Proc. of the EAA Symposium on Auralization, Espoo, Finland, 15-17 June 2009

FAST EDGE-DIFFRACTION FOR SOUND PROPAGATION IN COMPLEX VIRTUAL ENVIRONMENTS

Micah Taylor, Anish Chandak, Zhimin Ren, Christian Lauterbach, and Dinesh Manocha

University of North Carolina

{taylormt,achandak,zren,cl,dm}@cs.unc.edu

http://gamma.cs.unc.edu/SOUND/Diffraction/

ABSTRACT

We present an algorithm for fast computation of diffraction paths for geometric-acoustics in complex environments based on the UTD formulation. Our method extends ray-frustum tracing to efficiently compute paths in the shadow region caused by long diffracting edges. Our approach can handle general scenes with moving sources, receivers, and dynamic objects. We evaluate the accuracy through comparisons with physically validated geometric simulations. In practice, our edge diffraction algorithm can perform sound propagation at nearly interactive rates in dynamic scenarios on a multi-core PC.

1. INTRODUCTION

Modeling realistic propagation of sound in virtual environments can provide important cues for user immersion. There is extensive literature on simulating the propagation of sound, including reflections and diffraction. However, prior methods do not provide sufficient flexibility and efficiency that is needed for use in interactive virtual environments.

In this paper, we primarily focus on simulating early sound propagation paths, namely edge diffraction contributions. The inclusion of paths in the diffraction region can convey important audio cues from sources that are not directly visible to the receiver. It is necessary to simulate diffraction accurately in order to obtain realistic and smooth transitions when the receiver or the source is moving.

The accurate solution of diffraction involves numerically solving the wave equation. However, the high computational requirements of numerical solvers limit their use to offline simulations. As a result, current interactive sound propagation methods are based on Geometric-Acoustic (GA) techniques. The strength of GA techniques lies in the fact that they are able to quickly calculate audio contributions arising from direct contributions and reflections. This performance benefit comes at a price in that GA techniques model only high-frequency components of direct propagation, specular reflection, and diffuse reflection [1, 2, 3, 4]. Other effects, such as diffraction, are relatively difficult to capture and can have a high computational cost [5]. There has been much research on diffraction methods that integrate well with GA solutions [5, 6, 7, 8].

Main results: We present a near-interactive approach to perform edge diffraction around long edges in complex virtual environments. Our approach extends the ray-frustum tracing method [3] to perform edge diffraction based on the Uniform Theory of Diffraction (UTD) [9]. The resulting method takes advantage of the fact that frustum tracing's discrete, sub-division based formulation provides a framework to create diffraction frusta with adaptable speed and accuracy. These new frusta are used to compute the diffraction contribution paths in the region around long edges. The overall algorithm retains the underlying advantages of frustum tracing with respect to efficiency and can handle complex, dynamic environments with moving sources or listeners.

Our algorithm has been implemented and integrated with the frustum-tracing based sound propagation method. We compare its accuracy with previous methods on the well-known Bell Labs box [10]. We also highlight its performance on complex indoor and outdoor scenes with dynamic objects, moving sources and listeners. In practice, we can accurately compute the propagation paths with diffraction and specular reflections at near-interactive rates on a multi-core PC workstation. Out method can be used to generate plausible sound rendering in complex and dynamic virtual environments.

Organization: The rest of the paper is organized in the following manner. We briefly survey prior work in sound propagation in Section 2. We present the details of our algorithm in Section 3. We then analyze its accuracy in Section 4 and highlight the performance on complex models in Section 5.

2. PREVIOUS WORK AND BACKGROUND

In this section, we give a brief overview of prior work in sound propagation and diffraction computation. Numerical methods of audio propagation provide the most accurate results, correctly modeling diffraction as well as low frequency wave effects. There are several techniques that may be used, such as boundary element methods, finite element methods, digital waveguides [11], and finite difference time domain [12]. However, even with significant performance improvements [13], these methods can take a few hours on simple scenes and are limited to static scenes.

