D sound, and late reverberation.

Our interactive sound rendering system can handle models consisting of tens of thousands of scene primitives (e.g., triangles), as well as dynamic scenes with moving sound sources, listener, and scene objects. We can perform interactive sound propagation including specular reflections, diffuse reflections, and diffraction of up to 3 orders on a multi-core PC.

PREVIOUS WORK

In this section, we give a brief overview of prior work in acoustic simulation. In this paper, we focus on the areas of interactive sound propagation and audio rendering.

SOUND PROPAGATION

Sound propagation deals with modeling how sound waves propagate through a medium. Effects such as reflections, transmission, and diffraction are the important components. Sound propagation algorithms can be classified into two approaches: numerical methods and geometric methods.

Numerical Methods: These methods [7, 21] solve the sound wave equation numerically to perform sound propagation. However, despite recent advances [22], these methods are too slow for interactive applications, and are limited to static scenes only.

Geometric Methods: The most widely used methods for interactive sound propagation are based on geometric acoustics (GA). GA techniques represent acoustic waves as rays and can accurately model the early reflections (up to 3 – 6 orders). They compute propagation paths from a sound source to the listener. Specular reflections of sound are modeled with the image-source method [2]. Image-source methods recursively reflect the source point about all of the geometry in the scene to find specular reflection paths. Some methods compute exact visibility [11, 15], some use sampling [14], and some fall in between [6, 17].

There has also been work on complementing specular reflections with diffraction effects. Diffraction effects are very noticeable at corners, as the diffraction causes the sound wave to propagate in regions that are not directly visible to the sound source. Diffraction effects have been used in several interactive simulations [4, 25, 27].

Another important effect that can be modeled with GA is diffuse reflections. Diffuse reflections have been shown to be important for modeling sound propagation [8] and can be modeled with ray tracing based methods [13].

The GA methods described thus far are used to render the early reflections. The rest of the acoustic response, late reverberation, must also be calculated. This is often done through statistical methods [10] or ray tracing [9]. Sound propagation deals with modeling how sound waves propagate through a medium. Effects such as reflections, transmission, and diffraction are the important components. Sound propagation algorithms can be classified into two approaches: numerical methods and geometric methods.

AUDIO RENDERING

Audio rendering is the final step that generates audio for output over headphones or speakers. In the context of geometric sound propagation, it involves using the

propagation paths to create impulse responses which represent how an input sound is changed by the environment. Once the impulse response is known, it can be convolved with a dry input audio signal to produce the output audio. Rendering dynamic scenes is challenging, and there has been work on generating artifact-free audio rendering in such scenes [26, 28]. Introducing 3D cues in the final audio signals is an important effect, and requires convolution of an incoming sound wave with a Head Related Impulse Response (HRIR) [16].

OVERVIEW

In this section, we give an overview of the wave effects our system simulates, and highlight the main components. The main components are further detailed in the following sections.

ACOUSTIC MODELING

All GA techniques deal with finding propagation paths between each source and the listener. The sound waves travel from a source (e.g., a speaker) and arrive at a listener (e.g., a user) by traveling along multiple propagation paths representing different sequences of reflections, diffraction, and refractions at the surfaces of the environment. Figure 1 shows an example of such paths. In this paper, we limit ourselves to reflections and diffraction paths. The overall effect of these propagation paths is to add reverberation (echoes) to the dry sound signal. Geometric propagation algorithms need to account for different wave effects that directly influence the response generated at the listener.



Figure 1: Example scene showing (left) specular, (middle) diffraction, and (right) diffuse propagation paths.

When a small, pointlike sound source generates nondirectional sound, the pressure wave expands out in a spherical shape. If the listener is set a short distance from the source, the wave field eventually encounters the listener. Due to the spreading of the field, the amplitude at the listener is attenuated. The corresponding GA component is a direct path from the source to the listener. This path represents the sound field that is diminished by distance attenuation.

