High-Oder Diffraction and Diffuse Reflections for Interactive Sound Propagation in Large Environments (Supplementary Material)

Carl Schissler[*] Ravish Mehra[†] Dinesh Manocha[‡]
University of North Carolina at Chapel Hill
http://gamma.cs.unc.edu/HIGHDIFF/

In this supplementary document, we present details related to our algorithm and implementation. These include

- Detailed results and analysis of our algorithms.
- Simplification details and results.
- Comparing the performance and accuracy of our algorithm with prior techniques.

1 Diffuse Reflection Computation

The state of the art algorithms for diffuse reflections are based on path tracing and we compare the accuracy of our approach with these methods. We provide an additional comparison in the Sibenik Cathedral scene for a moving listener and single static source. Figure 1 shows how our diffuse algorithm computes the sound intensity at the listener to be very close to the result for naive path tracing with 10x as many rays. The path tracing results with the same number of rays (1000) are significantly more noisy than our approach, especially in scenarios where the source and listener are occluded from each other or the number of rays that intersect the listener is low. We average the contributions of many previous frames to compute a better estimate of the sound intensity than is possible with a naive approach and the same number of rays emitted from the source.

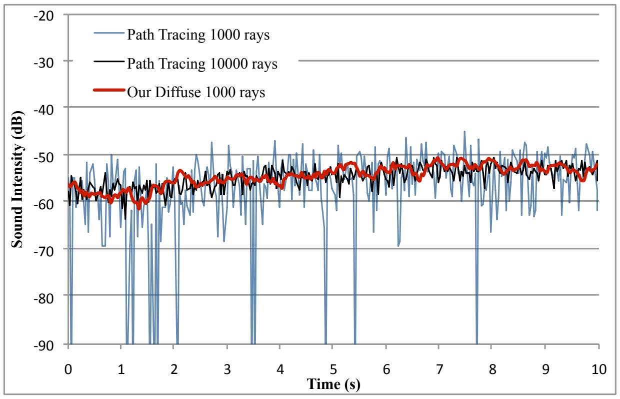

Figure 1: *This comparison shows how our diffuse algorithm closely matches the results of path tracing (in black) with 10x more rays in the Sibenik Cathedral scene (79,942 triangles). The noise in the blue line is due to insufficient sampling and produces audible discontinuities. Our approach (in red) avoids these frame-to-frame discontinuities by taking into account the results of previous frames to increase the ray sample size.*

In figure 2, we examine the effect on convergence of varying the averaging window, τ, in the office scene with 1000 diffuse rays per frame. We measure the error in our approach for different values of τ when compared to the ground truth of brute-force path tracing with 20k rays. With $\tau = 0$, we observe similar results to naive path tracing with 1000 rays because no diffuse path caching occurs. As τ increases, the accuracy of our approach increases due to a larger averaging window that uses the ray contributions from more frames. However, our approach may introduce some error for large values of τ in dynamic scenes where there are abrupt changes in the sound intensity received at the listener. Figure 3 shows the time-domain smearing of the sound energy due to the averaging effect. When τ is large, the resulting audio is very smooth and free of sampling noise, but at the expense of quick reaction to changes in the scene. We found experimentally that $\tau = 300ms$ produced the best balance between utilizing ray coherence and time-domain accuracy.

We also examined the effect of varying the size of l, the surface patch resolution. Figure 4 shows that there is no significant correlation in the error caused by our method for different values of l. All subdivision sizes tested (0.02m to 4.0m) produced a similar improvement over naive path tracing for the same number of rays traced. In Figure 5, we observe no change in the response time or accuracy due to varying values of l when compared to brute-force path tracing. However, very large values of l (e.g. ≥ 1 m) may result in significant error in the delay time or the listener-relative direction for propagation paths. Because bigger subdivisions group together more rays that are incoherent and may have different path lengths, the time-domain accuracy or directionality of propagation paths may be affected. These errors are generally imperceptible for diffuse sound because individual propagation paths cannot be distinguished.