Geometric Acoustic (GA) techniques assume that high frequency sound waves can be modeled as rays. The primary GA techniques include image-source [14], ray tracing [15], beam tracing [2], and frustum tracing [3]. Image-source methods have a very high cost when simulating higher orders of reflection and are often combined with ray tracing to reduce the computation time needed for high order reflections. Ray tracing techniques are fast, but can suffer from aliasing or sampling errors. Beam tracing is a volumetric technique and performs accurate geometric propagation, but relies on elaborate clipping algorithms and acceleration structures that are limited to static scenes. Frustum tracing attempts to strike a balance between ray tracing and beam tracing by performing discrete clipping along with volumetric tracing. This greatly

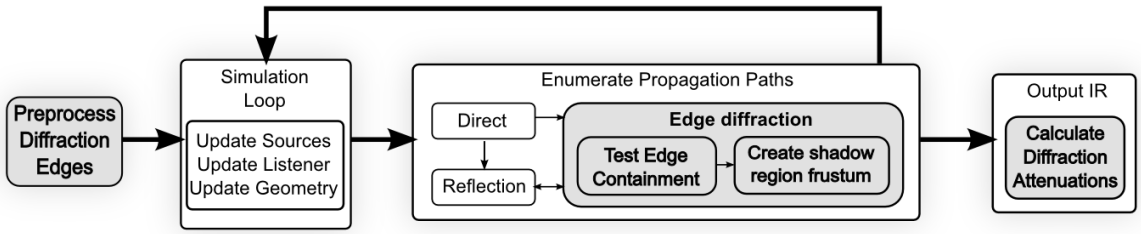


Figure 1: ***Overview of our edge diffraction algorithm:*** *Possible diffracting edges are detected and marked as a preprocess. During the simulation, frusta are checked for diffracting edge containment. If so, a new diffraction frustum is created. After the propagation is complete, the diffraction paths are attenuated by the UTD coefficients.*

reduces sampling issues and uses hierarchical acceleration structures to handle complex, dynamic scenes.

There has been much work combining diffraction with GA methods. The two primary diffraction models used in geometrical simulations are the Uniform Theory of Diffraction (UTD) [9] and the Biot-Tolstoy-Medwin (BTM) [16, 17] method. These methods are widely used since they describe the diffraction of a ray path incident on an edge. The BTM method is considered more accurate than UTD and can be formulated for use with finite edges [6]. However, the BTM method is compute intensive, which has led to different techniques to improve its performance [8, 18].

The UTD method has lower computational requirements and has been used to calculate diffraction coefficients for several interactive simulations, based on beam tracing [5] and 2D visibility diagrams [19]. In practice, these approaches have been mainly limited to static scenes. In this paper, we use the UTD as it has a much lower computational overhead and is more amenable to interactive applications. We primarily focus on diffraction contributions in the shadow region (out of line-of-sight) for speed concerns.

3. ALGORITHM

In this section, we present our algorithm in detail and address the issues that arise in terms of incorporating edge diffraction in ray-frustum tracing. For details on our approach, we refer to [3, 20]. The underlying algorithm generates 4-sided frusta based on specular reflections and edge diffractions and intersects the frusta with the scene primitives. After intersection, if a frustum is not fully contained within a scene triangle, part of the frustum must lie outside the triangle edges. In this case, the frustum is sub-divided into sub-frusta using a quad-tree structure. These sub-frusta are then intersected with the triangle and the process repeats to a user defined limit. This sub-division allows a frustum to more accurately represent the shape of the scene primitives encountered.

In order to reduce the runtime overhead of checking if a triangle edge can diffract, our algorithm precomputes all the diffraction edges in the scene as part of a preprocessing step. When a sub-frustum is found to contain a potential diffracting edge, a new frustum is created to contain the possible diffraction contributions. After all propagation paths are found, they are attenuated based on the path characterization and scene primitives. Figure 1 shows the various steps of our algorithm. While we are immediately concerned with the direct, specular, and diffraction components, our algorithm can also be combined with diffuse and late reverberation calculations for more accurate simulations [21].