As the sound field propagates, it is likely that the sound field will also encounter objects in the scene. These objects may interact with the waves. Large objects can reflect the field specularly, as a mirror does for light waves. Surfaces that have fine details or

roughness of the same order as the wavelength can diffusely reflect the sound wave. This means that the wave is not specularly reflected, but reflected in a scattered manner. These diffuse reflections complement the specular components [8]. Diffraction effects occur at the edges of objects and cause the sound field to be scattered around the edge. As a listener moves behind a corner, a shadow region is encountered where sound cannot directly reach. Diffraction effects provide a smooth transition as the listener moves into this region.

The previous effects contribute to the early reflection components of the IR. As propagation continues, the amplitude of the sound slowly decays, forming the late reverberation components. Late reverberation is related to the scene size [10] and conveys an important sense of space.

SYSTEM COMPONENTS

Our system consists of three main processing steps. These are outlined in Figure 2. For details on the system components, refer to [24].

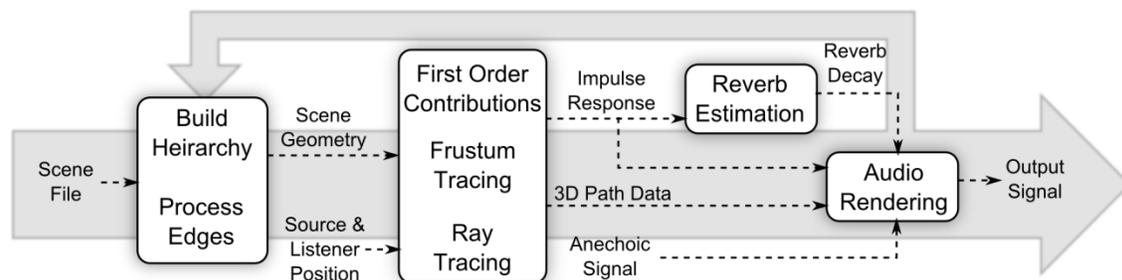


Figure 2: The main components of our sound pipeline: scene preprocessing; geometric propagation for specular, diffuse, and diffraction components; estimation of reverberation from impulse response; and final audio rendering.

Preprocessing: Our system uses a unified ray representation for specular reflections, diffuse reflections, and diffraction path computations. Thus, as part of preprocessing, a bounding volume hierarchy is created for the scene. Recent work allows such hierarchies to be built in parallel on multi-core systems [19]. This hierarchy allows fast intersection tests and is updated when the objects in the scene move [20]. The edges of objects in the scene are also analyzed to determine appropriate edges for diffraction.

Interactive Sound Propagation: This stage computes the paths between the source and the listener. A volumetric frustum tracer is used to find the specular and edge diffraction paths. A stochastic ray tracer is used to compute the diffuse paths.

Audio Rendering: After the paths are computed, they need to be auralized. A statistical reverberation filter is estimated using the path data. Using the paths and the

reverberation filter as input, the waveform is attenuated by the auralization system. The resulting signal represents the acoustic response and is output to the system speakers.

INTERACTIVE SOUND PROPAGATION

In this section, we give an overview of our sound propagation algorithm. Propagation is the most expensive step in the overall sound rendering pipeline. This large computational cost arises from the calculation of the acoustic paths that sound takes as it is reflected or scattered by the objects in the scene. The direct sound contribution is easily modeled by casting a ray between the source and listener. If the path is not obstructed, there is a direct contribution from the source to the listener. The other propagation components are more expensive to compute, so we parallelize all possible ray operations across multiple cores.

FRUSTUM TRACING

We use volumetric frustum tracing [18] to calculate the specular paths between the source and listener. The frusta are constructed of 4 bounding rays. From a point sound source, we cast many of these frustum primitives such that all the space around the source is covered. Each frustum is intersected with the scene primitives. The frustum is specularly reflected, and this gives rise to another frustum that is recursively propagated.

It is also possible to create diffraction frusta [25] using the Uniform Theory of Diffraction (UTD). When reflecting a frustum off a triangle face, the triangle edges are checked whether they are marked as diffracting edges. If so, a new diffraction frustum is swept out in the shadow region behind the triangle. This frustum then propagates through the scene as normal.

