Wave-Based Sound Propagation in Large Open Scenes using an Equivalent Source Formulation

RAVISH MEHRA
University of North Carolina at Chapel Hill
and
NIKUNJ RAGHUVANSHI
Microsoft Research
and
LAKULISH ANTANI and ANISH CHANDAK and SEAN CURTIS and DINESH MANOCHA
University of North Carolina at Chapel Hill

We present a novel approach for wave-based sound propagation suitable for large, open spaces spanning hundreds of meters, with a small memory footprint. The scene is decomposed into disjoint rigid objects. The free-field acoustic behavior of each object is captured by a compact per-object transfer-function relating the amplitudes of a set of incoming equivalent sources to outgoing equivalent sources. Pairwise acoustic interactions between objects are computed analytically to yield compact inter-object transfer functions. The global sound field accounting for all orders of interaction is computed using these transfer functions. The runtime system uses fast summation over the outgoing equivalent source amplitudes for all objects to auralize the sound field for a moving listener in real-time. We demonstrate realistic acoustic effects such as diffraction, low-passed sound behind obstructions, focusing, scattering, high-order reflections, and echoes, on a variety of scenes.

Categories and Subject Descriptors:

1. INTRODUCTION

Interactive sound propagation has emerged as a powerful tool in computer graphics to enhance the realism of virtual worlds by predicting the behavior of sound as it interacts with the environment [Takala and Hahn 1992; Funkhouser et al. 1998; Manocha et al. 2009]. In order to accurately capture important acoustic phenomena in general scenarios, including interference, diffraction, scattering, sound focusing (caustics), and higher-order wave effects resulting from their combination, it is important to develop techniques that can directly solve the acoustic wave equation. There is extensive work in scientific computing and acoustics on numerical methods to solve the wave equation. Furthermore, there has been considerable interest in developing interactive wave-based techniques to model free-space sound radiation [James et al. 2006], first-order scattering from surfaces [Tsingos et al. 2007], and sound propagation for indoor scenes [Savioja 2010; Raghuvanshi et al. 2010].

Large, open scenes, which arise in many applications ranging from games to training or simulation systems, present a significant challenge for interactive, wave-based sound propagation techniques. State-of-the-art wave simulation methods can take hours of computation and gigabytes of memory for performing sound propagation in indoor scenes such as concert halls [Sakamoto et al. 2006; Raghuvanshi et al. 2009]. For large, open scenes spanning hundreds of meters, it is challenging to run these techniques in real-time. On the other hand, geometric (ray-based) acoustic techniques can provide real-time performance for such environments. However, geometric techniques are better suited for higher frequencies due to the inherent assumption of rectilinear propagation of sound waves. Therefore, accurately modeling diffraction and higher-order wave effects with these techniques remains a significant challenge, especially at low frequencies.

In this paper, we present a novel approach for precomputed, wave-based sound propagation that is applicable to large, open scenes. It is based on the *equivalent source method*, which has been widely studied for radiation and scattering problems in acoustics and electromagnetics [Doicu et al. 2000] and more recently introduced to computer graphics [James et al. 2006]. Our approach consists of two main stages: preprocessing and runtime. During preprocessing, we decompose the scene into disjoint, well-separated rigid objects. The acoustic behavior of each object, taken independently, is characterized by its *per-object transfer function* that maps an arbitrary incident field on the object to the resulting scattered field. We propose an equivalent source formulation to express this transfer function as a compact *scattering matrix*. Pairwise acoustic coupling between objects is then modeled by computing *inter-object transfer functions* between all pairs of objects that maps the outgoing scattered field from one object to the incoming field on another object. These transfer functions are represented compactly by using the same equivalent source framework to yield *interaction matrices*. Acoustic transfer between multiple objects can therefore be represented using chained multiplication of their scattering and interaction matrices. Finally, the acoustic response of the scene to a static source distribution is computed by solving a global linear system that accounts for all orders of inter-object wave propagation.

At runtime, fast summation over all outgoing equivalent sources for all objects is performed at the listener location. The computed response is used for real-time sound rendering for a moving listener. Multiple moving sources, with a static listener, are handled by exploiting acoustic reciprocity. The runtime memory and computational requirements are proportional to the number of objects and their outgoing scattered field complexity (usually a few thousand equivalent sources per frequency for a few percent error), instead of the volume or surface area of the scene. Thus, our technique takes an object-centric approach to wave-based sound propagation. The key contributions of our work include:

Object-based sound field decomposition using per-object and inter-object acoustic transfer functions for enabling real-time, wave-based sound propagation on large, open scenes.

Compact per-object transfer using equivalent sources to model the scattering behavior of an object mapping arbitrary incident fields to the resultant scattered fields.

Compact analytical coupling of objects is achieved by expressing inter-object transfer functions in the same, compact equivalent source basis as used for per-object transfer.

A fast, memory-efficient run-time enables real-time sound rendering, while requiring only a few tens of megabytes of memory.

Our approach is well-suited for quick iterations while authoring scenes. Per-object transfer functions, which take a significant portion of the precomputation time of our method, are independent of the scene and can thus be stored in a lookup table. Therefore, adding, deleting or moving a few objects in an existing scene has low precomputation overhead, linear in the number of objects.

We have tested our technique on a variety of scenarios (see Figure 1) and integrated our system with the Valve's Source^{TM} game engine from Half-Life 2. Our technique generates realistic acoustic effects and takes orders of magnitude less runtime memory compared to state-of-the-art wave solvers, enabling interactive performance. To the best of our knowledge, it is the first real-time technique for accurate, wave-based sound propagation in large, open scenes.

2. RELATED WORK

Our technique has close theoretical parallels with prior numerical wave solvers. We first explore these connections, followed by related work on interactive geometric and wave-based techniques.

2.1 Numerical Wave Solvers

Research in wave-based acoustic simulation techniques spans a broad range of areas such as noise control, automotive design, urban architectural planning, and concert hall design. Wave solvers can be classified into *volumetric* and *surface-based* approaches. The most common among volumetric techniques are the finite element method (FEM) [Zienkiewicz et al. 2006; Thompson 2006] and finite difference time domain (FDTD) method [Yee 1966; Taflove and Hagness 2005; Sakamoto et al. 2006], which require a discretization of the entire volume of the 3D scene. The compute and memory usages of these methods scale linearly with the volume of the scene. Faster methods like pseudospectral time domain (PSTD) [Liu 1997] and adaptive rectangular decomposition (ARD) [Raghuvanshi et al. 2009] have been proposed and achieve good accuracy with a much coarser spatial discretization. Volumetric techniques are well-suited for scenes with high surface area and low air volume, which makes them highly applicable to indoor spaces.

Surface-based techniques are better suited for open scenes, for which scattering geometry is sparse with large regions of air with uniform wave-propagation speed. The most common approach here is the boundary element method (BEM) [Cheng and Cheng 2005] that expresses the global acoustic field as the sum of elementary radiating fields from monopole and dipole sources placed on a uniform, sub-wavelength sampling of the scene's surface. Traditional BEM scales as the square of the surface area but recent research on the fast multipole method for BEM (FMM-BEM)[Liu et al. 2009; Gumerov and Duraiswami 2009] has improved the complexity to linear in surface area by creating a hierarchical clustering of BEM monopoles and dipoles using an octree, and approximating their interactions compactly using high-order multipole Green's functions. Offline FMM-BEM solutions are infeasible for interactive applications because of the large, dense number of monopole and dipole sources in the final solution that need to be stored and summed on the fly.