3.1. Preprocess

Rather than explicitly testing whether an edge is a candidate for diffraction at runtime, we find all possible diffraction edges as part of a preprocess by analyzing the scene and marking the long edges. Specifically, we use a data structure that links each edge to its incident triangles. The edges of each triangle are classified based on the surface normals of the incident faces (see Figure 2).

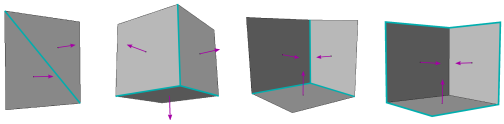


Figure 2: ***Preprocessed edge types:*** ***(a)*** *Planar edges that never diffract;* ***(b)*** *exterior edges that always diffract;* ***(c)*** *interior edges and* ***(d)*** *disconnected edges that can be configured by user choice to diffract.*

If the triangles incident to an edge have similar normals, they are considered almost planar and the resulting edge is not considered as a candidate for diffraction (see Figure 2(a)). If the normals are exterior and point away from one another, the edge is part of a diffracting wedge (Figure 2(b)). The two surface normals are used to compute the wedge angle that is later used for calculating the diffraction coefficients. There are two other remaining cases that can be marked as diffracting or non-diffracting depending on the scene design. In scenes where triangles form both the interior and exterior sides of a wall, the user can elect to have triangles with normals facing inwards (Figure 2(c)) marked as diffracting edges from the backface. Similarly, disconnected edge (i.e. triangles without neighbors, see Figure 2(d)) can be marked as diffracting edge if the user desires. If marked, each disconnected edge would have a wedge angle of 2π radians. Edges may also be marked by other general criteria, such as direct user selection or minimum and maximum length.

3.2. Edge containment

During scene traversal, it is necessary to identify the diffracting edges that are contained within a propagating frustum. Consider the case where a frustum intersects a triangle and is not fully contained within the triangle. In this case, at least one of the corners of the frustum face lies outside of the triangle edges (Figure 3(a)). After many iterations of the adaptive frustum subdivision, the subdivision limit is reached and the edge is approximated by many sub-frusta, as shown in Figure 3(b). Some of these sub-frusta must contain the edge that caused the initial subdivision.

A series of tests determine whether a diffracting edge is contained within a frustum and thus whether we need to compute a

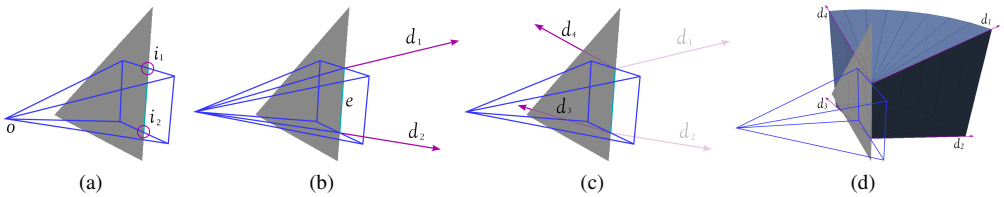


Figure 4: ***Diffraction frustum creation:*** ***(a)*** *Given a frustum's origin o and its edge intersection points i_1 and i_2,* ***(b)*** *the edge axis e and the initial diffraction vectors d_1 and d_2 are created.* ***(c)*** *Rotating d_1 and d_2 about the edge axis towards the far side of the diffracting wedge sweeps a diffraction cone in the shadow region bounded by the final vectors d_3 and d_4.* ***(d)*** *We create the complete the frustum volume.*

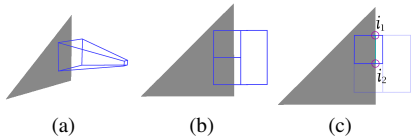


Figure 3: ***Edge containment check:*** *After the frustum encounters a triangle* ***(a)****, its face is projected into the triangle plane* ***(b)****. Each diffracting edge is then checked for intersection with the face* ***(c)*** *to find the intersection points i_1 and i_2.*

diffraction frustum. Since the preprocessing step has assigned each of the triangle's edges a type, if none are marked as a diffracting edge, the test terminates. However, if one of the edges is diffracting, it is necessary to find the portion of the edge that is exposed to the sound field. This is tested by performing intersection between the edge and the four lines that form the frustum face boundaries. The diffracting edge is checked for intersection against the boundaries of the frustum face. If the edge does not cross the bounds of the sub-frustum, the diffracting edge must not be within this sub-frustum and the test is repeated with the next sub-frustum.