Reflection and diffraction continue until a specified order of recursion is achieved. To increase the intersection accuracy, we adaptively subdivide each frustum into new frusta at each reflection or diffraction [6].

For any of these frusta, if the listener is contained within the volume, there must exist some sound path from the source to the listener. The path is extrapolated through all parent frusta back to the source position to find the path. The path distance is used to calculate the time it takes for the sound to reach the listener and the reflection and diffraction count is used to attenuate the amplitude. Figure 3(left) shows a visual overview of steps in the frustum engine.

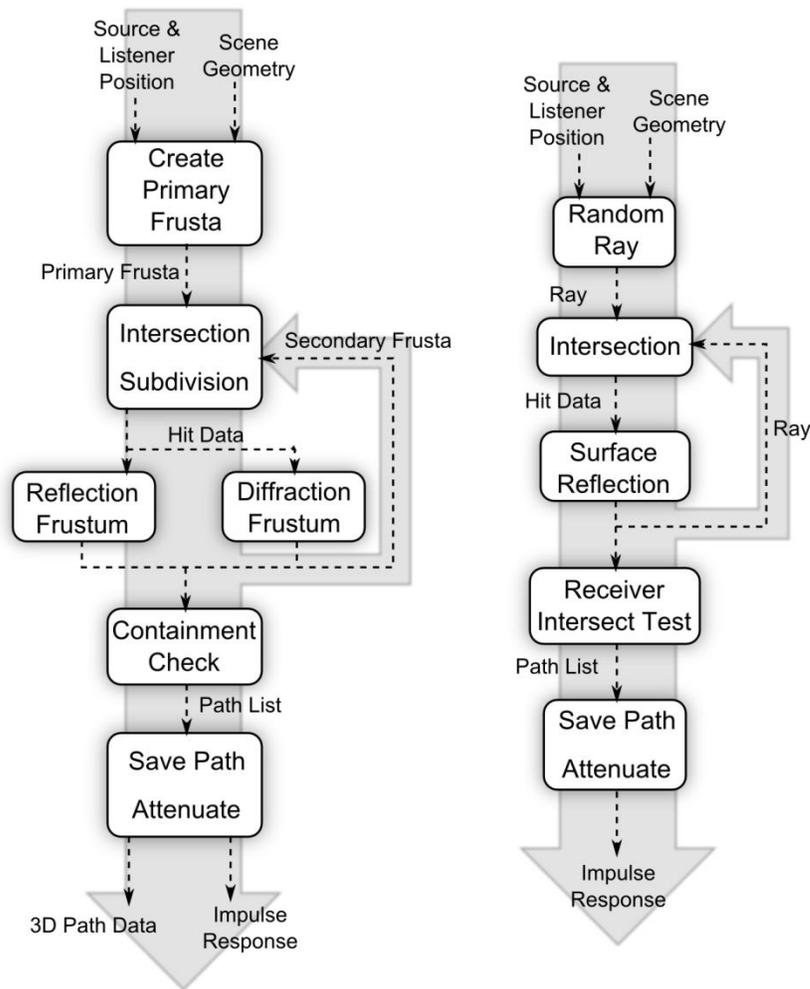


Figure 3: Unified ray engine: Both (left) frustum tracing and (right) ray tracing share similar rendering pipeline.

RAY TRACING

In order to compute sound reflected off diffuse materials, we use a stochastic ray tracer (Figure 3 (right)). Rays are traced out from the sound source in all the directions. When a ray encounters a triangle, it is reflected and tracing continues. The reflection direction is determined by the surface material. The listener is modeled by a sphere that approximates the listener's head. As the rays propagate, we check for intersections with this sphere. If there is an intersection, the path distance and the surfaces encountered are recorded for the audio rendering step. All contribution paths are adjusted [9] based on the reflection materials encountered. The accumulated paths are then sent to the audio rendering stage for output.

