19th INTERNATIONAL CONGRESS ON ACOUSTICS
MADRID, 2-7 SEPTEMBER 2007

Adaptive sampling for frustum-based sound propagation in complex and dynamic environments

PACS: 43.55.Ka,43.58.Ta

Lauterbach, Christian[1]; Chandak, Anish[2]; Manocha, Dinesh[3]
[1]University of North Carolina at Chapel Hill, CB #3175, Chapel Hill, NC 27599, USA; cl@cs.unc.edu
[2]University of North Carolina at Chapel Hill, CB #3175, Chapel Hill, NC 27599, USA; achandak@cs.unc.edu
[3]University of North Carolina at Chapel Hill, CB #3175, Chapel Hill, NC 27599, USA; cl@cs.unc.edu

ABSTRACT
Frustum tracing is a geometric method that allows to perform sound propagation simulation in general complex and dynamic scenes. We describe the underlying algorithm and analyze its behavior with regard to different sampling and scene parameters. We also present an extension of the basic technique that allows to adaptively change the sampling with regard to local geometric complexity and show that this yields the same quality as uniform high sampling at much lower simulation time. We demonstrate our results in an interactive system for intended for virtual environments and prototyping purposes running in real-time on a current high-end PC.

INTRODUCTION

Modelling the acoustic properties of a virtual environment is an important step in simulating the acoustic response of buildings such as auditoriums, but also for interactive applications such as games or virtual environments where spatialized sound adds a stronger immersion effect to the application. Geometric algorithms are usually the fastest choice for this simulation, but are only accurate for the early reflections in sound propagation paths. However, they allow a good approximation of the full simulation, especially with statistically added reverberations, for purposes of prototyping or where absolute accuracy is not an issue. Considering that still both a high number of paths and reflections need to be taken into account, this is a very compute intensive task. Many algorithms for this therefore take a long time to simulate or do not scale to complex 3D environments.

This paper describes a new method for geometric sound propagation, *frustum tracing,* based on recent advances in interactive ray tracing. This allows us to perform simulations at much higher simulation speed, achieving interactive performance on a workstation system. In addition, this approach also handles general, non-densely-occluded scenes as well as complex dynamic environments. The frustum tracing approach is most closely related to previous methods using beam tracing [11], but can trade off quality for speed by using a discrete approximation to the clipping algorithm used in beam tracing. We show that even at low quality settings, our algorithm generates a good quality result, and converges to the same result as beam tracing very quickly. Finally, we also present an extension to the original algorithm that chooses the sampling parameters for frustum tracing adaptively. We show that this allows us to generate responses of higher quality in the same time as needed before.

The rest of the paper is organized as follows: first, we show previous and related work, with a special focus on giving an overview of recent advances in interactive ray tracing that motivated our work. We then describe the original frustum tracing algorithm and the extension to adaptive sampling. Afterwards, we show results from an implementation of that algorithm. Finally, we conclude and list areas of future work.

PREVIOUS WORK

Sound Propagation and Acoustic Modeling

Broadly, previous approaches to sound propagation can be categorized into numerical and geometric techniques. The former solve the Helmholtz-Kirchoff integral equation describing the wave field of the sound. Since this field cannot be solved analytically, the scene is usually discretized and the equations solved numerically instead. Examples of this are boundary-element methods [18] or finite-element methods [19]. In general, this produces a very accurate result, but is expensive both in memory as well as computation time and therefore not suitable in an interactive context.

Geometric methods simulate sound directly based on the rectilinear propagation of waves, which allows them to model the early reflections in a scene. Ray and stochastic path tracing methods [13,15] were among the first to be used for sound propagation, but may need a very high number of samples or important paths may be missed. If the listener is moving, this can also result in artifacts due the variance in the estimate. Some approaches using a discrete particle representation of sound called sonels or phonons have recently been developed [1] but so far only demonstrated on simple scenes and also suffer from sampling problems. Image source methods [2,3] explicitly mirror the sound sources at all object surfaces by constructing virtual sound sources. However, for increasing number of reflections the number of reflections increases exponentially both in computation and memory requirements, which makes this technique feasible only in very simple environments.

The most promising recent method has been beam tracing [6,8,7], which avoids the sampling problems of path tracing by using volumetric beams. Beams are recursively traced out from the sound source throughout the scene by intersecting them with all surfaces in the scene and constructing reflected, transmitted and diffracted beams for all hit surfaces. Since the beams fully cover the space, there is no sampling problem as in ray tracing. However, beam tracing is computationally expensive and current techniques require a precomputed BSP tree and therefore cannot handle dynamic scenes. The need for a precomputed high-level structure with cells and portals also limits beam tracing to environments that are very densely-occluded, i.e. most points are not visible to each other, such as indoor scenes with many small rooms. This means that open scenes such as an auditorium cannot be handled with current implementations.

