

Diffraction Kernels for Interactive Sound Propagation in Dynamic Environments

Atul Rungta, Carl Schissler, Nicholas Rewkowski, Ravish Mehra, and Dinesh Manocha *Fellow, IEEE*

URL: <http://gamma.cs.unc.edu/diffractionkernel>



Fig. 1: We highlight the performance of our algorithm to generate diffraction effects from smooth objects in these complex, dynamic scenes. We precompute diffraction kernels using wave-based methods and they are coupled with interactive ray tracing-based propagation algorithms. The overall algorithm can simulate diffraction effects, along with specular and diffuse reflections, in complex dynamic scenes and results in improved acoustic effects.

Abstract—We present a novel method to generate plausible diffraction effects for interactive sound propagation in dynamic scenes. Our approach precomputes a diffraction kernel for each dynamic object in the scene and combines them with interactive ray tracing algorithms at runtime. A diffraction kernel encapsulates the sound interaction behavior of individual objects in the free field and we present a new source placement algorithm to significantly accelerate the precomputation. Our overall propagation algorithm can handle highly-tessellated or smooth objects undergoing rigid motion. We have evaluated our algorithm’s performance on different scenarios with multiple moving objects and demonstrate the benefits over prior interactive geometric sound propagation methods. We also performed a user study to evaluate the perceived smoothness of the diffracted field and found that the auditory perception using our approach is comparable to that of a wave-based sound propagation method.

Index Terms—sound propagation, diffraction, dynamic environments, spatial presence

1 INTRODUCTION

Research in virtual environments over the last few decades has demonstrated that improved sound simulation and rendering can significantly augment a user’s sense of presence [13]. Sound can induce a sense of “object presence” and “spatial presence” at the same time, raising the fidelity of VR and AR simulations [6]. As the characteristics of the environment or source locations vary in real time, it is important to perform interactive auralization that accurately captures any changes caused by the user or the environment, and to generate smoothly rendered audio.

The most accurate algorithms for sound simulation are based on directly solving the acoustic wave equation using numerical methods and compute the pressure field. Recently, different precomputation-based solvers have been proposed to compute an acoustic kernel, which is used at runtime for interactive propagation for dynamic sources or listeners [9, 19, 38]. However, these techniques have two major limitations: (a) the precomputation time and the memory overhead can be very high requiring large compute clusters; (b) they are limited to static scenes and cannot handle dynamic objects, a common scenario

in virtual environments.

Most interactive algorithms for sound simulation and rendering for dynamic scenes are based on geometric acoustics and ray tracing [11, 14, 26, 31]. Recent ray tracing algorithms can handle a high number of sources and compute higher order reflections at interactive rates on commodity desktop processors [2]. It is well-known that pure geometric-acoustics techniques work well for high frequencies, and can’t model low-frequency wave effects such as diffraction or occlusion. In practice, it is important to model these wave effects to correct the spectral content of reflected sound from finite surfaces, such as overhead reflectors or wall edges. Diffraction becomes very important for listeners located inside the shadow zones of obstacles and the inaccurate modeling of these effects can lead to a loss of realism in VR [23].

There is significant literature on augmenting the geometric acoustic techniques with diffraction approximation. Most prior techniques are designed to model edge diffraction [26, 31, 34], which includes propagating sound around the corners as well as scattering sound in all directions from wedges of any angle. However, current interactive diffraction algorithms have some limitations. First, they are less accurate for highly tessellated objects or smooth surfaces and can result in discontinuous sound field in occlusion scenarios. Second, it is computationally challenging to handle highly tessellated objects because the computational complexity increases exponentially with the number of diffracting edges.

Main Results: We present a novel approach based on object-based diffraction kernels to model sound propagation in dynamic environments. Our hybrid formulation combines the accuracy benefits of wave-based computation with the efficiency and flexibility of geometric ray-tracing methods. The resulting approach can handle virtual environments composed of highly tessellated, dynamic objects at interactive

• Atul Rungta, Carl Schissler, and Dinesh Manocha are with the Department of Computer Science, University of North Carolina, Chapel Hill. E-mail: {rungta,schissle,dm}@cs.unc.com.

• Nicholas Rewkowski is with the Department of Computer Science, University of North Carolina, Chapel Hill. Email: nrewkows@live.unc.edu

• Ravish Mehra is with Oculus Research. E-mail: ravish.mehra@oculus.com

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

rates and offers these benefits:

- Efficient source-placement algorithm that significantly reduces the precomputation time by reducing the required number of wave-based simulations needed to compute the diffraction kernel of an object.
- Handles highly tessellated or smooth objects while modeling diffraction and occlusion effects.
- Efficient runtime based on ray tracing with minimal overhead enabling interactive performance for dynamic scenes.

In the preprocessing stage, we efficiently compute the *diffraction kernels* that encapsulate the sound interaction behavior of individual objects in *free field*. These kernels capture all the interactions of sound waves with the objects, including reflections, diffraction, scattering and interference, and we present a novel source placement algorithm for efficient computation. Our algorithm exploits the symmetric properties of the scattering field and the object shape to compute diffraction kernels at only a few incoming directions and accelerates the precomputation by 1 – 2 orders of magnitude (Section 3) for efficient desktop computation.

We present a new coupling algorithm that integrates these diffraction kernels with interactive ray tracing at runtime. Our modified ray tracing algorithm uses the object-based diffraction kernels to approximate the wave effects such as diffraction and combines them with standard geometric ray tracing techniques to compute reflections at interactive rates (Section 4). We demonstrate the interactive performance on many dynamic scenes with smooth, highly tessellated objects undergoing rigid motion (Section 5). We highlight improved accuracy as compared to prior interactive, geometric methods for capturing diffraction effects and also compare the performance with wave-based solvers (Section 6). We also perform a perceptual evaluation using a user study to compare the auditory perception of our algorithm with a wave-based propagation algorithm (Section 7).

2 RELATED WORK

In this section, we give a brief overview of prior work on sound propagation.

2.1 Wave-Based Methods

These methods are the most accurate way of simulating sound propagation as they solve the acoustic wave equation directly. Some of the frequency domain solvers include methods based on the finite-element method (FEM), boundary-element method (BEM), and the time domain solvers include methods such as finite-difference time domain (FDTD) and adaptive rectangular decomposition (ARD). However, their space and time complexity increases as a third or fourth power of frequencies. Many interactive propagation techniques have been proposed for static scenes that precompute an acoustic kernels and use them to compute the impulse responses at runtime as a function of the source or listener positions. Equivalent source method based techniques have been used to precompute the acoustic radiation characteristics of rigid objects [9] or the per-object and inter-object transfer functions for sound propagation [16, 17] and can also be combined with ray tracing algorithms [38]. However, the computational overhead of these methods is very high and they require large compute clusters for pre-computations. Furthermore, none of these methods can handle dynamic objects in the scene owing to need to recompute the total field if the objects in the scene move (i.e., the inter-object transfer function computation would change), thereby limiting their application to static scenes. [17] can handle moving sources and listeners but not moving objects while [16] can handle either a moving source or a moving listener. In contrast, our method has a lower computational overhead and can handle dynamic source, listeners, and objects, though our accuracy is slightly lower. Raghuvanshi et al. [19, 20] use the adaptive rectangular decomposition method to precompute acoustic responses on a sampled spatial grid.