REVERBERATION ESTIMATION

The propagation paths computed by the frustum tracer and stochastic ray tracer described in Section 4 are used only for the early reflections that reach the listener. While they provide important perceptual cues for spatial localization of the source, capturing late reflections (reverberation) contributes significantly to the perceived realism of the sound simulation.

We use well-known statistical acoustics models to estimate the reverberant tail of the energy IR. The Eyring model [10] is one such model that describes the energy decay within a single room as a function of time.

Given the energy IR computed using GA, we fit a curve to the data. From the curve, we can estimate the RT60, which is defined as the time required for the energy to decay by 60 dB (Figure 4). The RT60 value is used in the audio rendering step to generate late reverberation effects.

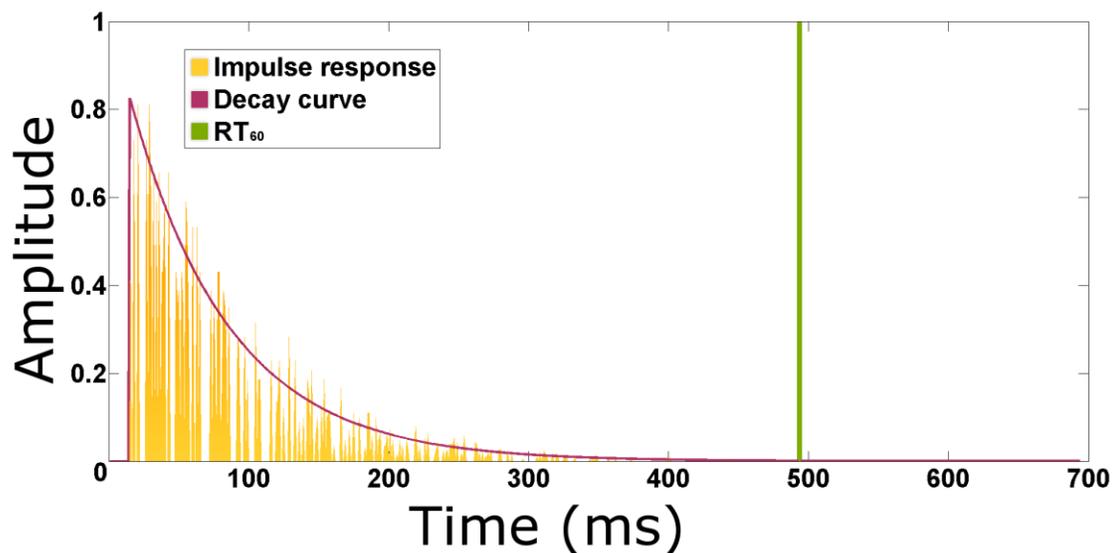


Figure 4: Extrapolating the IR to estimate late reverberation: The red curve is obtained from a least-squares fit (in log-space) of the energy IR computed by GA, and is used to add the reverberant tail to the IR.

AUDIO RENDERING

Audio rendering is the process of generating an audio signal which can be heard by a listener using headphones or speakers. In this section, we provide details on the real-time audio rendering pipeline implemented in our interactive sound propagation system.

Our sound propagation algorithm generates a list of specular, diffuse, and diffracted paths from each source to the listener. These paths are accessed asynchronously by the audio rendering pipeline as shown in Figure 5. The direction of the contribution paths

arriving at the listener is used to introduce 3D sound cues in the final audio. Additionally, since the source, listener, and scene objects can move dynamically, the path data sent to audio rendering may vary greatly from one propagation cycle to the next. Thus, our approach mitigates the occurrence of artifacts by various means. Our system also uses the previously described reverberation data to construct the appropriate sound filters.