Interactive Ray Tracing

Ray tracing is an important foundation for many algorithms used in areas such as visual rendering, wave and nuclear simulations or line-of-sight computations, among others. Most recent interest in ray tracing has occured in Computer Graphics where several factors contributed to the development of interactive ray tracing as an active research area. First of all, the continued growth of computational power of CPUs according to Moore’s law coupled with the fact that ray tracing is embarrassingly simple to parallelize has led to the availability of enough processing speed to perform ray tracing in real time even on commodity hardware. More interestingly, recent research in visual rendering has led to the development of algorithms exploiting *ray coherence,* i.e. the common behaviour of rays with similar directions, for greatly increased performance. Wald *et al.* [21] showed a ray tracer that was able to bundle groups of rays into so-called ray packets and process them in parallel, through which they achieved a 2 to 4 times overall improvement in speed. Subsequent improvements led to multi-level ray tracing [20], which was able to improve on this by an order of magnitude.

However, all of these approaches are limited in that they can only handle static environments due to a pre-computed acceleration structure. Recent work has concentrated on applying the same ray coherence approaches to alternative structures such as grids [23] and bounding volume hierarchies (BVHs) [22,16] that can either be constructed on the fly or updated efficiently. For BVHs, previous work has shown that ray coherence techniques provide similar or even improved benefits, while at the same time dynamic environments are possible since the acceleration structure can be adapted very efficiently at run-time to any motion. At the same time, the BVH can handle any arbitrary input models and it therefore well-suited for general environments. So far, all these techniques have been limited to applications in visual rendering;

however, the good results achieved in that area are a strong motivation to apply them to other areas as well.

FRUSTUM TRACING

We will now give an overview of the frustum tracing algorithm and its relation to previous methods. The main motivation in developing this method was to apply the algorithmic advances described in the previous section to the sound propagation problem in order to achieve similar performance results. At the same time, the advantages of beam tracing such as the volumetric beams should be maintained. Since ray coherence methods are critically dependent on having groups of coherent rays available, previous ray tracing approaches to sound propagation are not good candidates for this, as the randomly sampled rays used in the stochastic sampling process do not exhibit that kind of coherence.

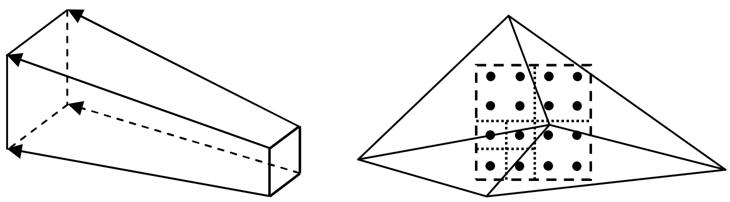

Figure 1: Left: A frustum as used in the algorithm. Right: Samples hitting the same object are recombined afterwards.

The frustum tracing algorithm can essentially be viewed as a discrete approximation of beam tracing: instead of using a polygonal beam that is intersected with geometric objects and can therefore attain complex, non-convex shapes, we use a simple frustum (i.e. cut-off pyramid) as the basic primitive. To support intersection of the frustum with surfaces, the frustum is uniformly subdivided into sub-frusta (see Fig. 1), which are then represented by a sample ray for intersection purposes. All of these rays are combined into a ray packet, which means we can apply ray coherence methods to speed up the ray tracing process significantly. Since the sample rays are uniformly sampled, we can also use some simplifications during the intersection method to avoid having to test all rays. Once the intersection with scene objects has been fully determined, secondary frusta representing reflections and transmissions are constructed from that information. Since the number of frusta grows exponentially with the number of reflection simulated, it is of a significant important to generate only as few secondary frusta as possible. Therefore, instead of a new frustum for each sub-frustum hitting an object, we combine sub-frusta that hit the same surface (see Fig. 1). Please refer to [17] for more detail on this process.

We use a BVH as an acceleration structure. As was described in the previous section, recent advances have made it possible to use the BVH with ray coherence methods and also update it to use it for dynamic environments. In our case, it is also easily possible to test the frustum for intersection with the BVH, which we use to find the set of intersecting geometric surfaces efficiently. The use of a BVH allows handling arbitrary and general scenes without being limited to densely-occluded environments.