In the case that the frustum boundaries intersect the diffracting edge, the exact orientation of the edge within the frustum needs to be determined. This is performed by completing the intersection calculation and finding the two intersection points i_1, i_2 (see Figure 3 (c)), of the edge on the frustum boundary. These intersection points are used in the construction of the diffraction frustum.

3.3. Diffraction frustum construction

When a diffracting edge is found within a frustum, a diffraction frustum is created and propagated through the scene. This diffraction frustum should encompass the shadow region that is hidden from the direct contribution or specular contribution. We will now detail the calculations used during frustum creation (see Figure 4). Since most diffracting edges are located at wedges formed where two triangles meet, we will differentiate between the two triangles as the *source* side and the *receiver* side of the wedge. The source side is the side that is exposed to the original propagation path; the receiver side is the side where the new diffracted field will propagate in the shadow region.

In order to create a diffraction frustum, given the diffracting edge, the region of the edge that is contained within the initial frustum must be known. The intersection points from the edge containment test describe this portion of the edge. Using these points i_1 and i_2 on the edge and the origin of the original frustum o, two new vectors d_1 and d_2 are defined as $d_1 = i_1 - o$ and $d_2 = i_2 - o$. These vectors describe the side of the new diffraction frustum that borders the transition from line-of-sight contribution to shadow contribution.

Next, we construct the vectors that are used to represent the far plane of the diffraction frustum. This far plane will border the face of the triangle on the receiver side of the diffracting edge, and combined with the first set of vectors, bounds a portion of the shadow region. We begin the computation by defining an edge axis vector $e = i_1 - i_2$. There is a vector f_r which is perpendicular to the diffracting edge and lies in the plane of the triangle that represents the receiver side of the diffracting edge. This receiver face vector is defined as $f_r = e \times n_r$, where n_r is the normal of the receiver side triangle. We also compute the vector d_{perp} by projecting d_1 onto the plane perpendicular to e. Once these vectors are computed, we find the angle between them, and rotate d_1 and d_2 about e towards f_r by this angle. Beginning at d_1 and d_2, at intervals along the rotation, new frusta are created to approximate the diffraction cone, with the rotation ending at the vectors d_3 and d_4, which lie in the plane of the triangle corresponding to the receiver side.

In order to create the full diffraction region about the edge (not just the shadow region), the vectors d_3 and d_4 can be created efficiently as:

$$d_f = \begin{cases} (d_s \cdot e)e + (d_s \cdot f_r)f_r & \text{if } d_s \cdot f_r < 0 \\ (d_s \cdot e)e - (d_s \cdot f_r)f_r & \text{otherwise} \end{cases}$$

Where d_f is the resulting vector in the plane of the receiver triangle and d_s is a vector that borders the shadow region. Similar to the example in Figure 4, d_4 results from d_1 and d_3 results from d_2. The vectors d_4 and d_3 are then rotated about e towards the triangle face on the source side of the wedge. At intervals along this swept region, new frusta are created to approximate the diffraction region.

3.4. Path generation

The frustum tracing algorithm generates new reflection and edge diffraction frusta based on the intersections with scene primitives. As each frustum traverses the scene, the data needed to attenuate its contribution is pushed on a stack. This includes the data that describes the direction and location of the frustum and the geometric primitives encountered. The reflected frusta have the material attenuation values pushed, while diffraction frusta have the wedge angle and triangle data pushed onto the stack. This data is later used to create the contribution paths used in generating an IR (Impulse Response).