2.2 Geometric Acoustics and Diffraction

Geometric techniques model the acoustic effects based on ray theory and typically work well for high-frequency sounds to model specular and diffuse reflections [24]. Wave phenomena such as diffraction must be modeled explicitly or separately and prior methods are limited to edge diffraction. The Biot-Tolstoy-Medwin (BTM) model is an accurate time-domain diffraction formulation that evaluates an integral of diffracted sound along finite rigid edges, can be extended to higher-order diffraction, and can be combined with wave-based methods [21, 30]. However, it is expensive to evaluate for complex scenes and limited to offline computations. An alternative approach, the uniform theory of diffraction (UTD), is a less accurate frequency-domain model of diffraction for infinite edges that can generate plausible results for interactive simulation in certain scenarios [31, 34]. The complexity of these edge-based diffraction techniques can increase exponentially with the maximum diffraction order, since each edge in the scene can interact with every other edge. To reduce the cost of visibility testing for high-order UTD diffraction, a precomputed edge-to-edge visibility graph can be used for static scenes, but current interactive systems are limited to low orders of edge-diffraction [26]. However, it is not clear whether techniques based on UTD can handle complex (highly tessellated) models that are frequently used in gaming and VR due to the high number of potential diffraction edges [33]. An accurate sound particle model of edge diffraction based on the Heisenberg uncertainty principle has been proposed for high-order diffraction [29], but is not robust for complex objects.

2.3 Hybrid Methods

Given the relative benefits of wave-based and geometric methods, hybrid techniques have been proposed to combine them. These include methods based on spectral decomposition of the low frequencies (i.e., less than 1kHz or 2kHz) are modeled using wave-based solvers, such as FDTD or FEM, and the high frequencies are modeled using ray tracing or beam tracing [7, 15, 28]. The computational complexity of these hybrid approaches is dominated by the wave-based methods that are performed over the entire acoustic domain. Another set of hybrid algorithms performs a spatial decomposition of the simulation domain into near-object regions and far-field regions for precomputation [38]. It uses an equivalent source formulation to compute the per-object and inter-object transfer functions, and combines that with a geometric ray tracing method to handle higher frequencies. However, the computation of per-object and inter-object transfer is expensive and also requires a large compute cluster for precomputation. As with [16, 17], moving objects in the scene would require recomputing the inter-object transfer functions making this method limited to static scenes with either a moving source or a moving listener. Furthermore, different coupling techniques have been proposed to combine the results at the interfaces, based on BEM [8], FDTD [37], FEM [1], and ESM [38]. However, none of these approaches can handle dynamic scenes at interactive rates.

3 ACOUSTIC FIELD & DIFFRACTION KERNEL

In this section, we present a high-level overview of our sound propagation algorithm based on diffraction kernels. Figure 2 shows the overall pipeline of our approach divided into two distinct stages: precomputation to compute the diffraction kernels and runtime based on interactive ray tracing. Table 1 gives a list of all the symbols used.

3.1 Acoustic wave equation

The acoustic wave-equation models the scattering behavior of objects. In spherical coordinates the wave-equation can be expressed as:

$$\nabla^2 p = \underbrace{\frac{\partial^2 p}{\partial r^2} + \frac{2}{r} \frac{\partial p}{\partial r}}_{\text{radial part}} + \underbrace{\frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial p}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 p}{\partial \phi^2}}_{\text{angular part}} \quad (1)$$

where p is the pressure and (r, θ, ϕ) correspond to \mathbf{x} in spherical coordinates. The complete solution to the equation above can be expressed

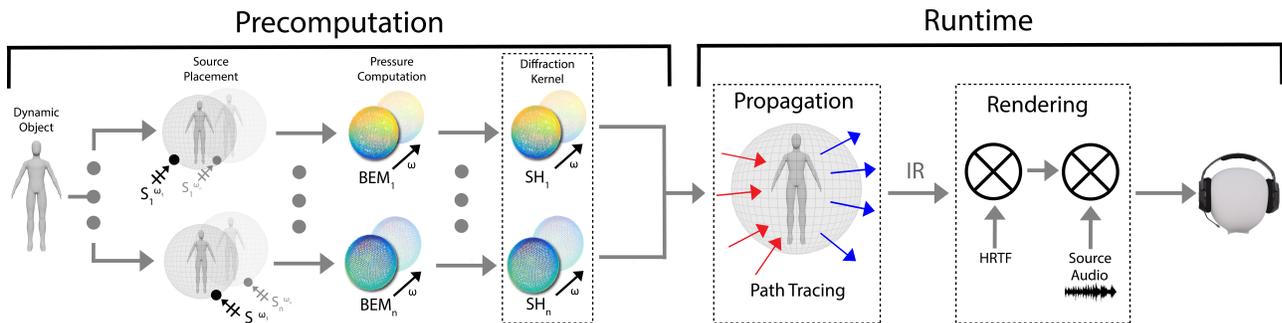


Fig. 2: **Interactive Sound Propagation and Rendering:** We highlight different stages of our novel sound propagation and rendering pipeline, which uses per-object diffraction kernels. In the precomputation stage, we adaptively perform BEM simulations for certain directions (computed using our novel source placement algorithm) and measure the outgoing pressure fields produced by the scattering of plane waves at various frequencies. These pressure fields encode the scattering as a function of frequency, the input and output directions and converted into an efficient spherical harmonic representation called the diffraction kernel. At runtime, the diffraction kernel is coupled with an interactive path tracing algorithm to simulate sound propagation and auralization in dynamic scenes.

Symbols	Meaning
\mathbf{x}	Incident direction
\mathbf{y}	Outgoing direction
ω	Frequency
$d(\mathbf{y}, \omega)$	Scattered pressure field
$\tilde{d}(\mathbf{y}, \omega)$	SH representation of $d(\mathbf{y})$
$D(\mathbf{x}, \mathbf{y}, \omega)$	Diffraction kernel
$P(\mathbf{y})$	Probability density function
I_i	Incident sound intensity
I_o	Outgoing sound intensity
A^{proj}	Projected area
H	Visible-curvature histogram
SS	Shape signature

Table 1: A table of important mathematical symbols used in the text.

in terms of the radial and angular parts:

$$\psi_{lm}(\mathbf{x}, \mathbf{y}) = \Gamma_{lm} \underbrace{h_l^2(kr)}_{\text{radial}} \underbrace{Y_l^m(\mathbf{x} - \mathbf{y})}_{\text{angular}} \quad (2)$$

The angular part of the solution is described using spherical harmonics Y_l^m while variation in the pressure because of distance is controlled by the Hankel function h_l^2 .

3.2 Diffraction Kernels

We use a diffraction kernel representation to capture the angular portion of the solution while the radial variation of pressure is approximated by a geometric sound propagation technique. We consider a spherical grid of incoming directions and generate plane-waves from each direction of this grid. For each plane wave, we compute the scattered field for the object on an offset surface of the object using a wave-based method. The angular portion of this scattered field is expressed using the *diffraction kernel* in a compact spherical harmonic basis. With the angular scattering behavior of an object computed for all the plane wave directions and frequencies, we use a geometric sound propagation method to handle the radial portion thus approximating the solution to the wave-equation.

Our diffraction kernel encapsulates the sound field interactions of the object and maps the incoming sound field reaching the object to outgoing, diffracted field emanating from the object. In contrast to the per-object transfer function [16], the diffraction kernel formulation is defined in the far-field of the object. This can significantly reduce the precomputation overhead and makes it easier to integrate with interactive ray tracing. Mathematically, the incoming pressure field in the far-field can be expressed in the plane-wave basis whereas the outgoing sound field in the far-field is expressed using spherical harmonic basis, as shown in Equation 1.

Scene Classification Our approach is based on computing the diffraction kernel for each objects in the scene. As a preprocess, we classify the scene in terms of the object type. The scene is first classified into *static* and *dynamic (moving)* objects. Static objects typically include walls, buildings, and other typically large, immovable objects in the scene. Dynamic objects can include cars, humans, chairs, and doors, all of which can potentially undergo rigid motion in the environment. Our approach is designed to capture the scattering behavior of dynamic objects, while the static environment is handled by other sound propagation techniques.

3.3 Source Placement

Diffraction is a direction dependent phenomenon and in order to capture the variations in the diffracted field, we need to capture the sound interaction behavior of an object from all possible directions. This can be naively computed by constructing a densely sampled sphere around the object and evaluating the diffracted field for each vertex on the sphere. However, such a method would incur a large precomputation cost because the wave-based solvers typically used to compute the scattering are slow and the complexity increases as a function of the geometric tessellation and maximum frequency. In order to reduce the pre-computation overhead, we present a novel source placement algorithm that exploits the *acoustic scattering invariance* of a dynamic object to reduce the number of sources we need to place on the sphere to capture its scattering behavior from all incident angles. Our source placement algorithm is used in the first stage of our precomputation pipeline and computes the representative source positions, as shown in Figure 2.