2 Wavelength Dependent Simplification

A key component of our simplification algorithm is computation of candidate diffraction edges. These edges are used to precompute the visibility graph and used at runtime for computing diffraction computations. We use the following heuristic to compute a set of candidate diffraction edges. The main idea is to compute edge with a significant deviation from being planar. If two triangles T_1, T_2 that share an edge have normals n_1, n_2 and the value of $|n_1 \cdot n_2| < \cos(\beta)$ for some angle β, then the edge shared by T_1, T_2 is not planar and is placed in the set of candidate diffraction edges.We check each edge in the set to see if has any neighboring candidate edges that are close to being collinear with the original edge. Our chosen metric is to compute the absolute value of the dot product of the edge directions. If this value is close to one, the edges are considered collinear and are merged into a single diffraction edge, as shown in Fig. 6. This process proceeds until no more edges are found that are collinear with respect to the starting edge for the search. Edges that are merged into the starting edge are marked as inactive and are not considered in subsequent steps. The final result is a reduced set of diffraction edges for the model that can be used for geometric acoustics.

Figure 7 shows the results of simplification for different resolutions. For large voxels, significant errors may be introduced in the simplified diffraction edges. On the other hand, using too small a voxel resolution will negate the benefits of simplification because small details are not removed.

[*] schissle@cs.unc.edu
[†] ravish.mehra07@gmail.com
[‡] dm@cs.unc.edu

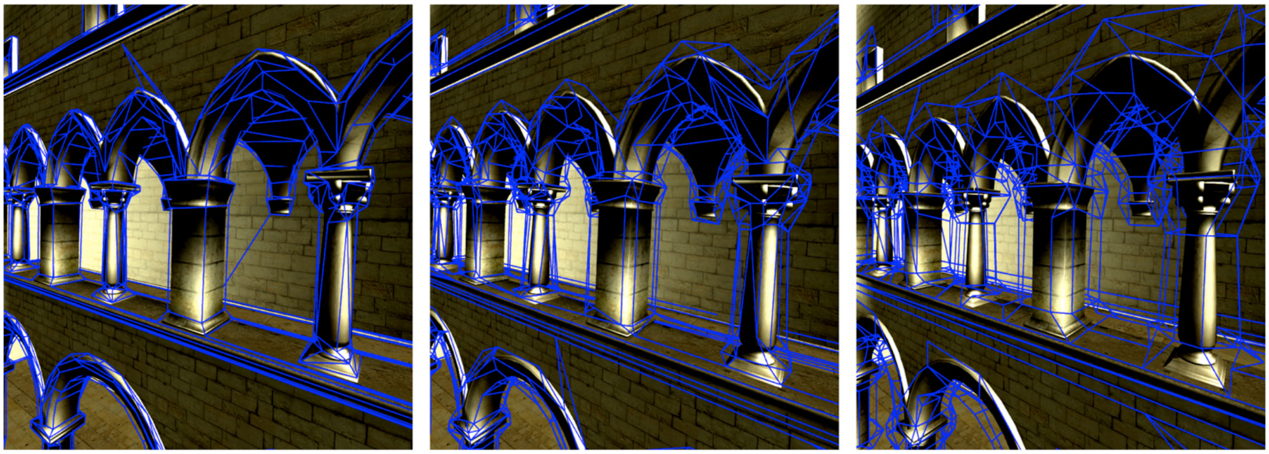

Figure 7: *Simplification results for 3 different wavelengths in a test scene (Sponza Atrium). Simplified diffraction edges are shown in blue. Small details in the original mesh are removed by performing the voxelization.*

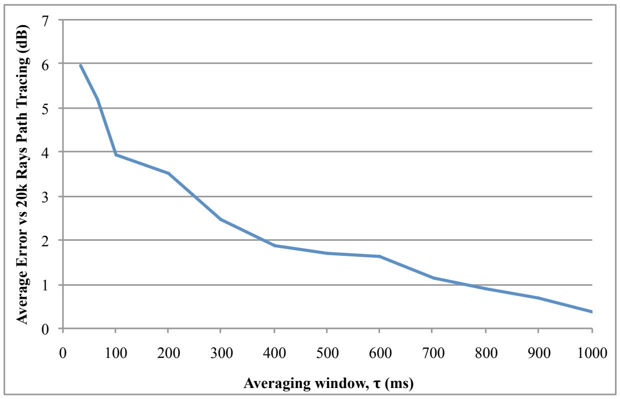

Figure 2: *By using a longer averaging window, τ, we demonstrate that our diffuse approach converges to a small error when compared to brute-force path tracing.*