As each frustum is propagated through the scene, it is checked for containment of the receiver. If the receiver is contained in the frustum, there is some reflection or diffraction path from the source to the receiver. We wish to compute the path segment found inside the receiver containing frustum as well as the segments inside each parent frustum that was propagated up to the containing frustum.

These path segments are computed from a linear combination of the four the rays that form the frustum edges. Together, these path segments represent the entire contribution path through the scene.

This is quick to compute and for specular reflection, accurately represents the contribution path. However, for diffraction, there is slight error in the path vectors since the UTD cone is being approximated by a finite number of frusta. The details of this error are outlined in Section 4.2.

3.5. Attenuation

After all the frusta are computed, the contribution paths are used to compute an impulse response for the scene. Algorithms to compute the attenuation values for reflection have been described previously [3]. In this section, we highlight the calculation of the diffraction attenuation coefficients.

The UTD was chosen as the method to generate the diffraction coefficients for the paths since it works well with the discrete intersection tests performed for ray-frustum culling. The UTD assumes that an edge has infinite length and the actual length or subset of the edge that is exposed to sound energy is not used. Specifically, a single frustum can be checked if it contains a valid diffraction path completely independently of all other frusta.

Each path is attenuated using the UTD calculations for a user defined number of frequency bands (see Figure 6). Since only paths in the shadow region are attenuated, there will be a discontinuity in the field at the boundary of the shadow region. Near this boundary, approximate attenuation values are found by normalization [5]. This allows a smooth transition at the shadow boundary.

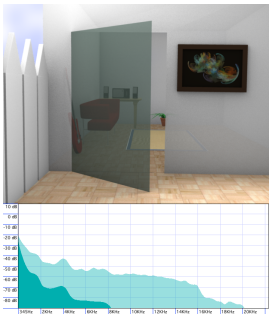


Figure 6: ***UTD attenuation:*** *A radio is playing behind the door. The light green region shows the spectrum for the direct path when the door is open. The dark green region shows the spectrum of strongest diffraction path as the door closes.*

4. ACCURACY

In this section we validate the results computed by our edge diffraction algorithm against results from the well known Bell Lab Box benchmark [10]. We compare the number of paths our system find for various subdivision levels to the number of paths that the beam tracing algorithm finds in the Bell Lab Box benchmark. In addition, we discuss the frustum tracing approximation of diffraction cones and detail the limitations of our approach.

Throughout this section and the next, we will refer to the subdivision level chosen for the frustum tracing system. Given a maximum subdivision level of x, a frustum may be recursively split up to 2^x times.

4.1. Bell Lab Box comparison

The "Bell Lab Box" is an acoustic benchmark. The Bell Lab Box is a large closed box which contains a sound source and receiver. An optional baffle may be inserted that obstructs the visibility between portions of the box. This Bell Lab Box was used to conduct a controlled study [10] of audio diffraction. Knowing the scene dimensions and layout, physical measurements were compared to a beam tracing simulation of a similar virtual scene. The resulting output compares well with the physical measurements from the Bell Lab Box.

The referenced paper provides the earliest 60 geometric paths traced from a source position to a receiver out of line-of-sight. We match the path sequences from their highly accurate geometric simulation to sequences generated by our diffraction frustum propagation. For these comparisons, we compute frusta for the full diffraction region about the edge as predicted by the UTD, instead of using the shadow region approximation.

In the Bell Lab Box path data there are a large number of paths that encounter the diffracting edge, and depart the edge traveling parallel along the diffracting wedge. Due to its basis on ray tracing methods, frustum tracing may not find these paths that are parallel to and travel along the plane of the diffracting wedge. Consider a frustum with corner rays that travel parallel to the wedge face plane. Even if there are more diffracting edges in this plane, they will not be found, since such edges are not contained within the frustum shape. This is similar to intersecting a ray with a triangle oriented such that the triangle normal is perpendicular to the ray direction.