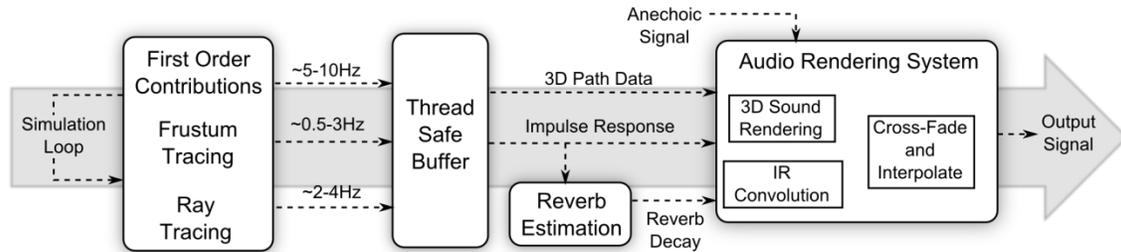


Figure 5: An overview of the integration of audio rendering system with the sound propagation engine. The propagation paths are input to the audio rendering system. The audio rendering system queries the buffer and performs convolution and 3D audio mixing. Crossfading and interpolation components smooth the final audio output signal.

INTEGRATION WITH SOUND PROPAGATION

Using the contribution paths from the propagation stage, an impulse response (IR) is created. These impulse responses store the acoustic response of the room, that is, the delay and attenuation effects that the room has on sound signals. The IRs are then convolved with the input audio to compute output audio.

ISSUES WITH DYNAMIC SCENES

Our sound propagation system is general and can handle moving sources, moving listener, and dynamic geometric primitives. Due to the motion of the sources, listener, and scene objects, the propagation paths could change dramatically and thus the IRs computed could change dramatically. Therefore, we impose physical restrictions on the motion of sources, listener, and the geometric primitives to produce artifact-free audio rendering. To further mitigate the effects of the changing paths, we use the current and previous IR data to crossfade when outputting a changing audio signal.

3D SOUND RENDERING

In a typical sound simulation, many sound waves reach the listener from different directions. These waves diffract around the listener's head and provide cues regarding

the direction of the incoming wave. This diffraction effect can be encoded in a Head-Related Impulse Response (HRIR) [1]. Thus, to produce a realistic 3D sound rendering effect, each incoming path to the listener can be convolved with an HRIR. However, for large numbers of contributions this computation can quickly become expensive and it may not be possible to perform audio rendering in real-time. Thus, only direct and first order reflections are used in 3D audio output.

ADDING LATE REVERBERATION

To add late reverberation effects, we use an artificial reverberation filter, which can add late decay effects to a sound signal. The previously calculated RT60 value is used to set the decay of the filter. This approach provides a simple, efficient way of complementing the computed IRs with late reverberation effects.

PERFORMANCE

Our system makes use of several levels of parallel algorithms to accelerate the computation. Ray tracing is known to be a highly parallelizable algorithm, and our system threads to take advantage of multi-core computers. Our results show that frustum tracing is also highly parallelizable (Figure 6). Additionally, frustum tracing uses vector instructions to perform operations on a frustum’s bounding rays in parallel. Using these optimizations, our system achieves interactive performance on common multi-core PCs.

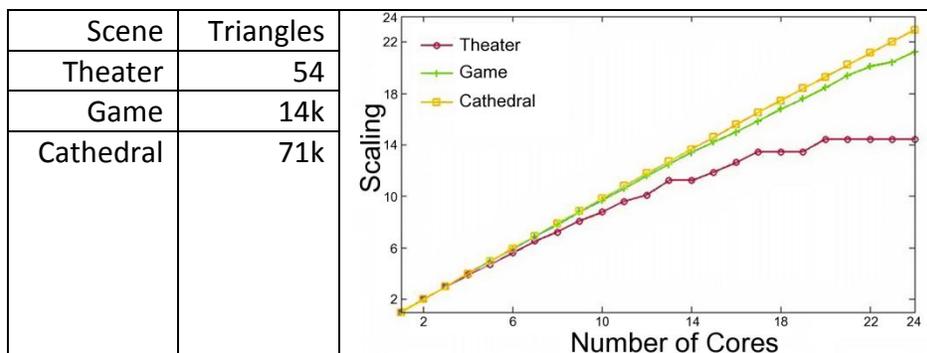


Figure 6: Parallel scaling: frustum tracing scales nearly linearly with number of cores on large scenes

In this section, we highlight the performance of our system. We show the parallel nature of our algorithms and highlight each subsystem’s performance on a varying set of scenes. The details of the scenes and system performance are presented in Table 1, and the scenes are visually shown in Figure 7. For performance timings, we ran on a multi-core Intel® Xeon® X5355 2.66Ghz processor-based system; the number of cores varies

per component. Only the propagation components are parallelized using multiple cores, as the reverberation and rendering components have low time cost.