For acoustic radiation and scattering problems, an efficient and powerful surface-based technique is the equivalent source method (ESM) [Fairweather 2003; Kropp and Svensson 1995; Ochmann 1999; Pavic 2006] that forms the basis of our formulation. Instead of relying on a boundary-integral formulation, as BEM does, ESM exploits the uniqueness of solutions to the acoustic boundary value problem. Equivalent multipole sources, Green's functions, are placed at variable locations in space with the intent of making the total generated field match boundary conditions on the object's surface, since uniqueness guarantees the correctness of the solution (Section 3 in [Ochmann 1995]). The flexibility of location results in fewer multipole sources. The ESM can yield large gains in performance and memory-efficiency for scattering and radiation problems in large spaces, and has been used widely in both acoustic and electromagnetic applications [Doicu et al. 2000]. Equivalent sources were introduced to computer graphics in the seminal work of [James et al. 2006] on sound generation from vibrating objects. ESM is an attractive starting point for such precomputation-based approaches, and our method, because it allows very flexible performance-to-accuracy tradeoffs. More importantly, the compactness of the solutions reduces runtime memory and compute requirements by a large factor, making them amenable to real-time evaluation.

A related technique, called the transition-matrix method, has been used extensively for electromagnetic scattering, and also developed for acoustics [Waterman 2009]. The method relates the incoming and outgoing fields in the scattering process in terms of the coefficients of a complete system of vector basis functions that are not necessarily Green's functions, unlike BEM or ESM.

2.2 Interactive Geometric Techniques

Most current interactive sound propagation systems are based on geometric acoustics, which applies the high-frequency Eikonal (ray) approximation to sound propagation. The image source method [Allen and Berkley 1979] is the most commonly used geometric technique, and there has been much research on improving its performance [Funkhouser et al. 1998]. However, the image source method can only model purely specular reflections. Other techniques based on ray tracing [Krokstad et al. 1968; Vorlander 1989; Lentz et al. 2007] or radiosity [Tsingos and Gascuel 1997] have been developed for modeling diffuse reflections, but these energy-based formulations may not model phase accurately. Techniques based on acoustic radiance transfer [Siltanen et al. 2007; Siltanen et al. 2009] can model arbitrary surface interactions for wide-band signals, but cannot accurately model wave phenomena such as diffraction. The two main approaches for modeling diffraction in a geometric acoustics framework are the uniform theory of diffraction (UTD) [Tsingos et al. 2001] and the Biot-Tolstoy-Medwin (BTM) formulation [Svensson et al. 1999]. UTD is an approximate formulation, while the BTM yields accurate results with a significant performance cost. Methods based on image source gradients [Tsingos 2009] and acoustic radiance transfer operators [Antani et al. 2012] have been developed to interactively model higher-order propagation effects. Recent developments in fast ray tracing have enabled interactive geometric propagation in dynamic scenes, but these methods only model first-order edge diffraction based on UTD [Taylor et al. 2009].

2.3 Interactive Wave-simulation Techniques

In recent years, we have seen increasing interest in developing interactive wave-simulation techniques for sound propagation in indoor and outdoor spaces. Sound radiation from a single vibrating object in free space can be efficiently modeled using precomputed acoustic transfer [James et al. 2006]. These acoustic transfer functions approximate the radiation behavior of a complicated geometry by expressing it in terms of equivalent sources, which can be quickly evaluated at runtime to enable real-time performance. Tsingos et al. [2007] solves the boundary integral formulation of the Helmholtz equation subject to the Kirchhoff approximation in real-time. Raghuvanshi et al. [2010] relies on a volumetric sampling of acoustic responses on a spatial grid and perceptual encoding based on the acoustic properties of indoor spaces. Recent

Figure 1: *Our algorithm accurately models realistic acoustic effects, such as diffraction, scattering, focusing, and echoes, in large, open scenes. We reduce the runtime memory usage by orders of magnitude compared to state-of-the-art wave solvers, enabling real-time, wave-based sound propagation in scenes spanning hundreds of meters: a) reservoir scene (Half-Life 2), b) Christmas scene, and c) desert scene.*

work [Savioja 2010] has shown that FDTD simulations can run in real-time on the GPU, but only for very small spaces that span a few meters across. We compare our method in more detail with these closely related interactive wave-simulation techniques in Section 6.4.

3. THE EQUIVALENT SOURCE METHOD

In this section, we give a brief review of the Equivalent Source Method. Consider the exterior scattering problem [Thompson and Pinsky 2004], a solid three-dimensional object A immersed in an unbounded air volume (see Figure 2(a)). Considering only time-harmonic vibrations, with angular frequency ω and a homogeneous medium with constant speed of sound c, acoustic wave propagation can be expressed as a boundary value problem for the Helmholtz equation:

$$\nabla^2 p + \frac{w^2}{c^2} p = 0 \ \text{ in } A^+, \tag{1}$$

where p is the (complex-valued) pressure field, A^+ is the domain exterior to the object, and ∇^2 is the Laplacian operator. At the boundary of the domain, ∂A, the pressure is specified using a Dirichlet boundary condition:

$$p = f(\mathbf{x}) \ \text{ on } \partial A. \tag{2}$$

To complete the problem specification, the behavior of p at infinity must be specified, usually by the *Sommerfeld radiation condition* [Pierce 1989]:

$$\lim_{r\to\infty} r \left[\frac{\partial p}{\partial r} - \hat{j}\frac{w}{c}p\right] = 0, \tag{3}$$

where $r = \|\mathbf{x}\|$ is the distance of point $\mathbf{x}$ from the origin and $\hat{j} = \sqrt{-1}$. The equivalent source method [Ochmann 1995; 1999; Pavic 2006] relies on the existence of fundamental solutions also called Green's functions or equivalent sources $q(\mathbf{x}, \mathbf{y})$, of the Helmholtz equation (1) subject to the Sommerfeld radiation condition (3) for all $\mathbf{x} \neq \mathbf{y}$. An equivalent source $q(\mathbf{x}, \mathbf{y}_i)$ is the solution field induced at any point $\mathbf{x}$ due to a point source located at $\mathbf{y}_i$, and can be expressed as the sum:

$$q(\mathbf{x}, \mathbf{y}_i) = \sum_{l=0}^{L-1}\sum_{m=-l}^{l} d_{ilm}\varphi_{ilm}(\mathbf{x}) = \sum_{k=1}^{L^2} d_{ik}\varphi_{ik}(\mathbf{x}), \tag{4}$$

where k is a generalized index for (l, m). The fundamental solution $\varphi_{ilm}(\mathbf{x})$ is the field due to a *multipole* source located at $\mathbf{y}_i$, d_{ilm} is its strength, and L is the order of the multipole ($L = 1$ is just a monopole, $L = 2$ includes dipole terms as well, and so on). The field due to a multipole located at point $\mathbf{y}_i$ is defined as

$$\varphi_{ilm}(\mathbf{x}) = \Gamma_{lm} h_l^{(2)}(wr_i/c)\psi_{lm}(\theta_i, \phi_i) \tag{5}$$

where (r_i, θ_i, ϕ_i) is the vector $(\mathbf{x} - \mathbf{y}_i)$ expressed in spherical coordinates, $h_l^{(2)}(wr_i/c)$ are the (complex-valued) spherical Hankel functions of the second kind [Abramowitz and Stegun 1964], $\psi_{lm}(\theta_i, \phi_i)$ are the (complex-valued) spherical harmonic functions [Hobson 1955], and Γ_{lm} is the (real-valued) normalizing factor that makes the spherical harmonics orthonormal.