We compare the number of paths found by each simulation in Figure 7. As the subdivision level of frustum tracing increases, more paths are found and the accuracy increases. For clarity, we include results from a hypothetical frustum tracing simulator that correctly calculates all paths except those that are parallel to the diffracting surface. As shown in the figure, the number of paths found by high subdivision frustum tracing compares very favorably with this ideal frustum tracer.

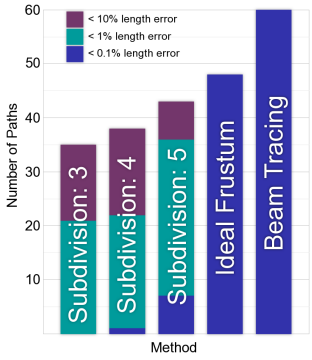
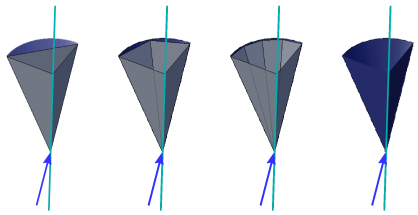


Figure 7: ***Path length:*** *As subdivision level increases, more paths are found and the error decreases.*

4.2. Accuracy of diffraction frustum

Frustum tracing is an approximate method and can achieve high update rates by reducing simulation accuracy. Conversely, higher accuracy can be achieved by reducing the simulation update rate. The creation of diffraction frusta follows this same property. Diffraction frusta are initially subdivided based on the subdivision level chosen by the user. Since each diffraction frustum is bounded

above and below by an approximate diffraction cone, it is helpful to evaluate the difference in volume between a subdivided approximation and a perfect diffracting cone. Figure 8 and Table 1 shows that as the subdivision level increases, the diffraction frustum quickly converges to the ideal volume.



(a) (b) (c) (d)

Figure 8: ***Frustum Subdivision accuracy:*** *the resulting diffraction cone with a subdivision of 0* ***(a)****, subdivision of 1* ***(b)****, and subdivision of 2* ***(c)****. The diffraction frustum approximates the ideal cone* ***(d)****.*

Subdivision level	0	1	2	cone
Volume error	36.34%	9.97%	2.55%	0.0%

Table 1: ***Volume error:*** *As the subdivision level increases, the error in the volume of the diffraction cone decreases*

4.3. Limitations

Since the UTD is used to calculate the diffraction coefficients, the underlying restrictions of the UTD model naturally apply to our algorithm. UTD is a high frequency approximation, and is not very accurate for low frequency diffraction. Moreover, UTD assumes that the diffracting edge is of infinite length and the source and receiver are far from the edge (relative to wavelength). These restrictions are discussed in further detail in [9].

Frustum tracing can be regarded as an approximation of beam tracing and introduces some additional limitations. While it is a volumetric technique, some paths are missed since the frusta cannot be subdivided infinitely to represent the scene primitives. The level of subdivision is controlled either using a uniform global parameter [3] or an adaptive subdivision scheme [20]. This error can be avoided entirely by extending frustum tracing to compute accurate object-space visibility and perform accurate geometric acoustics [22].

Since our method approximates the diffraction cone with many linear frusta, this subdivision can result in over-estimation or under-estimation in the final frustum volume. This results in the back projected path having a slightly over or under estimation of the path length. As discussed in 4.2, this error is reduced as the subdivision limit is increased. The error may be eliminated entirely by computing conservative diffraction frusta [23].

Due to the approximation of computing only shadow region diffraction, a discontinuity exists at the shadow boundary. While this can certainly be resolved by computing the entire diffraction region exactly, this greatly increases the number of frusta that must be propagated. As such, we resolve the discontinuity by using approximate normalized attenuation values.

As previously mentioned, frusta may have difficulty finding paths that lie parallel to the corner ray of the frustum. This may cause some important contribution paths to be unaccounted for. These paths could be found by including impostor structures at edges such that impostor will be encountered during propagation. Other potential approaches would be 2d ray intersection of the scene primitives in the plane of the wedge or propagating a special frustum along the plane that contains the region of the plane.