3.3.1 Visual Symmetry vs. Scattering Field Symmetry

The goal of our source placement tends to exploit the symmetry in the acoustic scattering field of each object, and thereby compute a few representative source positions. One possibility is to exploit the visual or shape symmetry of each object. There is extensive work on symmetry detection in computer vision and geometry processing [18], which are used to compute a representation of their Euclidean symmetries. However, the criteria used in these methods are not sufficient for detecting the symmetry in the acoustic scattering field of an object. For example, there are objects that exhibit little or no shape symmetry, but still exhibit symmetry in their acoustic scattering field. As a result, our goal is to develop an approach that generalizes the notion of shape-similarity and is not sensitive to the small variations in the viewed-geometry. One of the metrics in our source placement algorithm is to use projected areas to overcome these issues.

3.3.2 Multi-stage Algorithm

Next, we describe various stages of our source placement algorithm including mesh simplification, computing the projected area for each

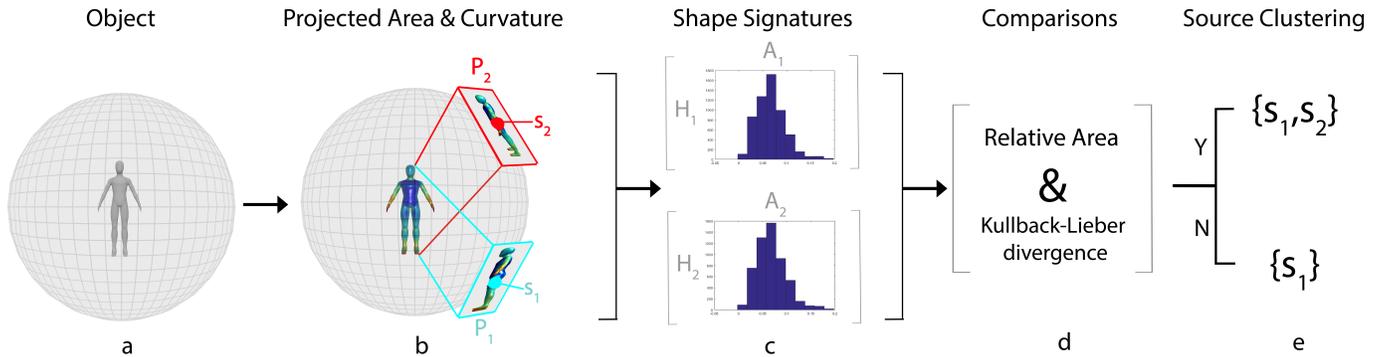


Fig. 3: **Overview of our source placement algorithm:** We use a novel source placement algorithm to compute the representative source positions for each object: (a) Given a scatterer (human), we consider a densely sampled sphere around it; (b) For each point s_i on the sphere the projected area and viewed curvatures are computed; (c) The curvature values are binned into histograms H_i and together with the projected area A_i give a shape signature SS_i at s_i . (d) These shape signatures are used to compute geometric similarity between different viewpoints. (e) Points are grouped together if their shape signatures are within error thresholds ϵ_A and ϵ_H . The overall algorithm results in 1 – 2 order of magnitude improvement in the precomputation stage.

incident direction and identifying the shape and diffraction field from each direction. We compare the view-dependent shape information to compute the geometric invariance among various incident directions and clump them together.

Projected Area: The diffraction field is a strong function of the shape and orientation of the object. In particular, for convex objects it has been shown that diffraction is a function of the projected area of the object [3, 36]. Formally,

$$P_{sc}(\mathbf{x}) = KA^{proj}(\mathbf{x})P_{in}, \quad (3)$$

where $A^{proj}(\mathbf{x})$ is the projected area at \mathbf{x} , P_{in} is the incident field, P_{sc} is the scattered field (or diffraction field), and K is a constant. We exploit this dependence of the scattered field on the projected area and extend it to arbitrary or non-convex objects by augmenting the projected area with curvature histograms (described below) to uniquely identify the *shape signature*.

Shape Signature: Our source placement algorithm initially considers a densely-sampled list of possible source positions S on a sphere. For a point $s_i \in S, \forall i \in \{1..|S|\}$, we compute an orthographic projection matrix $P(\mathbf{x}_i)$ and compute the projection of the vertices of the object ($v \in V$), whose normals N_j^v satisfy the $N_j^v \cdot \mathbf{s}_i > 0$. Next, we construct a boundary B_i using the *alpha* or α -shape of v and compute the area enclosed by the boundary A_i^{proj} . α -shape is the generalization of the notion of a convex-hull of a point set M , with $\alpha \rightarrow 0$ giving us M , while $\alpha \rightarrow \infty$ giving us the convex-hull of M . At the end of this step, we have computed the projected area of the object for each point on S .

In practice, the projected area alone cannot be used as a unique signature of the viewed shape and may result in false positives, in terms of classifying rather different shapes as similar. Therefore, we augment our metric by using the curvature of the object to define the *shape signature* of the object for each s_i . This view-dependent shape signature encapsulates the intrinsic characteristics of the shape when viewed from different source points (or incident angles). We use well-known techniques [4] to compute the principal curvatures κ_1 and κ_2 for the scatterer, and compute them for each v . Instead of using κ_1 & κ_2 separately, we consider them as $\kappa_{v_j} = |\kappa_{1,v_j}| + |\kappa_{2,v_j}|$ and bin them in a histogram H_i that uses N bins. The bin values range between the minimum and maximum values of $|\kappa_1| + |\kappa_2|$. Using the projected area and curvature, we get a shape signature (SS_i) for each s_i :

$$SS_i = \begin{pmatrix} A_i^{proj} \\ H_i \end{pmatrix} \quad (4)$$

Rotational symmetry: After computing the shape signature, we iterate over the points in S . Starting with a point s_i , we compare its shape

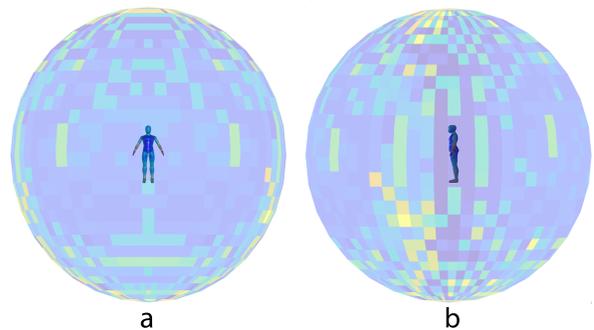


Fig. 4: **Similar Shape Signatures:** We highlight different points that have similar shape signatures. Each set of points with the same color on the sphere corresponds to a set that is computed as geometrically invariant and will be represented using a single sound source. (a) The source placement automatically detects symmetry in the model which is bilateral in this case; (b) shows another viewpoint and since no symmetry exists in that plane, the two hemispheres have different colors.

signature SS_i with every other point's shape signature SS_j by computing the relative difference in the projected area A_i^{proj} and A_j^{proj} , and we also compute the difference in histogram H_i and H_j using the Kullback-Liebr divergence:

$$D_{KL}(H_i||H_j) = \sum_k H_i(k) \log\left(\frac{H_i(k)}{H_j(k)}\right) \quad (5)$$

This metric gives us a measure of the mutual information contained in two shapes. A D_{KL} value of zero indicates that the shapes are similar and would likely have similar scattering properties. On the other hand, a value of one would indicate that the shapes are very dis-similar. Using appropriate thresholds for the relative projected areas and D_{KL} , we cluster the points that fall within the threshold bounds with respect to s_i . The threshold values are used to strike a balance between the number of sources that are selected from S and the error in computing the total scattering function of an object. Finally, we choose one representative point in each cluster and use that point as the source position for which the scattering function is computed using a wave-based solver. The scattering functions for other points in the cluster are extrapolated from this representative point.