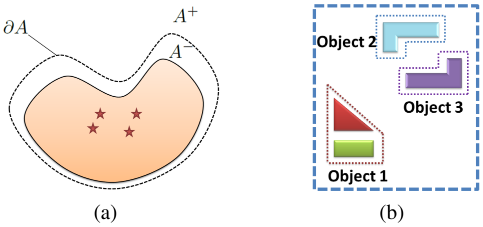


Figure 2: *a) A diagram illustrating a radiating object A, its corresponding boundary ∂A, exterior region A^+, interior region A^-, and the set of equivalent sources (denoted by star shapes). b) Classification of objects in a scene. The triangle and rectangle constitute a single object, as their offset surfaces overlap. On the other hand, L-shaped shapes are classified as separate objects.*

The fundamental solutions $\varphi_{ilm}(\mathbf{x})$ (or $\varphi_{ik}(\mathbf{x})$) are used to solve the Helmholtz equation. Consider the outgoing scattered field due to an object, and the associated Dirichlet boundary value problem on ∂A. Consider a discrete set of R source locations $\{\mathbf{y}_i\}_{i=1}^R$, all contained in the interior region A^-. The total field due to these sources at any $\mathbf{x} \in A^+$ is

$$p(\mathbf{x}) = \sum_{i=1}^{R} c_i q(\mathbf{x}, \mathbf{y}_i) = \sum_{i=1}^{R}\sum_{k=1}^{L^2} c_{ik}\varphi_{ik}(\mathbf{x}), \tag{6}$$

where $c_{ik} = c_i d_{ik}$ are the corresponding strengths of the equivalent sources. The main idea of the ESM is that if the equivalent source strengths c_{ik} and positions $\mathbf{y}_i$ are chosen to match the Dirichlet boundary condition on ∂A,

$$p(\mathbf{x}) = \sum_{i=1}^{R}\sum_{k=1}^{L^2} c_{ik}\varphi_{ik}(\mathbf{x}) = f(\mathbf{x}); \quad \mathbf{x} \in \partial A, \tag{7}$$

then $p(\mathbf{x})$ is the correct solution over all A^+.

This process can also be used to represent the incident field of an object, the only difference being that the equivalent sources are now placed in the exterior region A^+. Again, by matching the boundary condition (7), we get the correct solution $p(\mathbf{x})$ for all $\mathbf{x}$ in the interior region A^-.

In practice, the boundary conditions (7) can only be satisfied approximately for a finite value of R, and the degree of approximation can be controlled by changing R. Since the strengths of multipoles of each source must be stored and its contribution evaluated at runtime, R is the main parameter for trading accuracy for runtime performance and memory requirements. This flexibility makes ESM highly suitable for interactive applications.

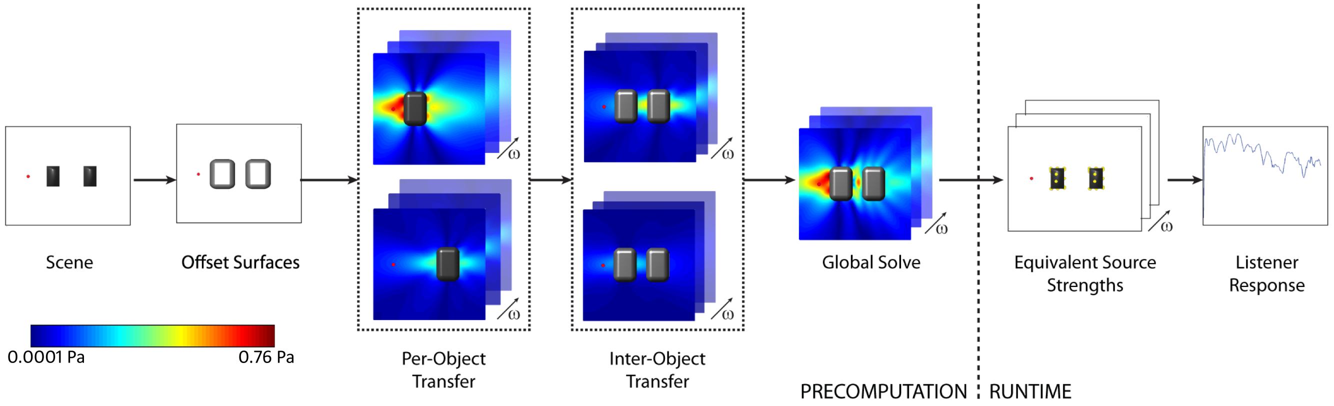


Figure 3: *Overview of our wave-based sound propagation technique based on equivalent sources on a simple scene composed of two objects and a sound source (shown with a red dot). The magnitudes of pressure fields are visualized using the color scheme shown.*

4. SOUND PROPAGATION USING ESM

We give a brief overview of the precomputation and runtime stages of our technique (see Figure 3). Our formulation is in the frequency domain. We construct a complex frequency response (containing magnitudes and phases), at regularly sampled frequencies, to model the delay information in the propagated sound. Thus, the steps outlined in this section, except the offset surface calculation, need to be performed for a regularly sampled set of frequencies in the range $[0, \nu_{\max}]$, where $\nu_{\max}$ is the maximum simulated frequency. We assume that the scene is composed of static objects. Table I provides a list of commonly used symbols.

4.1 Our Approach

Offset surface calculation: In the preprocessing stage, we classify objects in the scene and calculate the offset surface for each object.
Per-object transfer function: For each object, we compute a *per-object transfer function* that maps the incoming field incident on the object to the outgoing scattered field.
Inter-object transfer function: For each object pair, we precompute an *inter-object transfer function* that encodes how the outgoing scattered field of one object becomes the incoming field for the other object.
Global solve: Based on the per-object and inter-object transfer functions and a sound source configuration, we model acoustic interactions between the objects in the scene and solve for the global sound field. Thereby, we compute the strengths of all the outgoing scattered field equivalent sources of all objects.
Run-time pressure evaluation: At runtime, we add the pressure produced at the listener position by all outgoing field equivalent sources, for each frequency. This is an extremely fast computation, and can be performed for a moving listener in real-time.

4.2 Offset Surface Calculation

The first step is to decompose the input scene into *well-separated* objects. To decide if two objects are well-separated, we use the notion of an *offset surface*. The offset surface is defined by taking the constant offset along the normal direction at each point on the boundary of the object. Two objects are considered disjoint if and only if their offset surfaces do not intersect. Otherwise, we combine them and treat them as a single object (see Figure 2(b)). We compute the offset surface of an object using distance field and the marching cubes algorithm similar to James et al. [2006]. Typical values of distance field voxel resolution h and offset distance δ are specified in Table II. The offset surface serves as the boundary of the domain ∂A. After decomposing the scene into well-separated objects, we compute the scattering properties for each object independently.

Symbols	Meaning
q_i^{in}, q_j^{out}	i^{th} & j^{th} eq. src for incoming, outgoing field resp.
$\varphi_{ik}^{in}, \varphi_{jh}^{out}$	k^{th} & h^{th} multipole term of eq. src. q_i^{in} & q_j^{out} resp.
Q, P	number of incoming, outgoing eq. srcs resp.
M, N	order of incoming, outgoing field multipoles resp.

Table I. : *Table of commonly used symbols.*

4.3 Per-object Transfer Function

In order to capture an object's scattering behavior, we define the *per-object transfer function* f, a function which maps an arbitrary incoming field reaching the object to the corresponding outgoing scattered field after reflection, scattering and diffraction due to the object itself. This function is linear owing to the linearity of the wave equation and depends only on the shape and material properties of the object.