SAMPLING PARAMETERS FOR FRUSTUM TRACING

The choice of the subdivision parameter for the uniform subdivision has the most important effect on the quality of the simulated reflection and therefore the accuracy of the computed reflection paths. Consider Fig. 2 showing a comparison of beam tracing and frustum tracing in the context of construction of reflections and transmissions. For beam tracing (on the left), the constructed beams are aligned precisely at object boundaries since beam tracing performs an exact intersection using clipping. For frustum tracing, secondary frusta can only be generated at boundaries of sub-frusta. Even though no gaps are introduced, this means the size of the reflection may be over- or underestimated. The higher the subdivision factor, the lesser will be the effect of this sub-sampling. At the limit, this algorithm becomes exactly the equivalent of beam tracing. Since beam tracing has been demonstrated [7] to provide a good approximation for the first reflections, we can compare the quality of our approach for different parameters to see how it converges.

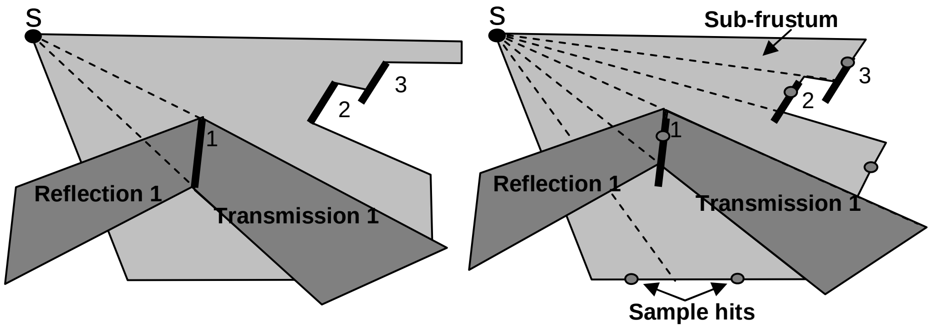

Figure 2: Effects of discrete sampling. Left: beam tracing. Right: frustum tracing. Sample locations are indicated by dots.

One of the main approaches in many applications when faced with sampling problems is to adaptively change the sampling density depending on the local complexity. Some of the earliest ray tracing approaches [24] already used this to shoot more rays at points of interest, and subsequent techniques improved on this [9,10]. However, those adaptive sampling approaches are intrinsically based on shooting of single rays in multiple passes, which has the significant disadvantage that it is not amenable to ray coherence approaches. The frustum tracing approach substitutes adaptive high sampling with uniform high sampling instead, which in practice is much faster despite being theoretically inferior.

Still, it is possible to add adaptivity to the frustum tracing algorithm by selecting the uniform subdivision parameter per frustum (as opposed to adaptively sampling inside the frustum). The general idea for our sampling approach here is that the sampling density should be roughly proportional to the area subtended by the frustum at the objects that it hits. We therefore choose the subdivision parameter based on the distance and spread of the frustum (see Fig. 3). The distance is compared to the overall size of the virtual environment, and the sampling is increased in intervals to maintain the relative size of the sub-frusta to scene geometry. Although there are other alternative ways to choose the subdivision factor, we found that this approach works very well in practice and can be implemented easily.

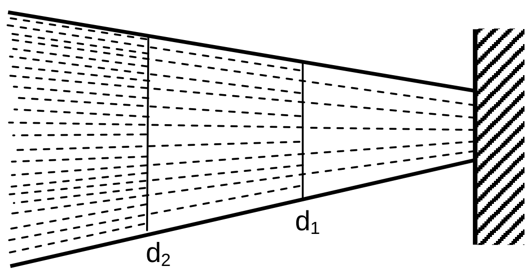

Figure 3: Scaling the sampling density based on the distance.