Scene	Triangles	Specular + diffraction (3 orders)			Specular + Diffraction (1 order)			Diffuse (3 orders)	
		Time (ms)	Frusta	Paths	Time (ms)	Frusta	Paths	Time (ms)	Paths
Room	6k	359	278k	4	77	7k	3	274	228
Conference	282k	1137	320k	7	157	5k	2	323	318
Sibenik	71k	2810	900k	14	460	10k	5	437	26
Sponza	66k	1304	598	8	260	10k	3	51	120

While the ease of parallel ray tracing is known, the nature of frustum tracing is unknown. When testing frustum tracing on large multi-core PCs, we have recorded nearly linear speedup with addition processing cores, as shown in Figure 7. As shown in the Theater scene, even very small workloads scale well using over 10 cores.



Figure 7: Test scenes used (left to right): Room, Conference, Sibenik, and Sponza.

Specular and Diffraction: We generate two separate IRs using frustum tracing. One IR includes only the first order specular and diffraction contributions. Since these paths are fast to compute, we devote one core to this task. The other IR we generate includes the contributions for 3 orders of reflection and 2 orders of diffraction. This is done using 7 cores. The performance details for both simulation cycles are described in Table 1.

Diffuse tracing: Our diffuse tracer stochastically samples the scene space during propagation. As such, the rays are largely incoherent and it is difficult to use ray packets. Nonetheless, even when tracing individual rays, our system can render at interactive rates as shown in the performance table. The timings are for 200k rays with 3 reflections using 7 cores.

QUALITY AND LIMITATIONS

While the underlying algorithms are based on the physical properties of high frequency acoustic waves, there are limitations to our methods. As such, we discuss the limitations of each component in our system. We also note the benefits that such an approach offers over simpler audio rendering systems.

LIMITATIONS

Our algorithm has several limitations. The accuracy of our algorithm is limited by the use of underlying GA algorithms. In practice, GA is only accurate for higher frequencies. Moreover, the accuracy of our frustum-tracing reflection and diffraction varies as a function of maximum subdivision. Our diffraction formulation is based on the UTD and assumes that the edge lengths are significantly larger than the wavelength. Also, frustum tracing based diffraction also is limited in the types of diffraction paths that can be found. Our approach for computing the diffuse IR is subject to statistical error [9] that must be overcome with dense sampling. In terms of audio rendering, we impose physical restrictions on the motion of the source, listener, and scene objects to generate an artifactfree rendering.

BENEFITS

Interactive audio simulations used in current applications are often very simple and use precomputed reverberation effects and arbitrary attenuations. In our approach, the delays and attenuations for both reflection and diffraction are based on physical approximations. This allows the resulting system to generate acoustic responses that are expected given scene materials and layout. In addition to calculating physically based attenuations and delays, our method also provides accurate acoustic spatialization. Simple binaural rendering often only uses the direct path, which may not be valid. With our approach, the reflection and diffraction path directions are included. Consider a situation when the sound source is hidden from the listener's view (Figure 8). In this case, without reflection and diffraction, the directional component of the sound field appears to pass through the occluder. However, propagation paths generated by our system arrive at the listener with a physically accurate directional component.