The incoming and outgoing fields for an object A are both expressed using equivalent sources. The outgoing field is represented by placing equivalent sources $\{q_1^{out}, q_2^{out}, q_3^{out}, ...\}$ in the interior region A^- of the object. Similarly, the incoming field is represented by placing equivalent sources $\{q_1^{in}, q_2^{in}, q_3^{in}, ...\}$ in the exterior region A^+. The transfer function f maps the basis of the incoming field (multipoles φ_{ik}^{in}) to the corresponding outgoing field expressed as a linear combination of its basis functions (multipoles φ_{jh}^{out}):

$$f(\varphi_{ik}^{in}) = \sum_{(j,h)=(1,1)}^{(P,N^2)} \alpha_{jh}^{ik} \varphi_{jh}^{out}; \qquad (8)$$

$$\begin{bmatrix} f(\varphi_{11}^{in}) \\ f(\varphi_{12}^{in}) \\ . \\ . \\ f(\varphi_{QM^2}^{in}) \end{bmatrix} = \begin{bmatrix} \alpha_{11}^{11} & \alpha_{12}^{11} & \dots & \alpha_{PN^2}^{11} \\ \alpha_{11}^{12} & \alpha_{12}^{12} & \dots & \alpha_{PN^2}^{12} \\ . & . & \dots & . \\ . & . & \dots & . \\ \alpha_{11}^{QM^2} & \alpha_{12}^{QM^2} & \dots & \alpha_{PN^2}^{QM^2} \end{bmatrix} \begin{bmatrix} \varphi_{11}^{out} \\ \varphi_{12}^{out} \\ . \\ . \\ \varphi_{PN^2}^{out} \end{bmatrix}$$

$$= T_A \Phi_A^{out}, \qquad (9)$$

where $\alpha_{jh}^{ik} \equiv T_A(ik, jh)$ is the (complex) amplitude for the outgoing multipole φ_{jh}^{out} induced by a unit-amplitude incoming multipole φ_{ik}^{in}. The per-object sound transfer function for object A is encoded in the coefficient matrix T_A, which we call the *scattering matrix*. We now explain how to compute the (complex) amplitudes α_{jh}^{ik} of the outgoing field multipoles. Details on choosing the number and positions of incoming and outgoing equivalent sources are given in Section 4.5.

Computing the Scattering Matrix. For each incoming field multipole φ_{ik}^{in} in turn, we place a unit-amplitude sound source and use a numerical wave solver to compute the total pressure field at n uniformly-sampled locations $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ on ∂A. We subtract the incident field from the total pressure field to compute the outgoing scattered field at these sampled locations (see Figure 4), denoted by $\bar{\mathrm{p}}_{ik} = \{p(\mathbf{x}_1), p(\mathbf{x}_2), ..., p(\mathbf{x}_n)\}$.

We fit the outgoing field multipole expansion to the sampled scattered field, in a least-squares sense, by solving an over-determined linear system ($n > PN^2$) subject to a pre-specified error threshold σ for all incoming field multipoles:

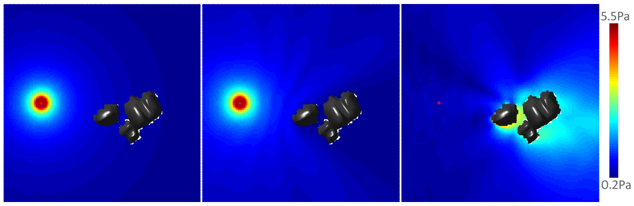
$$\sum_{(j,h)=(1,1)}^{(P,N^2)} \varphi_{jh}^{out}(\mathbf{x}_t)\ \alpha_{jh}^{ik} = p(\mathbf{x}_t),\ \text{for } t = 1, ..., n; \tag{10}$$

$$\mathrm{V}\boldsymbol{\alpha}^{ik} = \bar{\mathrm{p}}_{ik}. \tag{11}$$

The least-squares solution yields the coefficients $\boldsymbol{\alpha}^{ik}$ corresponding to the ik^{th} row of the scattering matrix T. This process is repeated for all incoming field multipoles to compute the scattering matrix. The solution can be computed efficiently using a single combined linear system

$$\mathrm{V}\, T_A^{tr} = \left[\bar{\mathrm{p}}_{11} \dots \bar{\mathrm{p}}_{QM^2}\right], \tag{12}$$

where T_A^{tr} is the transpose of T_A. The per-object transfer function is computed for all objects at sampled frequencies. The error threshold σ is used while deciding the number and placement of equivalent sources (Section 4.5) such that the above linear system gives error less than σ.



(a) Incident field (b) Total field (c) Scattered field ($\times 8$)

Figure 4: *Magnitude of the pressure field (in Pa) at* 170 *Hz in a simple scene with a single object (rocks) and a single sound source (red dot). The difference between total and incident fields is the scattered field (scaled eight times for visualization). Note the high amplitude of the scattered field between the rocks representing the large difference in incident and total field that results from diffracted occlusion.*

4.4 Inter-object Transfer Function

Scenes with multiple objects exhibit object-to-object interactions, where the outgoing field from one object serves as the incoming field for the other objects. For example, with two objects A and B, source s and listener l, the possible interactions that can occur from s to l are: direct sound (0^{th} order) $s \to l$, 1^{st} order $s \to A \to l; s \to B \to l$, 2^{nd} order $s \to A \to B \to l; s \to B \to A \to l$, and so on. We model these interactions by formulating an *inter-object transfer function*. For two objects A and B, the inter-object transfer function g_A^B expresses the outgoing field of A in terms of the basis of the incoming field of B. Like the per-object transfer function, the inter-object transfer function is also a linear function. The inter-object transfer function g_A^B maps each basis function of the outgoing field of A (multipoles φ_{jh}^{out}) to the corresponding incoming field of B expressed as a linear combination of its basis functions (multipoles φ_{ik}^{in}):

$$g_A^B(\varphi_{jh}^{out}) = \sum_{(i,k)=(1,1)}^{(Q,M^2)} \beta_{ik}^{jh} \varphi_{ik}^{in}; \tag{13}$$

$$\begin{aligned}
\begin{bmatrix} g_A^B(\varphi_{11}^{out}) \\ g_A^B(\varphi_{12}^{out}) \\ . \\ . \\ g_A^B(\varphi_{PN^2}^{out}) \end{bmatrix} &= \begin{bmatrix} \beta_{11}^{11} & \beta_{12}^{11} & \dots & \beta_{QM^2}^{11} \\ \beta_{11}^{12} & \beta_{12}^{12} & \dots & \beta_{QM^2}^{12} \\ . & . & \dots & . \\ . & . & \dots & . \\ \beta_{11}^{PN^2} & \beta_{12}^{PN^2} & \dots & \beta_{QM^2}^{PN^2} \end{bmatrix} \begin{bmatrix} \varphi_{11}^{in} \\ \varphi_{12}^{in} \\ . \\ . \\ \varphi_{QM^2}^{in} \end{bmatrix} \\
&= G_A^B \Phi_B^{in},
\end{aligned} \tag{14}$$

where $\beta_{ik}^{jh} \equiv G_A^B(jh, ik)$ is the (complex) amplitude of the incoming multipole φ_{ik}^{in} of B induced by a unit-amplitude outgoing multipole φ_{jh}^{out} of A. The inter-object transfer function from A to B is thus encoded as G_A^B, which we call the *interaction matrix*. Generally, the interaction matrix is *not* symmetric, i.e., $G_A^B \neq G_B^A$. Since the outgoing field of an object is not defined in its interior region, G_A^A and G_B^B are zero matrices. We now explain how to compute the (complex) amplitudes β_{ik}^{jh} of the incoming field multipoles.