Reflection Symmetry: Many objects used in the real world exhibit reflective symmetry (e.g., a pillar). Although our algorithm can recognize some sort of symmetry in the object, it cannot identify the nature of

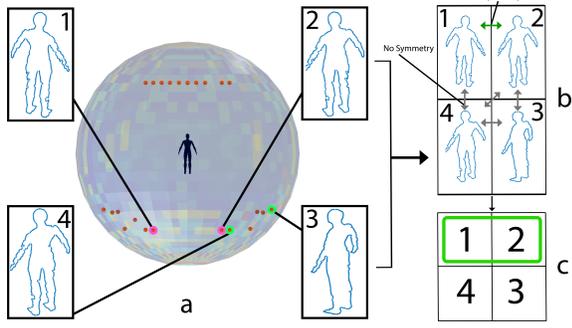


Fig. 5: **Reflection symmetry detection:** Given a cluster of points with similar shape signatures (a) we perform a pair-wise comparison of the boundaries and compute Hausdorff distance; (b) Boundaries B_1 and B_2 that nearly overlap after being reflected result in a large drop of their Hausdorff distances, while B_3 and B_4 do not exhibit reflection symmetry with each other or with B_1 or B_2 ' (c) A relative change in the distance indicates reflection symmetry between (B_1 and B_2).

that symmetry. The previous steps detect the invariance in geometry which includes rotational symmetry along with insignificant changes in shape with the change in incident angle. In order to explicitly identify the reflection symmetries in a cluster, we perform a pair-wise comparison between the points in the cluster (Fig. 5). For each such pair of points, we compute the silhouette of the object from these points and compute the Hausdorff distance between these boundaries. The Hausdorff distance (d_H) between two non-empty sets (X, Y) that are subsets of a metric space (M, d) is given by:

$$d_H(X, Y) = \max_{x \in X} \min_{y \in Y} \{d(x, y)\}, \quad (6)$$

where $d(X, Y)$ is some measure of distance in M (L_2 metric in our case). We reflect these 2D boundaries either along X or Y axis depending on the source positions and compare their Hausdorff distances. If the relative Hausdorff distance is below our threshold, we consider these boundaries as reflections of each other. (Fig. 5(b,c)). In case an object exhibits both rotational and reflection symmetry at the same point, our method automatically considers them to be rotationally symmetric.

3.4 Diffraction Kernel Computation

After the reflection symmetry test, we compute the set $RP(S) = \{\{A_1^{Proj}, H_1\}, \{A_2^{Proj}, H_2\}, \dots, \{A_n^{Proj}, H_n\}\}$, where each element of RP is the set of points with a similar projected area (A_i^{Proj}) and curvatures (H_i). In terms of diffraction kernel computation, we perform a single wave-based simulation for such a set, as explained below. Overall, our algorithm performs $O(n)$ wave simulations, where $n = |RP|$ with $n \leq |S|$, to capture the diffracted field from all incident directions. In practice, n is orders of magnitude smaller than $|S|$ (Table 2). After these n simulations, we extrapolate the field within a particular set by rotating and/or reflecting the computed field, thereby giving us the complete diffracted field for all $s_i \in S$.

For an incoming plane wave coming from direction \mathbf{x} , the outgoing sound field $d(\mathbf{y}, \omega)$ is computed using state-of-the-art wave-based methods (e.g. BEM) on a spherical offset surface in far-field. This outgoing field $d(\mathbf{y}, \omega)$ can be expressed in the spherical harmonic basis using least-squares fitting:

$$d(\mathbf{y}, \omega) \approx \tilde{d}(\mathbf{y}, \omega) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^l Y_l^m(\mathbf{y}) c_l^m(\omega) \quad (7)$$

where $d(\mathbf{y}, \omega)$ is the outgoing sound field computed using the wave-based solver, l_{max} is the spherical harmonic order, and $c_l^m(\omega)$ are the basis function coefficients as a function of frequency. This process is repeated for all the incoming plane wave directions for all the frequencies. We use a rectangular subdivision in spherical coordinates to

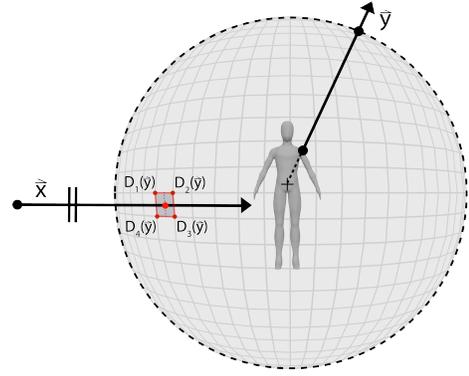


Fig. 6: We highlight how the diffraction kernel $D(\mathbf{x}, \mathbf{y}, \omega)$ can be integrated into Monte Carlo path tracing using two-way coupling. When an incoming ray with direction \mathbf{x} strikes a diffracting object, the ray is scattered in a randomly chosen direction \mathbf{y} with probability density function $P(\mathbf{y})$. The diffraction kernel in the direction \mathbf{y} is evaluated at the four corners of the quad intersected by \mathbf{x} , and the resulting pressures $D(\mathbf{x}_1, \mathbf{y}, \omega)$, $D(\mathbf{x}_2, \mathbf{y}, \omega)$, $D(\mathbf{x}_3, \mathbf{y}, \omega)$, and $D(\mathbf{x}_4, \mathbf{y}, \omega)$ are bilinearly interpolated according to \mathbf{x} to yield the pressure transfer function $D(\mathbf{x}, \mathbf{y}, \omega)$. The energy carried by the ray is then multiplied by $\frac{D(\mathbf{x}, \mathbf{y}, \omega)^2}{P(\mathbf{y})}$ to get the output ray energy.

compute the possible incoming plane wave directions. This enables efficient bi-linear interpolation of the outgoing field for any arbitrary incoming direction during the runtime stage of our pipeline.

4 INTERACTIVE RAY TRACING WITH DIFFRACTION KERNELS

In this section, we present our diffraction kernel-based technique for object-based sound propagation in dynamic scenes. We utilize a pre-computed *diffraction kernel* to model sound interactions for complex objects and couple it with a Monte Carlo path tracing framework to compute sound propagation for the entire scene.

For the simulation of diffuse reflections, many variants of Monte Carlo path tracing have been proposed that simulate the propagation of sound energy by rays in frequency bands [11]. These include backward ray tracing for multisource scenes [25], and bidirectional path tracing [2], which can also be accelerated by exploiting temporal coherence. Our approach extends these methods by developing new interactive techniques for two coupling between the rays and diffraction kernels.

The interactive ray tracing uses a bounding volume hierarchy to accelerate ray intersections. These hierarchies are updated using refitting algorithms, as the dynamic objects undergo rigid motion. After the diffraction kernel D of a particular object is computed according to Section 4, it can be used within any Monte Carlo path tracing sound propagation algorithm to efficiently compute diffracted sound for the object. This kernel information is stored in the bounding volume hierarchy nodes associated with those dynamic objects. Our formulation treats the diffraction kernel using a mathematical framework similar to surface scattering modeled using bidirectional scattering distribution functions (BSDF), which is widely used in visual rendering. BSDFs describe the distribution of sound energy as a function of frequency and the input and output direction of sound transport [5]. We use the diffraction kernel in a similar way to model the wave scattering induced by objects in all directions. Our modified path tracing algorithm uses the diffraction kernel information to compute the new paths using D for each ray, after it hits a dynamic object.