RESULTS AND DISCUSSION

We now present results from the adaptive frustum tracing algorithm. We have tested the adaptive algorithm on two different environments, a simple scene of two connected boxes as a simple case and a cathedral model with roughly 190,000 triangles as an example of a real-world architectural model. Here, we are interested in the implications of adaptive uniform sampling versus normal frustum tracing both in terms of quality and performance and refer to [17] for a more detailed discussion of performance results for frustum tracing. Figures 4 shows the

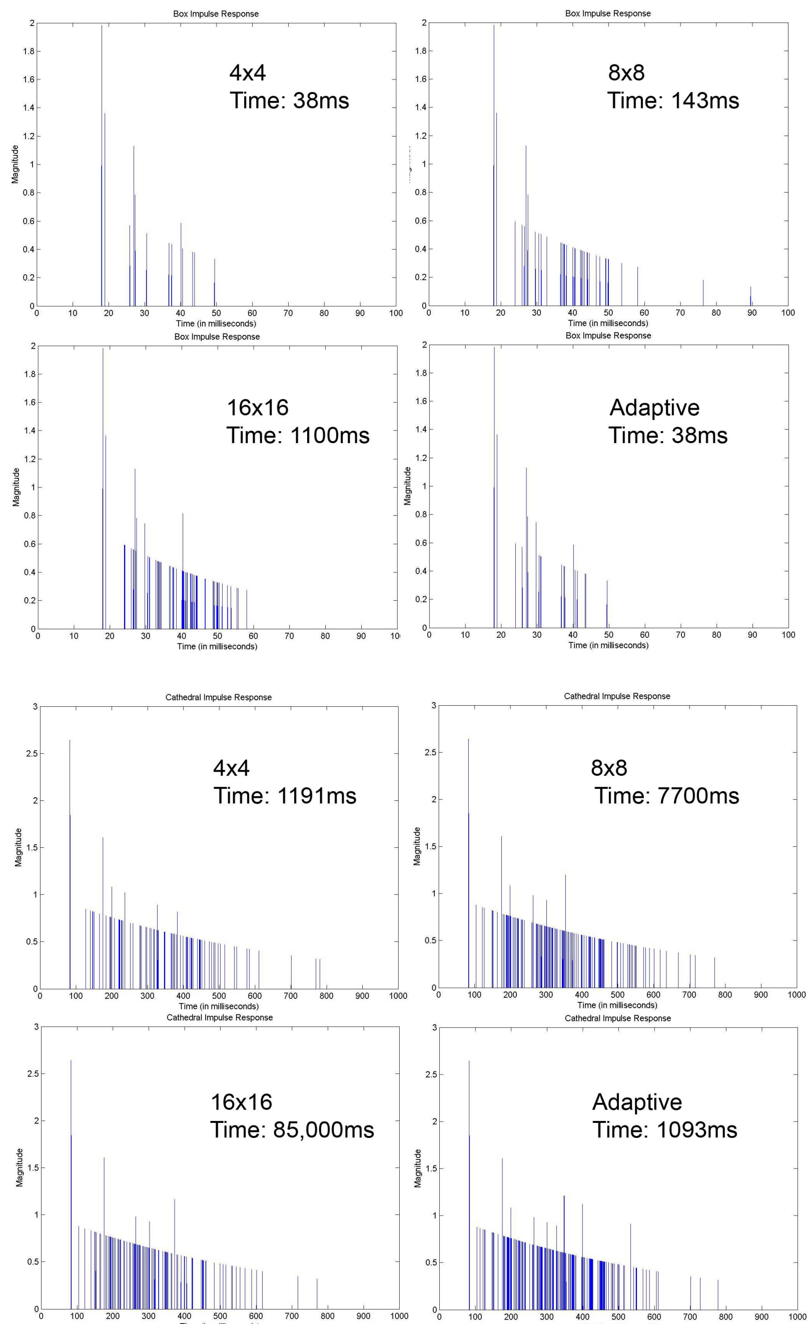

Figure 4: Quality results for the boxes (top) and cathedral (bottom) scenes. The graphs show the impulse response function for different uniform sampling parameters as well as adaptive sampling.

results for both scenes for a simulation of up to three reflections as an impulse response function. The results converge noticeable as the uniform sampling parameter is increased, but note that adaptive sampling always performs better than the low-sampling solution, but is as fast. This effect is particularly pronounced on the complex cathedral model where high sampling rates are necessary for good quality due to the high detail.

CONCLUSION AND FUTURE WORK

We have presented the frustum tracing algorithm with an extension to adaptive sampling for better quality and less simulation time. The algorithm uses recent advances in interactive ray tracing research to achieve real-time simulation performance, allowing the use of dynamic environments and moving sound sources, while having no limitations as to the kind of virtual scene that can be handled. For future work, we are interested in investigating integration of the sound simulation into an interactive application such as a virtual reality environment or games. In that context, we also plan to investigate combining the frustum tracing solution with a pre-computed numerical simulation for increased realism, which may allow better estimation of the late reverberations. Finally, the current simulation does not yet take diffraction into account, which may add important contribution paths. We intend to extend the frustum tracing algorithm to include these effects as well.