Computing the Interaction Matrix. The interaction matrix G_A^B can be computed using a least-squares formulation similar to the one used for computing scattering matrices. However, the pressure values at the offset surface samples of B, $\bar{\mathrm{p}}_{jh} = \{p(\mathbf{x}_1), p(\mathbf{x}_2), ..., p(\mathbf{x}_n)\}$ are simpler to compute. In a homogenous medium, the outgoing field due to a multipole is the same as the free space field, for which analytical expressions exist (Equation 5). Therefore, we simply evaluate the analytical expressions of the outgoing field multipoles φ_{jh}^{out} of A at the sample points on the offset surface of B. The resulting linear system is solved subject to a separate error threshold, η :

$$\sum_{(i,k)=(1,1)}^{(Q,M^2)} \varphi_{ik}^{in}(\mathbf{x}_t)\ \beta_{ik}^{jh} = p(\mathbf{x}_t),\ \text{for } t = 1, ..., n. \tag{15}$$

Again, this process is repeated for each outgoing multipole of B, and solved efficiently as a single combined linear system:

$$\mathrm{U}\, G_A^{B\,tr} = \left[\bar{\mathrm{p}}_{11} \dots \bar{\mathrm{p}}_{PN^2}\right]. \tag{16}$$

The inter-object transfer functions are computed for all object pairs, independently for each frequency.

4.5 Computing Equivalent Source Positions

Choosing Offset Surface Samples. Solving equations (12) and (16) at frequency ν involves computing the pressure at sampled locations $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ on the offset surface of each object. The number of sampled locations n depends on the spatial variation of the pressure field, which in turn, depends directly on its frequency ν or inversely on its wavelength λ since $\nu = c/\lambda$. As per the Nyquist Theorem, representing a signal of frequency ν with a finite number of samples requires a sampling rate of 2ν. The spatially-varying pressure field defined on the 2D offset surface must be sampled at a rate of 2ν in both dimensions. The distance between samples become $2\nu/c = 2/\lambda$. Therefore, we place $n \propto (2\nu/c)^2$x *surface area* $= (2/\lambda)^2$x *surface area* samples uniformly on the offset surface.

Choosing Incoming Equivalent Sources. Since the nature of the incoming field is not known *a priori*, it is difficult to optimize the number and position of incoming equivalent sources. We resolve this problem by generating another offset surface at distance $\Delta > \delta$ from the object, where δ is the original offset surface's distance, and placing incoming equivalent sources on this new surface (see Table II for the value of Δ). The number of incoming equivalent sources Q depends on the spatial variation of the incoming pressure field. As before, $Q \propto (2/\lambda)^2$x *surface area* equivalent sources are uniformly placed. This allows us to represent the incoming field on the inner offset surface to good accuracy.

Choosing Outgoing Equivalent Sources. The number of outgoing equivalent sources P and their positions are decided based on a multi-level source placement algorithm similar to James et al. [2006]. The previous algorithm was designed to satisfy a single radiating field $\bar{\mathrm{p}}$ of an object at each frequency. It places equivalent sources in a greedy manner, where at each step a set of candidate positions χ are ranked based on their ability to reduce the pressure residual vector $\bar{\mathrm{r}} = \bar{\mathrm{p}}/||\bar{\mathrm{p}}||_2$ on the offset surface. The best candidate position $\mathbf{x}^*$ is chosen via the largest projection, i.e., $\mathbf{x}^* = \arg\max_{x \in \chi} u$, where projection $u = ||(U_{\mathbf{x}})^H \bar{\mathrm{r}}||_2$. The unitary matrix corresponding to the subspace spanning all the previously selected positions is updated. The residual vector is updated by removing its component in that subspace. The process is repeated until the value of the residual $||\bar{\mathrm{r}}||_2$ falls below the error tolerance. The set of best candidate positions selected in the process is the set of outgoing equivalent sources and its size gives us the value of P.

Our algorithm is designed to satisfy multiple outgoing radiating fields at each frequency simultaneously. In our case, at each frequency, we have as many outgoing radiating fields $\left[\bar{\mathrm{p}}_{11} \ldots \bar{\mathrm{p}}_{QM^2}\right]$ as the number of incoming multipoles QM^2. This gives us a vector of pressure residual vectors $\mathbf{r} = [\bar{\mathrm{r}}_{11} \ \ldots \ \bar{\mathrm{r}}_{QM^2}]$ and a corresponding vector of projections $\mathbf{u} = [u_{11} \ \ldots \ u_{QM^2}]$ where $u_{ik} = ||(U_{\mathbf{x}})^H \bar{\mathrm{r}}_{ik}||_2$. We choose the best candidate as the one that minimizes the pressure residual of all outgoing fields simultaneously via a modified largest projection $\mathbf{x}^* = \arg\max_{x \in \chi} ||\mathbf{u}||_2$. We update the unitary matrix and for each residual vector we remove its component in the chosen subspace. We then compute the value of the modified residual $||\mathbf{d}||_2$, where $\mathbf{d} = [d_{11} \ \ldots \ d_{QM^2}]$ and $d_{ik} = ||\bar{\mathrm{r}}_{ik}||_2$. We repeat this process until the relative value of the modified residual falls below the error tolerance (σ in our case). Similar to the number of incoming equivalent sources Q, the number of outgoing equivalent sources P also increases with frequency. But it strongly depends on the shape of the object and the complexity of the outgoing scattered field that the object generates. We fit as many equivalent sources as necessary to satisfy the error threshold. As the frequency increases, more equivalent sources are needed but the accuracy of our technique is maintained. The candidate positions χ are chosen randomly on the surface of the object in the same manner as the previous algorithm. However, a minimum distance between any two equivalent sources is enforced to improve the condition number of the system; extremely close equivalent sources dominate the eigenvalues of the resulting system, adversely affecting its condition number. We choose a minimum distance of half the wavelength at any given frequency.

4.6 Global Solve

Once the scattering and interaction matrices are computed, and the sound source position has been decided, we solve for the global sound field and compute the outgoing equivalent source strengths of all the objects in the scene. The sound source can be a point source or a complex directional source (represented as a set of multipoles). We give an intuitive explanation here for a simple two-object scene and the detailed derivation can be found in Appendix 8.1. For a scene composed of multiple objects, we derive the same equation with the symbols having analogous meanings, as described in detail in Appendix 8.2. Assume the outgoing field in the scene is $\mathbf{C}$. This field when propagated through the scene, transferred via all possible object pairs using interaction matrix $\mathbf{G}$, generates an incoming field $\mathbf{GC}$ that, in addition to the source field $\mathbf{S}$, generates the total incoming field $(\mathbf{GC}+\mathbf{S})$ on the objects. This incoming field is then scattered by the object, via scattering matrix $\mathbf{T}$, to produce an outgoing field $\mathbf{T}(\mathbf{GC}+\mathbf{S})$. Under steady state, this outgoing field must equal $\mathbf{C}$. Mathematically, this can be written as

$$\mathbf{C} = \mathbf{T}(\mathbf{GC}+\mathbf{S}) \tag{17}$$

This yields a linear system for the outgoing source strengths for all objects:

$$(\mathbf{I}-\mathbf{TG})\mathbf{C} = \mathbf{TS}. \tag{18}$$

This linear system is solved for $\mathbf{C}$ at a regularly-sampled set of frequencies. This step has to be repeated for every sound source generating a distinct source field $\mathbf{S}$. In the absence of a source, the solution is identically zero.