4.1 Coupling between ray and diffraction kernels

Our propagation algorithm exploits a two-way coupling between D that are computed using BEM (i.e., wave-based method) and path tracing (i.e., geometric acoustics). For the case of a single ray with input sound intensity I_i and direction \mathbf{x} , the outgoing sound intensity I_o is given by a spherical integral of the diffraction kernel over the outgoing direction

\mathbf{y} :

$$I_o(\mathbf{x}, \omega) = \int_S I_i D(\mathbf{x}, \mathbf{y}, \omega)^2 dS. \quad (8)$$

Monte Carlo techniques are a simple way to numerically evaluate integrals of this form as a weighted sum of many random samples [11]. As the number of samples approaches ∞ , the expected value of the integral converges to the exact value. The outgoing scattered intensity can be approximated by a Monte Carlo estimator:

$$I_o(\mathbf{x}, \omega) \approx \frac{1}{N} \sum_{j=1}^N I_i \frac{D(\mathbf{x}, \mathbf{y}_j, \omega)^2}{P(\mathbf{y}_j)} \quad (9)$$

where N is the number of samples, \mathbf{y}_j are the samples, and $P(\mathbf{y}_j)$ is the probability of generating sample \mathbf{y}_j . If a uniform sampling strategy is used, $P(\mathbf{y}_j) = \frac{1}{4\pi}$. This formulation can be easily integrated with any Monte Carlo ray tracer to compute object-based scattering. We utilize this formulation to model the diffraction effects and approximate the sound field in the regions that are occluded from each source.

In traditional forward path tracing, N random rays are emitted from the surface of a sound source in the scene with energy $\frac{1}{N}$. These rays are then propagated through the environment until they strike a surface, where the rays are scattered and attenuated according to the sound material BSDF. The rays may undergo many interactions with the geometry before either exiting the scene, reaching a maximum interaction order or propagation time [25], or being eliminated via Russian Roulette [10]. If a ray hits the listener, the ray’s intensity at various frequency bands is accumulated to the impulse response at the appropriate delay time. At each interaction, a shadow ray can also be traced to the listener’s position to find additional propagation paths. This is known as next-event estimation or *diffuse rain* [27]. This procedure can also be conducted in reverse by emitting rays from the listener [25], or by emitting rays from both source and listener [2]. Figure 6 demonstrates how the diffraction kernel can be integrated into this path tracing framework. When a ray hits an object in the scene that has an associated precomputed diffraction kernel, we scatter the ray using the precomputed scattering function rather than the usual BSDF. This is performed by randomly sampling the outgoing ray direction \mathbf{y} according to probability density function $P(\mathbf{y})$. The diffraction kernel is evaluated at \mathbf{y} for the four precomputed scattering functions that are closest to the incident ray direction \mathbf{x} , then the evaluated pressure is bilinearly interpolated. The energy carried by the outgoing ray is then given by:

$$I_o(\mathbf{x}, \omega) \approx I_i \frac{D(\mathbf{x}, \mathbf{y}, \omega)^2}{P(\mathbf{y})}. \quad (10)$$

When many rays hit the scattering object, the integral of the outgoing energy over all rays converges to the exact solution.

5 IMPLEMENTATION & RESULTS

In this section, we discuss our implementation and highlight the results on complex benchmarks.

5.1 Performance and Comparisons

The preprocessing algorithm has been implemented using MATLAB. We used available MATLAB code for computing the curvatures of our objects. FastBEM is used as the boundary element method solver. The runtime interactive ray-tracer is based on the geometric sound propagation algorithm described in [26] and written in C++. We do not use the original UTD-based method proposed in [26], and rather use the coupled algorithm described in Section 4.2.1 for path tracing with diffraction kernels.

Precomputation: Table 2 gives the geometric details of the objects used in our scenes and the performance of our source placement algorithm. Since BEM computation can be expensive and increases as cubic function of the frequency, our novel source placement algorithm makes it possible to handle complex, smooth objects with thousands of triangles. We observe 8 – 137X speedups due to our source placement

algorithm. That enables us to perform the diffraction kernel precomputations on a desktop PC, as opposed to using a large compute cluster. Most prior wave-based methods [16, 20, 38] have significantly higher memory and computational requirements.

Runtime System: Our interactive sound propagation algorithm has been integrated with the Unreal Engine and used to evaluate the performance of complex, dynamic benchmarks shown in Fig. 1. All the timings were generated on a multi-core desktop PC CPU. The overall system with integrated visual and sound rendering runs at 60Hz or more, as shown in the video. The additional overhead of handling diffraction kernels is very small and the overall performance is comparable to UTD-based interactive propagation algorithms [26, 34].

Please refer to the supplementary video to see the capabilities of our system for the benchmarks described below.

5.2 Benchmarks

We have evaluated our approach on various scenarios to highlight the performance of our diffraction kernels in challenging environments (Table 3). To accentuate the effect of diffraction kernels, we turn off reflections in each of our benchmarks.

Concert: The concert scene shows the effectiveness of our diffraction kernels in handling complex diffracting objects such as humans in an open environment. Each human model is represented using 11K triangles and prior interactive UTD-based methods can’t handle such scenes for plausible diffraction effects. The listener moves among a crowd of people attending a concert and ducks to pick up a dropped phone. The complex interactions of the sound source and human bodies are efficiently and plausibly calculated using diffraction kernels.

City Block: This scene shows the listener moving through a modern metropolis with various high-rise buildings. A helicopter flying over the city goes behind one of the high-rises (cylindrical) causing the sound to diffract around highly-tessellated objects. This scene demonstrates the ability of our method to handle highly-tessellated, curved objects and generate a smooth diffraction field around them. This results in smooth audio rendering.

Parking Garage: This benchmark consists of a typical parking garage with multiple pillars and cars. We use the pillars in the garage and a moving ambulance as the diffracting objects. The listener moves through the garage experiencing diffraction effects as the pillars obstruct the line-of-sight between the listener and various sources. Then an ambulance comes into the garage to park and acts as a dynamic diffraction object.

Oculus® First Contact: This benchmark is a modified version of the famous First Contact demonstration that is being shipped by Oculus®, along with their HMD. In this scenario, a playful robot acts as the object and comes in between the sound source and the listener and creates diffraction effects dynamically due to no line-of-sight. The 3D printer in the scene generates an interactive object that also results in diffraction effects along with a static monitor. Our approach can model the diffraction effects due to these dynamic objects and generate smooth audio rendering effects. We highlight these benefits in the video, by only playing the diffracted sound with no reflections.

Multi-player Game: We showcase the efficiency of our approach in this multi-player networked game. In this scenario, two players play against each other in a networked environment and are trying to shoot at each other. As the players move around, the sound gets diffracted around different objects in the scene. As a result, simulating object-based diffraction is important to simulate a continuous sound field. We highlight these benefits in the video, by only playing the diffracted sound with no reflections.

6 ANALYSIS

In this section, we analyze the various steps of our pipeline and highlight the approximations and possible sources of error in the computations. We also compare the accuracy of our precomputation algorithm with a wave-based solver (BEM) to evaluate the numeric accuracy of the computed sound pressure field. There are three main sources of error in our pipeline: Error in source placement, error introduced by