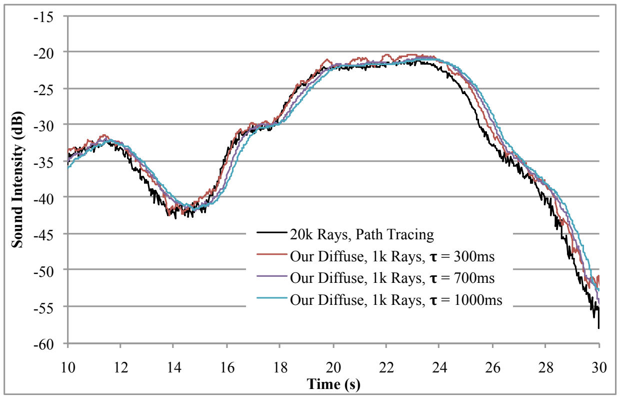

Figure 3: *The choice of τ determines how quickly the simulation will react to a change in the scene configuration. For increasing τ, we notice that there is a reaction delay of approximately τ versus brute-force path tracing. Changing τ has insignificant impact on the runtime performance of our approach.*

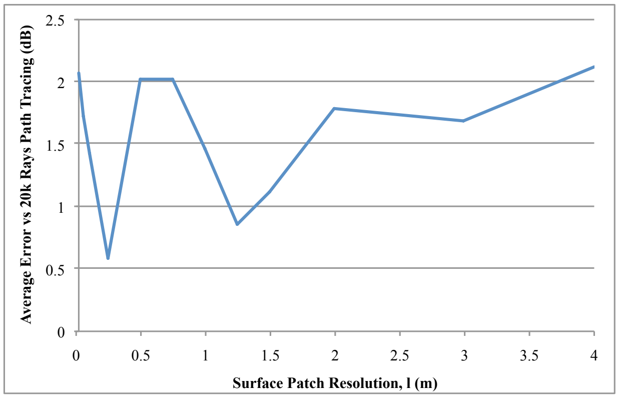

Figure 4: *When varying the surface patch resolution l, we found no significant change in the accuracy of the sound intensity received at the listener.*

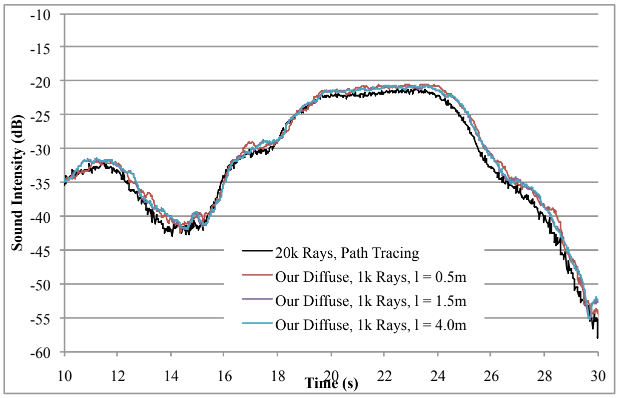

Figure 5: *This graph shows the sound intensity received at the listener for different values of l. Even large values of l produce no significant changes in the accuracy or response time of our algorithm.*

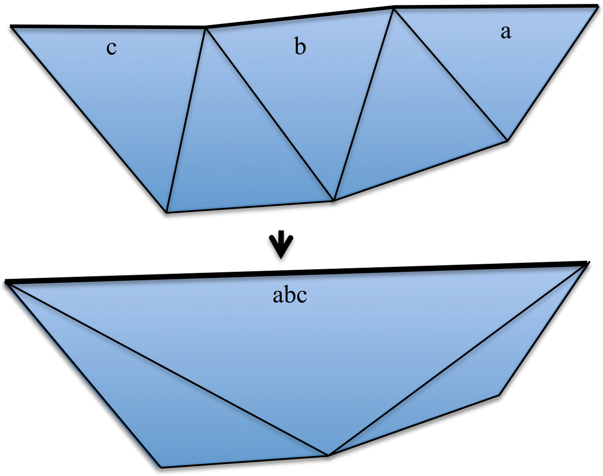

Figure 6: *Starting with a candidate diffraction edge $\overline{\mathrm{a}}$, our edge-merging algorithm inspects neighboring edge $\overline{\mathrm{b}}$ and finds it to be nearly collinear with edge $\overline{\mathrm{a}}$ based on the the value of $\mathrm{d} = \left|\frac{\vec{a}}{|\vec{a}|} \cdot \frac{\vec{b}}{|\vec{b}|}\right|$. If $\gamma \leq \mathrm{d} \leq 1$ for some γ, then the two edges are considered to be collinear. The algorithm then inspects the edge neighbors of $\overline{\mathrm{b}}$ and finds that, edge $\overline{\mathrm{c}}$ is also collinear with edge $\overline{\mathrm{a}}$. All three edges are then merged into the single diffraction edge $\overline{abc}$.*