5. PERFORMANCE

In this section we evaluate the performance of our method with various scenes and settings. Unless otherwise noted, all simulations are rendered with 8 frequency bands and 4 threads on a modern 2.66 Ghz multi-core machine. The scenes used are detailed in Table 2. We use a maximum subdivision level of 3 for all benchmarks. Also, unlike in the Bell Lab Box comparison, we only propagate diffraction frusta in the shadow region.

Scene	Type	Triangles	Diffracting edges
Q3dm1	Closed	14k	4032
Atrocity	Closed	12k	1531
Chartres	Open	192k	40489
Sibinek	Open	76k	1358
Sponza	Open	66k	1021
Highway	Open	350k	1248
Sodahall	Closed	1510k	9457

Table 2: ***Scene overview:*** *Data on the scenes used for the performance results. Some scenes are very open with much geometry visible from any given point. Others are closed, with short visibility distances.*

5.1. Diffraction cost and benefit

Generating the diffraction frusta during simulation incurs a time cost in addition to direct contributions and specular reflections. The time cost needed to propagate the diffraction frusta varies greatly from scene to scene due to the number of triangles and edges encountered. The benefit of using diffraction also varies; depending on the scene layout and source/receiver position, very few valid diffraction paths may be found. Table 3 shows the added cost and benefit of using diffraction in various scenes with 3 orders of recursion. To highlight the effects of diffraction, in each scene we chose the source and receiver positions such that few specular paths are found. For example, in the highway scene, the receiver is placed behind an occluder that blocks all direct and specular contributions.

6. CONCLUSION AND FUTURE WORK

The presented edge diffraction method enhances frustum tracing by allowing diffraction contribution paths that can be computed and auralized. Our resulting system computes direct contributions, specular reflections and edge diffraction using ray-frustum tracing. To the best of our knowledge, this is the first edge diffraction method that results in near-interactive performance in complex scenes with dynamic objects with reasonable GA accuracy. The overall performance of the system increases as the order of reflections and diffraction increases. We have observed comparable results with the beam tracing method on the Bell Labs Box, and our algorithm can generate plausible acoustic simulation results on complex benchmarks.

Scene	Diffraction	# Frusta	Time	Paths found
Q3dm1	Off	80844	219 ms	3
	On	114372	338 ms	5
Atrocity	Off	114183	282 ms	4
	On	140454	370 ms	7
Chartres	Off	219865	1306 ms	2
	On	292256	2078 ms	8
Sibinek	Off	370594	1614 ms	12
	On	377521	1636 ms	15
Sponza	Off	198022	861 ms	2
	On	209737	921 ms	11
Highway	Off	21178	62 ms	0
	On	23553	84 ms	5
Sodahall	Off	81269	436 ms	0
	On	91879	510 ms	3

Table 3: ***Diffraction benefit:*** *Diffraction incurs a slight performance decrease, but often finds more propagation paths.*

There are many avenues for future work. Stronger validation of the found diffraction paths could be conducted to reduce or eliminate the slight path error. Computing more conservative diffraction frusta would reduce the possibility of missing important paths. Conducting a conservative region visibility test from each encountered diffracting edge would make it possible to find paths suitable for auralization with the BTM (Biot-Tolstoy-Medwin) method. While this may have a large computational cost, it would allow more accurate simulation of lower frequencies and shorter diffracting edges. Another way to improve the performance is to reduce the number of diffraction frusta generated. This would allow more time to perform higher levels of subdivision or reflections. There has been some work in the area of diffraction culling [8]. This would reduce the number of insignificant frusta created.

Acknowledgements:

We would like to thank Paul Calamia for his many helpful suggestions. This work was supported in part by ARO contracts DAAD19-02-1-0390 and W911NF-04-1-0088, NSF awards 0400134, 0429583 and 0404088, DARPA/RDECOM contract N61339-04-C-0043, and Intel.