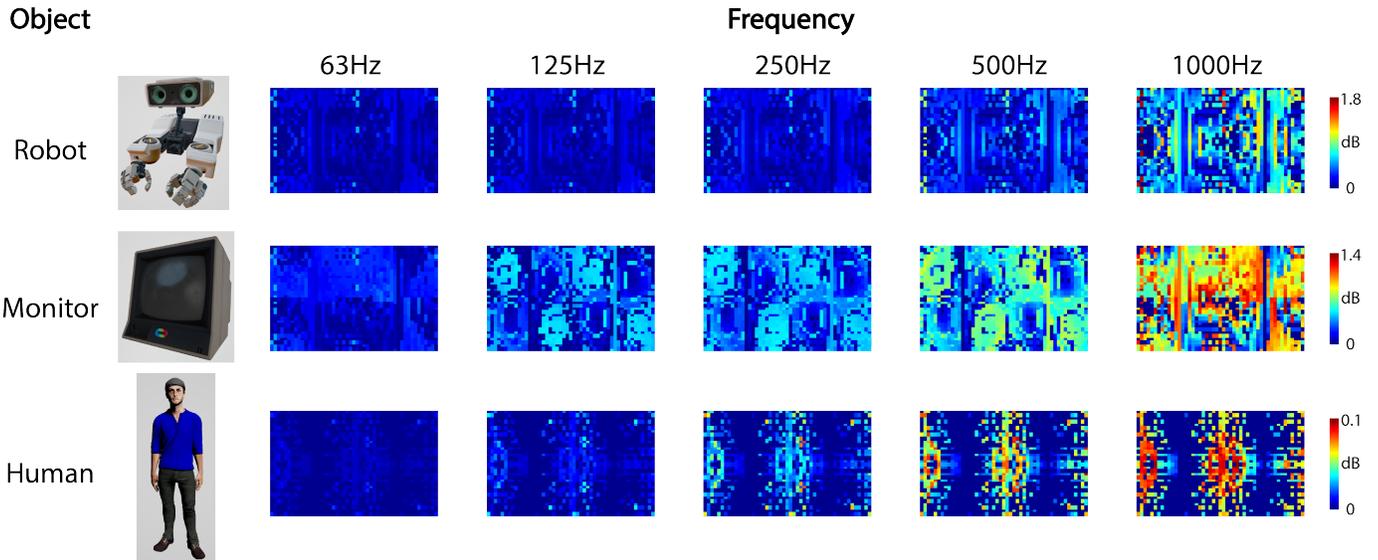


Fig. 7: The plot shows the heat map error introduced by our source placement algorithm for three different objects. The error is computed on a sphere representing all the incoming directions for the diffraction (S). Given the source positions (RP) computed by our source placement algorithm, we run BEM at these points and interpolate the field for the rest of the points in S using reflection and/or rotation. The plots here show the MAE at each point on S by unwrapping it on to a 2D plane. The horizontal axis represents the latitude while the vertical represents the longitude. As can be seen, even for complex objects at high frequencies, the error introduced by our source placement algorithm is < 2 dB.

Object	#Vert.	Size(m)	freq(Hz)	RP	Speedup
Ellipsoid	10242	2	1000	50	38X
Ambulance	21746	3.9	1000	150	13X
Human	11250	1.8	2000	254	8X
Column	29954	4.7	1000	118	16X
Tower	44168	15	500	14	137X
Ball	2562	0.5	2000	1	1922X
Monitor	3650	0.46	2000	99	19X
Robot	23971	0.44	1000	229	8X
Pillar	25746	3	500	221	8X
Planter	11114	2.78	500	323	6X

Table 2: **Diffraction Kernel Computations:** The table highlights the geometric complexity, size of objects (meter), maximum frequency, running times for computing the diffraction kernel of different objects. The value of $|S|$ is 1922 in all the benchmarks. The speedups obtained using our source placement algorithm are highlighted in the last column.

the band-limited diffraction kernel, and error incurred as a result of Monte-Carlo ray-tracing at runtime.

6.1 Source Placement

The source placement algorithm introduces errors due to simplification and metrics used to detect rotational and reflection symmetry. This error is also governed by the underlying mesh representation and the initial choice of source positions on the sphere. This could result in changes or errors in the final pressure field that is computed using BEM using those source cluster positions. This error is more at the higher frequencies, because the diffracted component of the sound field is a lot more "focused", as compared to that at the lower frequencies, and is sensitive to spatial variation. In our benchmarks, we limit the maximum frequency to 2kHz.

Fig. 7 shows the error introduced by our source placement algorithm expressed as mean absolute error (MAE) in dB. MAE is computed as:

$$MAE = \frac{\sum_{i=1}^n |P(i)_{computed} - P(i)_{ref}|}{n} \quad (11)$$

where $P_{computed}$ is the interpolated field computed by our algorithm at an incoming source direction and P_{ref} is the reference pressure at the

Scene	#Vert.	#D	PreC(Hr)	Runtime(ms)
Concert	10242	11	4	53
City-Block	21746	2	1	101
Parking-Garage	11250	4	7	115
First-Contact	29954	1	2	43
Game	44168	2	6	84

Table 3: **Runtime Performance Analysis:** We highlight the performance of our interactive sound propagation algorithm on a desktop multi-core PC. We highlight the number of diffraction objects (D-objects), precomputation time (PreC) in hours and the average frame time (ms) on a multi-core desktop CPU. Our algorithm can perform interactive sound propagation in dynamic scenes with specular and diffuse reflections and diffraction effects.

source direction. n is the number of the points on which the scattered pressure is computed.

As can be seen in Fig. 7, the error introduced for complex objects such as human and robot is below 2 dB even at frequencies as high as 1 kHz.

6.2 Diffraction Kernel

Diffraction Kernels represent the BEM pressure computed on a sphere based on a spherical harmonic basis (Eq. 7). Theoretically, spherical harmonics can fully represent a spherical function with $L_{max} \rightarrow \infty$, but in practice they have to be band-limited for practical reasons. This introduces an error given by $\epsilon_d = d(\mathbf{y}, \omega) - \tilde{d}(\mathbf{y}, \omega)$ in our diffraction kernels (Fig. 8). We highlight how this error increases in the diffraction kernel with increasing spherical harmonic order. In our current implementation, we use 9th order spherical harmonics and they generate plausible sound effects in our benchmarks.

6.3 Monte-Carlo Sampling

We show the plot (Fig. 9) of the pressure field generated by a densely sampling of the diffraction kernel and compare to the pressure field generated by BEM for the human models. This comparison highlights the numerical accuracy of the sound pressure that is approximated using the diffraction kernels. We use dense ray sampling to generate this plot for the diffraction field of an object. In this case, each point on the grid is used to trace a ray backwards from that position towards the

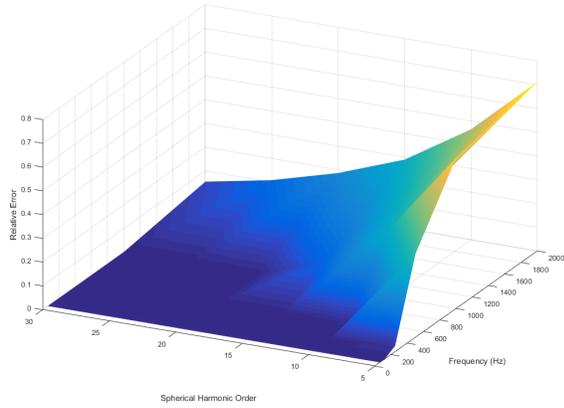


Fig. 8: The plot shows the variation of the relative error when trying to represent a diffracted field of a human in a spherical harmonic basis. As can be seen, the error increases sharply with frequency and low SH-order but stays close to zero with high order spherical harmonic. High-order spherical harmonics are more expensive to evaluate and tend to be numerically unstable

object. This ray is filtered by the diffraction kernel, depending on the angle of incidence γ and used to compute the pressure at that position. This process is akin to Monte-Carlo sampling in the limit with a very high sampling density. As mentioned in Section 4.2, Monte-Carlo path tracing methods converge to the value of the sampled function as the number of samples approach infinity. As shown in the figure, the diffraction kernels converge to the BEM computed pressure field for different frequencies. This indicates the accuracy of our diffraction kernel based method is governed by the underlying sampling criterion used in path tracing.

7 PERCEPTUAL EVALUATION

We performed a user study to evaluate the perceptual efficacy of diffraction-kernel-based sound propagation algorithm. Our study is based on the psycho-acoustic evaluation of numeric and geometric sound propagation algorithms [22, 23]. In particular, that study compared UTD-based interactive sound propagation algorithm with a wave-based sound propagation algorithm by evaluating the diffracted sound field around an obstacle by placing the subjects along a semi-circle. The study [23] demonstrated that auditory perception improves due to wave-based sound propagation and the computed diffracted field decays nearly linearly with an increasing diffraction angle. On the other hand, the diffracted field computed using UTD-based diffraction exhibited an erratic behavior. Given the known benefits of wave-based sound propagation algorithms, we perform a 2-way comparison between the diffracted sound fields computed using diffraction kernels and BEM based sound propagation.