4.7 Runtime Computation

At the end of the preprocessing stage, we obtain the outgoing equivalent source strengths for all objects at a regularly sampled set of frequencies corresponding to each sound source. During run-time, we use these strengths to compute the pressure field at any listener position $\mathbf{x}$:

$$p(\mathbf{x}) = \sum_{j=1}^{\kappa} C_{A_j}^{tr} \Phi_{A_j}^{out}(\mathbf{x}) + s(\mathbf{x}), \tag{19}$$

where κ is the number of objects in the scene, $C_{A_j}^{tr}$ and $\Phi_{A_j}^{out}$ are the strengths and multipoles of the outgoing equivalent sources for object A_j respectively, and $s\left(\mathbf{x}\right)$ is the field generated by the sound source. This computation is performed at a regularly-sampled set of frequencies and repeated for each source to compute a band-limited frequency response per source. Evaluating equation 19 for a new value of $\mathbf{x}$ is very efficient, allowing a moving listener to be handled naturally in real-time. Since the analytical expressions for multipoles of equivalent sources are used, the pressure can be evaluated at any position $\mathbf{x}$ in space and not necessarily at grid positions. Therefore, no spatial interpolation is required with our technique. Unlike grid-based approaches (such as FDTD), our equivalent source method is independent of the spatial discretiziation, resulting in a much smoother auralization for a moving listener.

Our technique allows auralization in a scene with multiple static sources and a moving listener. We can also handle the case of multiple moving sources and a static listener. First, we start with a scene with a static source and compute acoustic responses at multiple moving listeners using our runtime system. The principle of acoustic reciprocity states that we can reverse the sense of source and listener without changing the acoustic response [Pierce 1989, p. 195-199]. Using this principle, we now switch the roles of source and listeners while keeping the acoustic responses the same. This gives us acoustic response for the case of multiple moving sources with a static listener.

5. IMPLEMENTATION

In this section, we describe the implementation details of our technique. Typical parameter values used in our experiments are specified in Table II.

Implementation Details. The offset surface generation code is written in C++. When computing per-object transfer functions, outgoing scattered fields are computed on the offset surface (see Section 4.3) using an efficient GPU-based implementation of the ARD wave-solver [Raghuvanshi et al. 2009; Mehra et al. 2012]. The solver treats the scattering objects as rigid (with no transmission) and handles the material properties using perfectly matched layer interfaces. The remaining parts of the preprocessing stage, solving the linear system for per-object transfer functions, inter-object transfer functions, and equivalent source strengths, are implemented in MATLAB. The runtime code is implemented in C++, and has also been integrated with Valve's Source$^{\mathrm{TM}}$ engine, as demonstrated in the supplementary video.

The timing results for offset surface generation, the ARD solver, and runtime code are measured on a single core of a 4-core 2.80 GHz Xeon X5560 desktop with 4 GB of RAM and NVIDIA

GeForce GTX 480 GPU with 1.5 GB memory. Offset surface generation code takes $<$ 1sec for each object. The timing results for the MATLAB-based precomputation are measured on a 64-node CPU cluster (Xeon X5560 processor nodes, 8 cores, 2.80 GHz, 48 GB). Detailed statistics are provided in Table III. Precomputation for each frequency is performed in parallel over all the nodes (and individual cores) of the CPU cluster. Given more nodes on the cluster, the per-object, inter-object, and source-field computations can be further parallelized over all unique objects, all object-pairs, and all objects, respectively.

Due to the computational overhead of the precomputation stage, we treat band-limited sources that emit sound whose frequency range is bounded by maximum frequency $\nu_{\max}$ (see Table II), for the purpose of wave simulations (see Section 4.3). The pressure is computed at regularly sampled set of frequencies in the range $[0, \nu_{\max}]$ with a step size of $\Delta\nu$. The value of parameter $\Delta\nu$ is 4.08 Hz for concave, wall, rock, and parallel walls scenes and 2.04 Hz for desert, reservoir, and Christmas scenes.

Handling Ground Reflections. To handle ground reflections, we assume the ground to be an infinite plane. Similar to the image source method [Allen and Berkley 1979], we reflect our equivalent sources about the ground plane and multiply their source strengths by the (complex) reflection coefficient of the ground. Since sound waves traveling in air maintain their phase upon reflection from a hard surface, we do not need to invert the strengths of the equivalent sources. To incorporate last ground reflection, this step is performed after the "Global solve" step (see Section 4.6). In order to handle all levels of ground reflections before that, this step needs to be performed while computing the interaction matrices as well (see Section 4.4). More accurate physical models of ground reflection coefficient based on Darcy's law can also be used [Taraldsen and Jonasson 2011]. The assumption of infinite flat plane works very well for cases where the size of the ground perturbations is smaller than the minimum wavelength simulated (34cms for 1kHz). For cases where the ground contains terrain features that are much larger, like hillocks, these can be handled as separate objects in our ESM framework. Due to the increased number of objects, the precomputation time and runtime memory would increase, but the accuracy of our technique would be maintained.

Parameter	Value	Description
c	340 m/s	speed of sound
$\nu_{\max}$	1 kHz	highest frequency simulated
h	$c/2\nu_{\max} =$ 0.17 m	voxel resolution of distance field
δ	$5h =$ 0.85 m	inner offset distance
Δ	$8h =$ 1.36 m	outer offset distance
σ	15%	error threshold for scattering matrix
η	1%	error threshold for interaction matrix
M, N	2	order of incoming, outgoing multipoles resp.

Table II. : *Parameters used in our system.*

Spectral Extrapolation. The band-limited nature of the frequency responses of our technique necessitates a plausible extrapolation to higher frequencies at runtime. Prior work on interactive wave-based methods has shown that spectral extrapolation techniques can be used to produce plausible results for higher frequencies [Raghuvanshi et al. 2010]. However, using this method with our technique would incur an extra inverse FFT cost at every audio frame for time-domain processing. Therefore, we implemented a simple, fast extrapolation technique based on the edge-diffraction spectra [Svensson et al. 1999]. As observed, the typical edge diffraction spectra are roughly linear on a log-log scale. Hence, we first estimate a trend-line by a least-squares fit to the maximas of the log magnitude spectrum till $\nu_{\max}$. We then adjust for the trend, to create a flat response, by multiplying with the inverse of the trend on a log frequency scale. This adjusted response is replicated to higher frequencies and then multiplied by the trend again for the entire frequency range, yielding the final wide-band spectrum. If the trend-line has positive slope, indicating a high-pass response, we flatten the trend-line for frequencies beyond $\nu_{\max}$. This extrapolation technique does not change the spectrum up to $\nu_{\max}$.

We evaluate our spectral extrapolation technique by comparing the audio quality of the results with the wide-band spectrum produced by the Biot-Tolstoy-Medwin (BTM) technique (0-22 kHz) as the ground truth for the single, finite-edge scenario created in the right-angled wall scene. In the BTM method, edge diffraction impulse responses are computed by evaluating a time-domain line integral over the finite length of the edge. This is essentially based on Huygens theory, where a diffracting sound wave is modeled as a superposition of an infinite number of secondary point sources situated along the diffracting edge, each with different strengths and directivities. BTM has been shown to converge to the exact analytical solution for a simple scene like this [Svensson et al. 1999]. We use the MATLAB-based edge diffraction toolbox (http://www.iet.ntnu.no/ svensson/software/index.html) to generate the BTM results. As shown in the supplementary video, the final auralized audio generated by both the techniques on this scene sound similar.