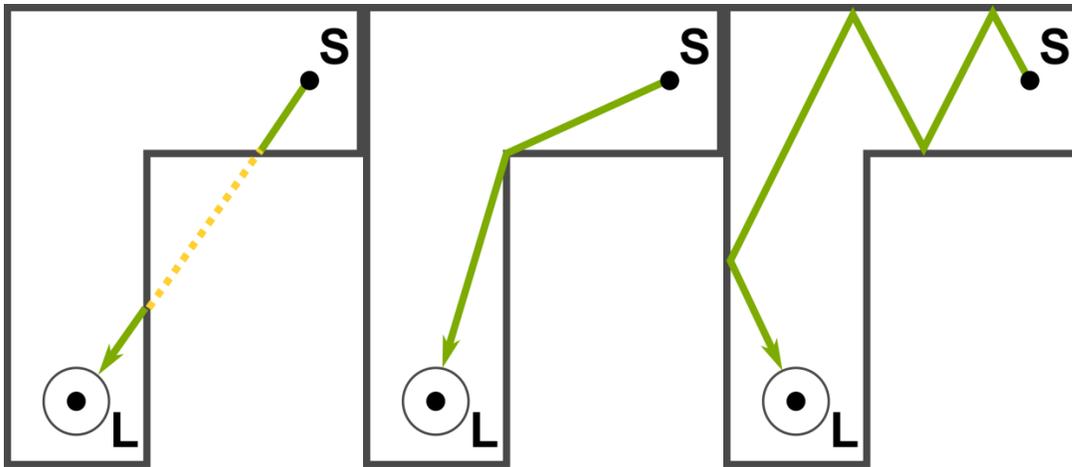


Figure 8: Path direction from sound source S to listener L: (left) The time direct path is physically impossible, but (middle) diffraction and (right) reflection paths direct the listener as physically expected.

CONCLUSION

We have presented an interactive sound rendering system for dynamic virtual environments. Our system uses GA methods to compute the propagation paths. By using an underlying ray-based representation, we compute specular reflections, diffuse reflections, and edge diffraction in parallel on multi-core systems. We also use statistical late reverberation estimation techniques and present an interactive audio rendering algorithm for dynamic virtual environments. We believe our system is the first to generate plausible interactive sound rendering in complex, dynamic virtual environments.

There are many areas to explore in the future. We are adapting our system to perform objectaccurate specular reflections [5] and more advanced edge diffraction [3].

ACKNOWLEDGMENTS

This research is supported in part by ARO Contract W911NF-04-1-0088, NSF award 0636208 , DARPA/RDECOM Contracts N61339-04-C-0043 and WR91CRB-08-C-0137, Intel, and Microsoft.

REFERENCES:

- 1) V. Algazi, R. Duda, and D. Thompson. The CIPIC HRTF Database. In IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, 2001.

- 2) J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, April 1979.
- 3) L. Antani, A. Chandak, M. Taylor, and D. Manocha. Fast geometric sound propagation with finite-edge diffraction. Technical report, Department of Computer Science, University of North Carolina at Chapel Hill, 2009.
- 4) F. Antonacci, M. Foco, A. Sarti, and S. Tubaro. Fast modeling of acoustic reflections and diffraction in complex environments using visibility diagrams. In *Proceedings of 12th European Signal Processing Conference (EUSIPCO '04)*, pages 1773–1776, September 2004.
- 5) A. Chandak, L. Antani, M. Taylor, and D. Manocha. Fastv: From-point visibility culling on complex models. In *Eurographics Symposium on Rendering*, 2009.
- 6) A. Chandak, C. Lauterbach, M. Taylor, Z. Ren, and D. Manocha. AD-Frustum: Adaptive Frustum Tracing for Interactive Sound Propagation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1707–1722, Nov.-Dec. 2008.
- 7) R. Ciskowski and C. Brebbia. *Boundary Element methods in acoustics*. Computational Mechanics Publications and Elsevier Applied Science, 1991.
- 8) B.-I. Dalenbäck, M. Kleiner, and P. Svensson. A Macroscopic View of Diffuse Reflection. *Journal of the Audio Engineering Society (JAES)*, 42(10):793–807, October 1994.
- 9) J. J. Embrechts. Broad spectrum diffusion model for room acoustics ray-tracing algorithms. *The Journal of the Acoustical Society of America*, 107(4):2068–2081, 2000.
- 10) C. F. Eyring. Reverberation time in “dead” rooms. *The Journal of the Acoustical Society of America*, 1(2A):217–241, January 1930.
- 11) T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proc. Of ACM SIGGRAPH*, pages 21–32, 1998.
- 12) T. Funkhouser, N. Tsingos, and J.-M. Jot. *Survey of Methods for Modeling Sound Propagation in Interactive Virtual Environment Systems*. Presence and Teleoperation, 2003.
- 13) B. Kapralos, M. Jenkin, and E. Miliotis. Acoustic Modeling Utilizing an Acoustic Version of Phonon Mapping. In *Proc. of IEEE Workshop on HAVE*, 2004.
- 14) A. Krokstad, S. Strom, and S. Sorsdal. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration*, 8(1):118–125, July 1968.
- 15) S. Laine, S. Siltanen, T. Lokki, and L. Savioja. Accelerated beam tracing algorithm. *Applied Acoustic*, 70(1):172–181, 2009.
- 16) V. Larcher, O. Warusfel, J.-M. Jot, and J. Guyard. Study and comparison of efficient methods for 3-d audio spatialization based on linear decomposition of hrtf data. In *Audio Engineering Society 108th Convention preprints*, page preprint no. 5097, January 2000.