7.1 Participants

Fourteen subjects participated in this study with informed consent. The ages ranged from 23 to 28 (Mean = 25.7 with SD = 3.22). The participants were recruited at a university campus. All participants reported normal hearing.

7.2 Apparatus

The setup consisted of a Dell T7600 workstation with the sound delivered through a pair of Beyerdynamic DT990 PRO headphones. The subjects wore a blindfold.

7.3 Stimuli

As in [23], the source was a ringing bell that was low-pass filtered with a cut-off frequency of 300 Hz, so that the diffraction effects are prominent. The sound source was placed 2m from the origin. The

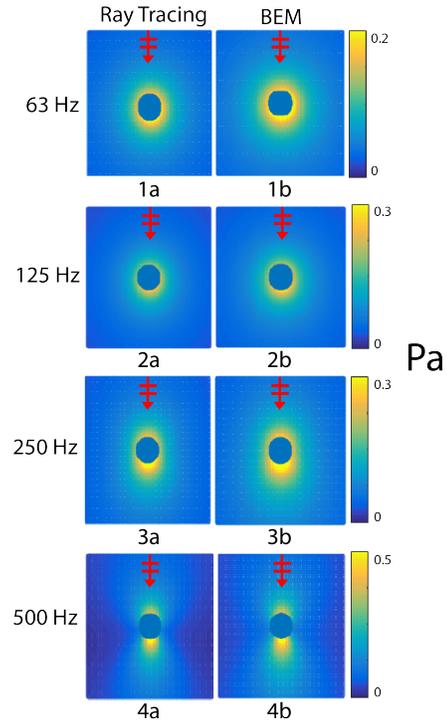


Fig. 9: We compare the sound pressure field for an object (human) computed using our modified ray tracing algorithm (Section 4) vs. BEM (wave-based solver). We perform a dense ray sampling using the diffraction kernel to compute the pressure field in the left figures. The red arrows indicate the incident direction of the plane-wave. We use 100×100 grid to sample at each point and filter them through the diffraction kernel to compute the angular variation in the diffracted sound field. The pressure values are in Pascals and we demonstrate the results for two different frequencies, where diffraction effects are prominent. These benchmarks show a close match between the sound fields computed using our method vs. BEM. In practice, our approach can perform these computations at interactive rates, where BEM solver can take minutes.

subjects were placed at 5 equispaced positions along an arc with a radius 3.5m of from the origin as shown in Fig. 10. The resulting sound was prerecorded at each of these 5 positions and two diffraction methods (diffraction kernel and BEM) and stored. On each trial the subject was randomly placed at one these 5 positions with the diffraction method randomized, too.

7.4 Design and Procedure

This was a within-subject study with the subjects wearing a blindfold. The audio was delivered through headphones and rendered monaurally. Before starting the experiments, the source sound clip was played to familiarize the subjects with it. A $1.2m \times 1m \times 4m$ column served as the diffraction object for the experiment.

The scene was open to make sure no reflections interfere with the experiments. The subjects were placed in the ‘shadow-zone’ of the diffracting object (Fig. 10) which is a region where the source is not in the line-of-sight and only the diffracted sound can reach the listener.

A total of 14 participants took part in each group. For each of the 5 positions, the subjects were asked to rate the loudness of the sound heard. The loudness was rated on an arbitrary, non-physical scale ranging from 1 – 20. The scale was explained to the subjects before the start of the experiment: the extrema of our scale was relative to a verbal standard with 1 corresponding to a very quiet sound such as that of a falling leaf, while 20 was a loud sound akin to someone shouting in one’s ears. It should be noted that loudness perception was not the focus of our experiment; rather, the smoothness of change in perceived loudness across spatial variations as measure of the quality of each

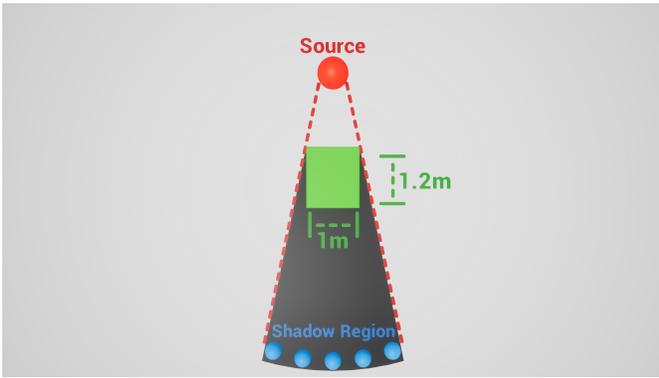


Fig. 10: The figure shows the setup used for the user study that compared the psychoacoustic characteristic of our diffraction kernel based algorithm with BEM-based wave propagation algorithm. We considered 5 equi-spaced points in the shadow region (black) of the obstacle (green). The obstacle is a column and only the diffracted sounds are audible in the shadow region. We evaluated the auditory perception using diffraction kernel and BEM-based sound propagation.

diffraction method. The loudness of the sounds for the two diffraction methods was level-matched by matching the root mean square (rms) of the sounds generated by the two methods at a reference position in the line-of-sight of the source.

A block consisted of 10 (5 positions \times 2 diffraction methods) trials with three blocks per subject giving a total of 30 (5 positions \times 2 diffraction methods \times 3 blocks) readings. The subjects were placed randomly at one of the 5 positions with the sound played through two diffraction methods which were chosen randomly, as well. The subjects were allowed to take any many breaks as needed. Subjects took an average of 10 – 15 minutes for the entire experiment.

7.5 Results

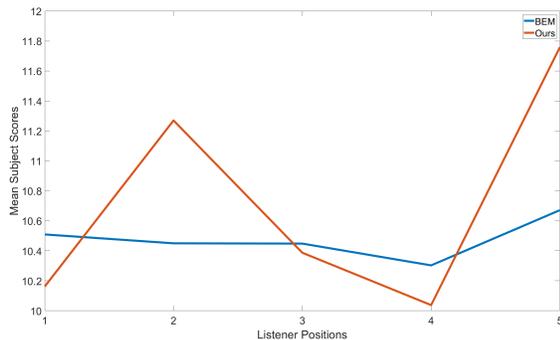


Fig. 11: Mean subject scores for different positions for the two methods of diffraction.

A two-way, repeated measures ANOVA (factors: diffraction method and listener positions) was performed on the subject’s ratings which were averaged over the three blocks, normalized by the subject’s mean score for all listener positions and diffraction methods, and scaled by the grand-mean. The test failed to show significance for position and diffraction method. Fig. 11 shows the mean values of the subject ratings for the two methods. The results show that our diffraction kernel algorithm performs comparably to the BEM-based wave propagation algorithm.

8 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

We present a novel approach to model diffraction effects for ray tracing based plausible sound generation algorithms. We introduce the

notion of diffraction kernels that can capture many wave effects like diffraction, reflections, scattering, intra-object interference and other interactions using wave-based precomputation. These kernels are computed independently for each dynamic object in the scene in a few minutes based on a novel source placement algorithm. Moreover, we can easily integrate these kernels with ray tracing based interactive geometric propagation algorithms and have small runtime overhead. We demonstrate the benefits over prior sound propagation algorithms on complex dynamic scenes. We also performed a user study to evaluate the perceived smoothness of the diffracted field and observed that the auditory perception using our approach is comparable to that of a wave-based sound propagation method. To the best of our knowledge, this is the first practical method to generate diffraction effects from a smooth object in dynamic scenes for VR applications.

Our approach has some limitations. While our hybrid approach offers many benefits over geometric acoustic methods, it is less accurate than wave-based propagation methods. Our approach is mainly designed for scenes with well-separated rigid objects, whose scattering behavior does not change at runtime. The diffraction kernels only encapsulate the sound interaction behavior of individual objects in the free field and do not account for phase or inter-object interactions. As a result, they may not work well in certain scenarios. Our formulation of diffraction kernels only take into account the magnitude and the direction, and not the phase.