This spectral extrapolation technique is approximate, and becomes exact in a specific, single-edge diffraction configuration. It does not guarantee accuracy on general scenes at high frequencies. While single-edge diffraction arises frequently in outdoor scenes, many other complex configurations also occur, such as double-diffraction and diffracted-reflection. Our extrapolation approach would be accurate in such cases only if the acoustic response is dominated by diffraction from a single edge. In other situations, we have observed that our extrapolation approach generates plausible results. A general spectral extrapolation approach for band-limited acoustic responses with guarantees on extrapolation error for arbitrary scenes, is an important area for future research.

Real-Time Auralization. The sound sources used in our implementation play a pre-recorded audio clip. Audio is rendered using FMOD, and is processed in frames of 1024 audio samples, at a sampling rate of 44.1 kHz. In-game ("dry") audio clips are preprocessed by computing a windowed short-time Fourier transform (STFT) on each frame (Blackman window). The STFTs are computed on audio frames after zero-padding by the maximum impulse response length to prevent aliasing artifacts. Real-time auralization is performed using overlap-add STFT convolutions. In each rendered frame, the dry audio frame for each source is multiplied in the frequency-domain with the corresponding frequency response. The results are then mixed, and an inverse FFT performed on the mixed audio. Finally, *overlap* from previous frames is added in, and overlap from the current frame is cached in a ring buffer. Frequency responses are updated asynchronously from the actual convolution processing. Spatialization is achieved by using a simplified spherical head model with two listeners, one for each ear. Richer spatialization that uses geometry information of an individual listener's ears, head, and shoulders can be modeled using *head related transfer functions* (HRTFs), and can be easily integrated in our approach, but is computationally more expensive.

Air Absorption. High-frequency sounds are absorbed more aggressively by the atmosphere than low frequencies. This frequency dependent air absorption is currently not modeled by our technique. However, it can be included as a post-processing step to our auralization pipeline. Since we compute complex frequency responses containing phase information, propagation delays are modeled. These delays yield the propagation distances which can be used to calculate and apply a per-frequency attenuation filter in frequency domain to model atmospheric absorption.

Scene	#objs.	# freq.	# srcs	wave sim.	per-object	inter-object	source field	global solve	wall clk	# eq. srcs	eval.	storage
				(total,	(per freq)	(per freq)	(per freq,	(per freq,	time	(total,	(total,	(total,
				per obj.)			per src)	per src)		per src)	per src)	fixed + per src)
Concave	1	250	1	80 min	51 min	NA	1 min	0.1 min	132 min	0.1 M	3 ms	(1 + 4) MB
Wall	1	250	1	50 min	101 min	NA	3 min	0.1 min	154 min	0.1 M	4 ms	(2 + 5) MB
Rock	1	250	1	80 min	87 min	NA	1 min	0.1 min	168 min	0.4 M	10 ms	(4 + 11) MB
Parallel	2*	250	1	50 min	101 min	13 min	6 min	1 min	171 min	0.2 M	8 ms	(4 + 10) MB
Desert	4*+2*	500	3	180 min	196 min	98 min	9 min	26 min	509 min	1.1 M	26 ms	(12 + 33) MB
Reservoir	4*+1	500	2	146 min	224 min	63 min	7 min	15 min	455 min	1.3 M	33 ms	(15 + 41) MB
Christmas	2*+2*+1	500	2	297 min	301 min	71 min	7 min	18 min	694 min	1.5 M	38 ms	(18 + 47) MB

Table III. : ***Performance statistics.*** *Abbreviations are as follows : "#objs." denotes the number of objects in the scene, "#freq." is the number of frequency samples in the range [0-1 kHz] and "#srcs" is the number of sound sources. For the precomputation stage, the term "wave sim." is the total simulation time of the numerical wave solver for all frequencies, "per-object" denote the compute time for the per-object transfer function for all unique objects and "inter-object" denote the compute time for inter-object transfer functions for all object pairs, "source-field" is time to express each sound source in terms of incoming multipoles for all objects, and "global-solve" is time to compute equivalent source strengths for all objects. The "wave sim." step is parallelized over all unique objects whereas the remaining precomputation steps are parallelized over all frequencies. The term "wall-clk time" is the total wall-clock time computed by uniformly distributing all the parallel processes over all the cores of the 64-node cluster with 8 cores per node (512 cores in total). At runtime, the total number of equivalent sources "# eq. srcs" (in million M), performance "eval." and storage requirement "storage" (fixed and per source cost) for all objects for all frequencies are also specified. For column "#objs.", notation* $a^* + b^*$ *denotes that first object has been instanced* a *times and second object instanced* b *times, but their per-object transfer functions are computed only once for each unique object.*

6. RESULTS AND ANALYSIS

In this section, we present the results of our technique on different scenarios, provide error analysis and compare it with prior work.

6.1 Scenarios

We have considered a variety of scenes for testing our technique. For auralizations corresponding to the scenes discussed below, refer to the supplementary video.

Single Object. We considered various objects having different scattering characteristics: rocks, a wall, and a concave reflector. The rocks scatter sound in all directions (see Figure 4). We show magnitude of the scattered sound field for the wall generated by our technique and by BEM in Figure 5. As shown in the figure, the wall strongly scatters sound in the direction perpendicular to itself. As a more challenging scene, the magnitude of scattered sound field for a concave reflector is also shown. The reflector generates significant interference effects, resulting in caustic formation in the focal region. This is clearly captured by our technique, as the high amplitude (red) region in the figure, showing that our technique is able to approximate the phase of the scattered wave field with reasonable accuracy. The relative error, defined in Figure 6 caption, between the total pressure fields generated by our technique and by the BEM technique is less than 2% for the wall and 5% for the concave reflector.

Parallel Buildings. This scene consists of two buildings situated parallel to one another. We show two walkthroughs of this scene, with a flying helicopter, and a person speaking, respectively. As the helicopter moves behind a building, diffraction leads to a distinct low-pass occlusion effect. The two walls trap sound between them, producing high-order reflections, so that the volume of someone talking between the buildings is markedly louder than someone standing even slightly to the side.

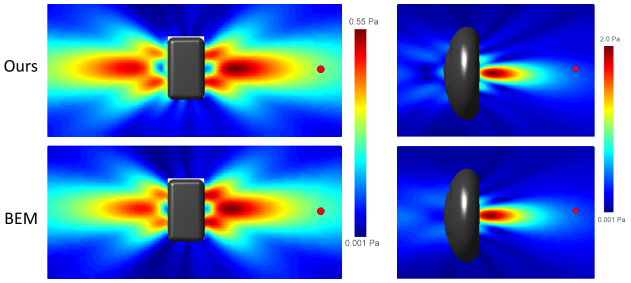


Figure 5: *We show the scattering behavior of a wall (*$2.3m$ *x* $4.5m$ *x* $3.7m$*) and a concave reflector (diameter* $8m$*, thickness* $1.2m$*) at* $160Hz$ *using our technique (top row) and BEM (bottom row). The sound source is shown with a red dot.*

Desert. This is a large scene with three sound sources spread throughout the scene: a jeep, a bird, and a radio. As the listener walks through the scene, the sound received from the various sources changes significantly depending on whether or not the listener is in the line-of-sight of the source(s). We also specifically demonstrate the effect of second-order diffracted occlusion of the jeep sound around two buildings.

Christmas Town. This scene demonstrates sound propagation in a village with many houses, a church, a bell tower and large buildings. It shows reflection from buildings, diffraction around houses, sound propagation over large distances from the bell tower, and reflections between two parallel buildings, for multiple sources.

Reservoir. We show that our technique can be integrated with an existing game (*Half-Life 2*) to generate realistic wave acoustic effects in a large, outdoor game map. Our method is the first wave-based sound propagation technique that can accurately model wave phenomena such as diffraction behind the rocks and scattering around buildings over large distances on such a scene in real-time.