ACKNOWLEDGEMENTS

The Candlestick theatre model is courtesy of Charles Ehrlich and the cathedral model is courtesy of Chris Neill. We would like to thank Paul Calamia for helpful feedback, and Nikunj Raghuvanshi for help with the sound implementation. This work was supported in part by ARO Contracts DAAD19-02-1-0390 and W911NF-04-1-0088, NSF awards 0400134 and 0118743, ONR Contract N00014-01-1-0496, DARPA/RDECOM Contract N61339-04-C-0043 and Intel.

References: [1] M. Bertram, E. Deines, J. Mohring, J. Jegorovs, and H. Hagen: Phonon tracing for auralization and visualization of sound. In: Proceedings of IEEE Visualization 2005, pages 151–158, 2005.

[2] J. Borish: Extension of the image model to arbitrary polyhedra. Journal of the Acoustical Society of America, 75(6):1827–1836, 1984.

[3] B.-I. Dalenbäck, P. Svensson, and M. Kleiner: Room acoustic prediction and auralization based on an extended image source model. The Journal of the Acoustical Society of America, 92(4):2346, 1992.

[4] I. A. Drumm: The Development and Application of an Adaptive Beam Tracing Algorithm to Predict the Acoustics of Auditoria. PhD thesis, 1997.

[5] A. Farina: Ramsete - a new pyramid tracer for medium and large scale acoustic problems. In Proceedings of EURO-NOISE, 1995.

[6] 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.

[7] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. West, G. Pingali, P. Min, and A. Ngan: A beam tracing method for interactive architectural acoustics. Journal of the Acoustical Society of America, 115(2):739–756, February 2004.

[8] T. A. Funkhouser, P. Min, and I. Carlbom: Real-time acoustic modeling for distributed virtual environments. In Proc. of ACM SIGGRAPH, pages 365–374, 1999.

[9] J. Genetti and D. Gordon: Ray tracing with adaptive supersampling in object space. In Graphics Interface ’93, pages 70–77, 1993.

[10] J. Genetti and G. Williams: Adaptive supersampling in object space using pyramidal rays. Computer Graphics Forum, 17(1):29–54, 1998.

[11] P. S. Heckbert and P. Hanrahan: Beam tracing polygonal objects. In Proc. of ACM SIGGRAPH, pages 119–127, 1984.

[12] B. Kapralos, M. Jenkin, and E. Milios: Acoustic modeling utilizing an acoustic version of phonon mapping. In Proc. of IEEE Workshop on HAVE, 2004.

[13] 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.

[14] K. Kunz and R. Luebbers: The Finite Difference Time Domain for Electromagnetics. CRC Press, 1993.

[15] K. H. Kuttruff: Auralization of impulse responses modeled on the basis of ray-tracing results. Journal of Audio Engineering Society, 41(11):876–880, November 1993.

[16] C. Lauterbach, S.-E. Yoon, D. Tuft, and D. Manocha: RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs. IEEE Symposium on Interactive Ray Tracing, 2006.

[17] C. Lauterbach, A. Chandak and D. Manocha: Interactive Sound Propagation in Dynamic Scenes Using Frustum Tracing. (under review)

[18] G.R. Moore: An Approach to the Analysis of Sound in Auditoria. PhD thesis, Cambridge, UK, 1984.

[19] T. Otsuru, Y. Uchinoura, R. Tomiku, N. Okamoto, and Y. Takahashi: Basic concept, accuracy and application of large-scale finite element sound field analysis of rooms. In Proc. ICA 2004, 479-482, April 2004.

[20] A. Reshetov, A. Soupikov, and J. Hurley: Multi-level ray tracing algorithm. ACM Trans. Graph., 24(3):1176–1185, 2005.

[21] I. Wald, C. Benthin, M.Wagner, and P. Slusallek: Interactive rendering with coherent ray tracing. In Computer Graphics Forum (Proceedings of EUROGRAPHICS 2001), volume 20, pages 153–164, 2001.

[22] I. Wald, S. Boulos, and P. Shirley: Ray Tracing Deformable Scenes using Dynamic Bounding Volume Hierarchies. ACM Transactions on Graphics, 2006.

[23] I. Wald, T. Ize, A. Kensler, A. Knoll and S. Parker: Ray Tracing Animated Scenes using Coherent Grid Traversal. Proc. ACM SIGGRAPH, 2006.

[24] T. Whitted: An improved illumination model for shaded display. Commun. ACM, 23(6):343–349, 1980.