There are many avenues for future work. It would be useful to model other interactions such as first order surface scattering based on Kirchoff approximation [33] or wave-based geometric acoustics [12] to model other wave interactions. It would be useful to design approximate schemes that can also model phase, as that is needed for certain applications, such as seat-dip effects in concert halls. The main goal is to estimate the propagation delays for all possible paths. Finally, we would like to perceptually evaluate our approach in other applications such as social VR and telepresence, where it is important to simulate diffraction effects and generate smooth sound fields.

9 ACKNOWLEDGEMENT

The authors would like to thank True Price, Nic Morales, Akash Bapat, and Tetsuya Takahashi for valuable insights and help at various stages of the project. The authors would also like to thank the anonymous subjects who took part in the user-study.

The work was supported in part by ARO grant W911NF14-1-0437, NSF grant 1320644, and Oculus Research.

REFERENCES

- [1] P. E. Barbone, J. M. Montgomery, O. Michael, and I. Harari. Scattering by a hybrid asymptotic/finite element method. *Computer methods in applied mechanics and engineering*, 164(1):141–156, 1998.
- [2] C. Cao, Z. Ren, C. Schissler, D. Manocha, and K. Zhou. Interactive sound propagation with bidirectional path tracing. *ACM Transactions on Graphics (TOG)*, 35(6):180, 2016.
- [3] P. A. Chinnery, V. F. Humphrey, and J. Zhang. Low-frequency acoustic scattering by a cube: Experimental measurements and theoretical predictions. *The Journal of the Acoustical Society of America*, 101(5):2571–2582, 1997.
- [4] D. Cohen-Steiner and J.-M. Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pp. 312–321. ACM, 2003.
- [5] J.-F. Dindart, J.-J. Embrechts, and C. Sémidor. Use of bidirectional reflectance distribution function in a particle tracing method. *The Journal of the Acoustical Society of America*, 105(2):1198–1198, 1999.
- [6] E. Dubois, P. Gray, and L. Nigay, eds. *The Engineering of Mixed Reality Systems*. Springer, London, 2010. doi: 10.1007/978-1-84882-733-2
- [7] E. Granier, M. Kleiner, B.-I. Dalenbäck, and P. Svensson. Experimental auralization of car audio installations. *Journal of the Audio Engineering Society*, 44(10):835–849, 1996.
- [8] S. Hampel, S. Langer, and A. Csilino. Coupling boundary elements to a raytracing procedure. *International journal for numerical methods in engineering*, 73(3):427–445, 2008.
- [9] D. L. James, J. Barbič, and D. K. Pai. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex

- vibration sources. In *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 987–995. ACM, 2006.
- [10] B. Kapralos, M. Jenkin, and E. Milios. Acoustical modeling using a russian roulette strategy. In *Proceedings of the 118th Convention of the Audio Engineering Society*, pp. 28–31, 2005.
- [11] A. Krokstad, S. Strom, and S. Sørsdal. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration*, 8(1):118–125, 1968.
- [12] Y. W. Lam. Issues for computer modelling of room acoustics in non-concert hall settings. *Acoustical science and technology*, 26(2):145–155, 2005.
- [13] P. Larsson, D. Vastfjäll, and M. Kleiner. Better presence and performance in virtual environments by improved binaural sound rendering. In *Virtual, Synthetic, and Entertainment Audio conference*, Jun 2002.
- [14] T. Lentz, D. Schröder, M. Vorländer, and I. Assenmacher. Virtual reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Advances in Signal Processing*, 2007:187–187, January 2007.
- [15] T. Lokki, A. Southern, S. Siltanen, and L. Savioja. Studies of epidaurus with a hybrid room acoustics modelling method. *Acoustics of Ancient Theaters Patras, Greece*, 2011.
- [16] R. Mehra, N. Raghuvanshi, L. Antani, A. Chandak, S. Curtis, and D. Manocha. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Transactions on Graphics (TOG)*, 32(2):19, 2013.
- [17] R. Mehra, A. Rungta, A. Golas, M. Lin, and D. Manocha. Wave: Interactive wave-based sound propagation for virtual environments. *IEEE transactions on visualization and computer graphics*, 21(4):434–442, 2015.
- [18] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 560–568. ACM, 2006.
- [19] N. Raghuvanshi, A. Allen, and J. Snyder. Numerical wave simulation for interactive audio-visual applications. *The Journal of the Acoustical Society of America*, 139(4):2008–2009, 2016.
- [20] N. Raghuvanshi, J. Snyder, R. Mehra, M. Lin, and N. Govindaraju. Pre-computed wave simulation for real-time sound propagation of dynamic sources in complex scenes. In *ACM Transactions on Graphics (TOG)*, vol. 29, p. 68. ACM, 2010.
- [21] S. R. M. Román, U. P. Svensson, J. Šlechta, and J. O. Smith. A hybrid method combining the edge source integral equation and the boundary element method for scattering problems. *The Journal of the Acoustical Society of America*, 139(4):2202–2202, 2016.
- [22] A. Rungta, N. Rewkowski, R. Klatzky, M. Lin, and D. Manocha. Effects of virtual acoustics on dynamic auditory distance perception. *The Journal of the Acoustical Society of America*, 141(4):EL427–EL432, 2017.
- [23] A. Rungta, S. Rust, N. Morales, R. Klatzky, M. Lin, and D. Manocha. Psychoacoustic characterization of propagation effects in virtual environments. *ACM Transactions on Applied Perception (TAP)*, 13(4):21, 2016.
- [24] L. Savioja and U. P. Svensson. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America*, 138(2):708–730, 2015.
- [25] C. Schissler and D. Manocha. Interactive sound propagation and rendering for large multi-source scenes. *ACM Transactions on Graphics (TOG)*, 36(1):2, 2016.
- [26] C. Schissler, R. Mehra, and D. Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (SIGGRAPH 2014)*, 33(4):39, 2014.
- [27] D. Schröder. *Physically based real-time auralization of interactive virtual environments*, vol. 11. Logos Verlag Berlin GmbH, 2011.
- [28] A. Southern, S. Siltanen, and L. Savioja. Spatial room impulse responses with a hybrid modeling method. In *Audio Engineering Society Convention 130*. Audio Engineering Society, 2011.
- [29] U. M. Stephenson. An energetic approach for the simulation of diffraction within ray tracing based on the uncertainty relation. *Acta Acustica united with Acustica*, 96(3):516–535, 2010.
- [30] U. P. Svensson, R. I. Fred, and J. Vanderkooy. An analytic secondary source model of edge diffraction impulse responses. *The Journal of the Acoustical Society of America*, 106(5):2331–2344, 1999.
- [31] M. Taylor, A. Chandak, Q. Mo, C. Lauterbach, C. Schissler, and D. Manocha. Guided multiview ray tracing for fast auralization. *IEEE Transactions on Visualization and Computer Graphics*, 18:1797–1810, 2012.
- [32] R. R. Torres, U. P. Svensson, and M. Kleiner. Computation of edge diffraction for more accurate room acoustics auralization. *The Journal of the Acoustical Society of America*, 109(2):600–610, 2001.
- [33] N. Tsingos, C. Dachsbacher, S. Lefebvre, and M. Dellepiane. Instant sound scattering. In *Proceedings of the Eurographics Symposium on Rendering*, pp. 111–120, 2007.
- [34] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 545–552. ACM, 2001.
- [35] N. Tsingos and J.-D. Gascuel. Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments. In *Audio Engineering Society Convention 104*. Audio Engineering Society, 1998.
- [36] G. Vickers. The projected areas of ellipsoids and cylinders. *Powder technology*, 86(2):195–200, 1996.
- [37] Y. Wang, S. Safavi-Naeini, and S. K. Chaudhuri. A hybrid technique based on combining ray tracing and ftdtd methods for site-specific modeling of indoor radio wave propagation. *IEEE Transactions on antennas and propagation*, 48(5):743–754, 2000.
- [38] H. Yeh, R. Mehra, Z. Ren, L. Antani, D. Manocha, and M. Lin. Wave-ray coupling for interactive sound propagation in large complex scenes. *ACM Transactions on Graphics (TOG)*, 32(6):165, 2013.