7. REFERENCES

[1] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, April 1979.

[2] 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 *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1998, pp. 21–32, ACM.

[3] C. Lauterbach, A. Chandak, and D. Manocha, "Interactive sound propagation in dynamic scenes using frustum tracing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1672–1679, 2007.

[4] M. Hodgson, "Evidence of diffuse surface reflections in rooms," *The Journal of the Acoustical Society of America*, vol. 89, no. 2, pp. 765–771, February 1991.

[5] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, "Modeling acoustics in virtual environments using the uniform theory of diffraction," in *SIGGRAPH 2001, Computer Graphics Proceedings*, 2001, pp. 545–552.

[6] U. P. Svensson, R. I. Fred, and J. Vanderkooy, "An analytic secondary source model of edge diffraction impulse responses ," *Acoustical Society of America Journal*, vol. 106, pp. 2331–2344, Nov. 1999.

[7] V. Pulkki, T. Lokki, and L. Savioja, "Implementation and visualization of edge diffraction with image-source method," in *The 112th Audio Engineering Society (AES) Convention*, 2002, pp. preprint no. 5603+.

[8] P. T. Calamia and U. P. Svensson, "Edge subdivision for fast diffraction calculations," in *Proc. 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005)*, October 2005, pp. 187–190.

[9] R. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *IEEE, Proceedings, vol. 62, Nov. 1974, p. 1448-1461.*, 1974.

[10] N. Tsingos, I. Carlbom, G. Elbo, R. Kubli, and T. Funkhouser, "Validation of acoustical simulations in the "bell labs box"," *IEEE Computer Graphics and Applications*, vol. 22, no. 4, pp. 28–37, June 2002.

[11] S. Van Duyne and J. O. Smith, "The 2-d digital waveguide mesh," in *Applications of Signal Processing to Audio and Acoustics, 1993. Final Program and Paper Summaries., 1993 IEEE Workshop on*, 1993, pp. 177–180.

[12] D. Botteldooren, "Acoustical finite-difference time-domain simulation in a quasi-cartesian grid," *The Journal of the Acoustical Society of America*, vol. 95, no. 5, pp. 2313–2319, 1994.

[13] N. Raghuvanshi, N. Galoppo, and M. C. Lin, "Accelerated wave-based acoustics simulation," in *Symposium on Solid and Physical Modeling*, 2008, pp. 91–102.

[14] J. Borish, "Extension of the image model to arbitrary polyhedra," *The Journal of the Acoustical Society of America*, vol. 75, pp. 1827–1836, June 1984.

[15] A. Krokstad, S. Strom, and S. Sørsdal, "Calculating the acoustical room response by the use of a ray tracing technique," in *Journal of Sound and Vibration*, 1968, vol. 8, pp. 118–125.

[16] M. A. Biot and I. Tolstoy, "Formulation of wave propagation in infinite media by normal coordinates with an application to diffraction," *Journal of the Acoustical Society of America*, vol. 29, no. 3, pp. 381–391, March 1957.

[17] H. Medwin, "Shadowing by finite noise barriers," *Journal of the Acoustical Society of America*, vol. 69, no. 4, pp. 1060–1064, April 1981.

[18] P. T. Calamia and U. P. Svensson, "Fast time-domain edge-diffraction calculations for interactive acoustic simulations," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 186–186, 2007.

[19] 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)*, September 2004, pp. 1773–1776.

[20] 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*, vol. 14, no. 6, pp. 1707–1722, December 2008.

[21] M. Taylor, A. Chandak, L. Antani, and D. Manocha, "Resound: Interactive sound rendering for dynamic virtual environments," Tech. Rep., University of Chapel Hill, 2009.

[22] A. Chandak, L. Antani, M. Taylor, and D. Manocha, "Fastv: From-point visibility culling on complex models," Tech. Rep., University of Chapel Hill, 2009.

[23] M. Taylor and D. Manocha, "Fast accurate diffraction paths with frustum tracing," Tech. Rep., University of Chapel Hill, 2009.