- 17) C. Lauterbach, A. Chandak, and D. Manocha. Adaptive sampling for frustum-based sound propagation in complex and dynamic environments. In Proceedings of the 19th International Congress on Acoustics, 2007.
- 18) C. Lauterbach, A. Chandak, and D. Manocha. Interactive sound rendering in complex and dynamic scenes using frustum tracing. IEEE Transactions on Visualization and Computer Graphics, 13(6):1672–1679, Nov.-Dec. 2007.
- 19) C. Lauterbach, M. Garland, S. Sengupta, D. Luebke, and D. Manocha. Fast bvh construction on gpus. In Proc. Eurographics '09, 2009.
- 20) C. Lauterbach, S. Yoon, D. Tuft, and D. Manocha. RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs. IEEE Symposium on Interactive Ray Tracing, 2006.
- 21) P. Monk. Finite Element Methods for Maxwell's Equations. Oxford University Press, 2003.
- 22) N. Raghuvanshi, N. Galoppo, and M. C. Lin. Accelerated wave-based acoustics simulation. In ACM Solid and Physical Modeling Symposium, 2008.
- 23) S. Siltanen, T. Lokki, S. Kiminki, and L. Savioja. The room acoustic rendering equation. The Journal of the Acoustical Society of America, 122(3):1624–1635, September 2007.
- 24) M. Taylor, A. Chandak, L. Antani, and D. Manocha. Resound: interactive sound rendering for dynamic virtual environments. In MM '09: Proceedings of the seventeen ACM international conference on Multimedia, pages 271–280, New York, NY, USA, 2009. ACM.
- 25) M. Taylor, A. Chandak, Z. Ren, C. Lauterbach, and D. Manocha. Fast Edge-Diffraction for Sound Propagation in Complex Virtual Environments. In EAA Auralization Symposium, Espoo, Finland, June 2009.
- 26) N. Tsingos. A versatile software architecture for virtual audio simulations. In International Conference on Auditory Display (ICAD), Espoo, Finland, 2001.
- 27) N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In Proc. of ACM SIGGRAPH, pages 545–552, 2001.
- 28) E. Wenzel, J. Miller, and J. Abel. A software-based system for interactive spatial sound synthesis. In International Conference on Auditory Display (ICAD), Atlanta, GA, April 2000.

FURTHER RESOURCES:

Sound Synthesis and Propagation: <http://gamma.cs.unc.edu/research/sound/>

RESound Project: <http://gamma.cs.unc.edu/Sound/RESound/>

RESound work on youtube:

<http://www.youtube.com/watch?v=odoniBfWXcs>

<http://www.youtube.com/watch?v=3aqSB-Ocr04>