In our single-object examples, helicopter behind rock, the occluder is placed in isolation without any surrounding objects. Due to the lack of reflections from surrounding objects, and the fact that high frequencies do exhibit quite sharp shadows, our diffraction and occlusion effects may sound exaggerated, compared to real life.

6.2 Error Analysis

Figure 6 shows the convergence of our method as the error threshold σ decreases. Since the number of outgoing equivalent sources is inversely proportional to σ, it also shows convergence of the technique with increasing number of outgoing equivalent sources. We also plot the variation in the number of outgoing equivalent sources with frequency to achieve given error thresholds (see Figure 7). In Figure 8, we compare the results of our ESM technique with the reference wave solver ARD [Raghuvanshi et al. 2009], BEM [Cheng and Cheng 2005], and FMM-BEM [Liu et al. 2009; Gumerov and Duraiswami 2009] techniques on a spatial grid of listeners at different frequencies for the two parallel walls scene. We used the state-of-the-art FastBEM simulator (http://www.fastbem.com/) for generating BEM and FMM-BEM results up to the maximum frequency possible (358 Hz). This scene is acoustically very complex, even though individual objects seem simple, since there are multiple orders of interaction happening between these two walls. Our

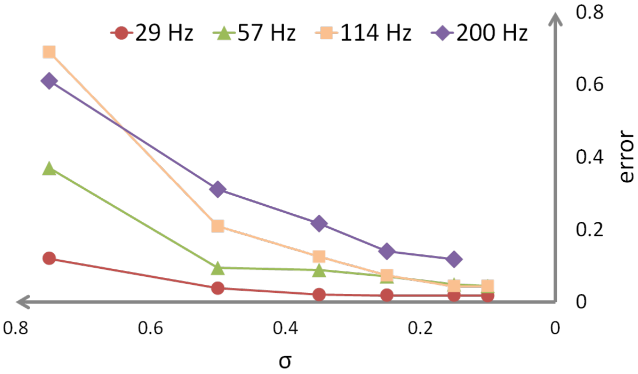


Figure 6: ***Convergence**: We show the variation of error $\|P_{ref} - P_{ESM}\|^2/\|P_{ref}\|^2$ between the reference wave solver and our technique for varying values of scattering matrix error threshold σ for the two parallel walls scene (fixed $\eta = 1\%$). P_{ref} and P_{ESM} are vectors consisting of (complex) pressure values at all the receiver locations as given by the reference wave solver and our technique, respectively. The receivers are placed on a XY grid for this scene.*

approach handles such effects accurately and produces good approximations, up to the user-specified error thresholds, while reducing memory usage by orders of magnitude (see Table IV).

6.3 Computational Complexity

Consider a scene with κ objects. To perform analysis for frequency ν, let the number of offset surface samples, incoming equivalent sources and outgoing equivalent sources at this frequency be n, Q and P respectively. We assume that all objects have equal volume u.

Pre-processing.
Scattering Matrix: For each of the QM^2 incoming multipoles of an object, wave simulations are performed and a dense linear system of size $n \times PN^2$ is solved to find the object's scattering matrix. The cost for each simulation is $u \log u$, and the cost of solving the linear system[1] is nP^2N^4. Hence, the total cost is $O\left(\kappa QM^2\left(nP^2N^4 + u\log u\right)\right)$.

Interaction Matrix: For every pair of objects, PN^2 linear systems of size $n \times QM^2$ need to be solved to find the interaction matrix. In total, we have κ^2 object pairs. The cost of evaluating analytical expressions for multipole pressure is $O(1)$ each, and is dominated by the cost of solving the linear systems. Hence the total cost is $O\left(\kappa^2PN^2nQ^2M^4\right)$.

The size of these linear systems vary linearly with n, which in turn varies quadratically with frequency (see Section 4.5). Thus, ensuring a few hours precomputation time on a small computational cluster (see Table 3) limits our technique to 1-2 kHz on typical outdoor scenes.

Computing Strengths: The incoming field produced by each sound source is represented in terms of the incoming equivalent sources of the objects. This requires solving κ linear system of size $n \times QM^2$ resulting in cost $O\left(\kappa nQ^2M^4\right)$. The size of the final linear system for finding outgoing equivalent source strengths for all objects in response to a sound source is $\kappa PN^2 \times \kappa PN^2$. Solving it takes $O\left(\kappa^3P^3N^6\right)$ time.

It follows that the total pre-processing cost at frequency ν thus scales as

$$O(\kappa QM^2(nP^2N^4 + u\log u \\ + \kappa PN^2nQM^2 + nQM^2) + \kappa^3P^3N^6)$$

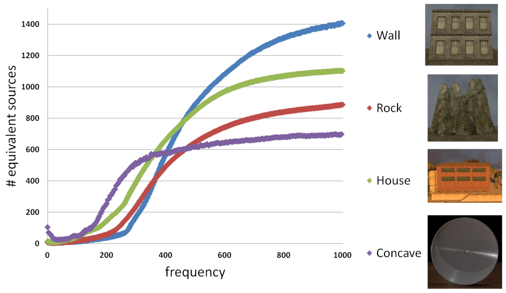


Figure 7: *Variation of the number of outgoing equivalent sources with frequency, for four different objects. As the frequency increases (wavelength decreases), surface details of the size of the wavelength increase the complexity of the sound field. This results in a larger number of equivalent sources. When all the details of the object are captured, increasing the frequency has little effect and the number of equivalent sources begin to stabilize. Error thresholds are $\sigma = 15\%$ and $\eta = 1\%$.*

Runtime. At runtime, we evaluate equation (19), which takes $O\left(\kappa PN^2\right)$ at frequency ν. The runtime memory requirement consists of positions (3 floats) and (complex-valued) strengths (2 floats) of equivalent sources, which comes out to be $\kappa(3P{+}2PN^2)$ at frequency ν.

The precomputation and runtime complexity and memory requirement depend on the number of equivalent sources P, which scales quadratically with frequency, in an asymptotic sense. However, for practical objects and frequencies up to 1 kHz, we observed linear scaling of equivalent sources with frequency, as shown in Figure 7.

We have to compute the pressure at the listener position over a regularly sampled set of frequencies in the range $[0, \nu_{max}]$ with a step of $\Delta\nu$. The total number of frequency samples becomes $\nu_{max}/\Delta\nu$. Thus, the above expressions are summed for all frequency samples in this range to give the total computational complexity and memory requirement. Since the computational cost and runtime memory scales with the multipole order, we limit equivalent sources to monopoles and dipoles, i.e., N=M=2. Low multipole orders (N, M) result in a larger number of equivalent sources for satisfying the same error thresholds. However, since we place only as many equivalent sources as required, low multipole order does not effect the quality of the final result. The theoretical runtime memory requirements for other wave-solvers are discussed in Appendix 8.3. We also compare the runtime memory requirements of these solvers with our technique on a variety of scenes (see Table IV).

Scene	air. vol.	surf. area	FDTD	ARD	BEM/ FMM	Ours
Concave	$(85m)^3$	107 m^2	33 TB	0.9 TB	0.5 GB	5 MB
Wall	$(85m)^3$	71 m^2	33 TB	0.9 TB	0.3 GB	7 MB
Rock	$(85m)^3$	159 m^2	33 TB	0.9 TB	0.8 GB	15 MB
Parallel	$(85m)^3$	142 m^2	33 TB	0.9 TB	0.7 GB	14 MB
Desert	$(180m)^3$	1626 m^2	625 TB	17 TB	15 GB	45 MB
Reservoir	$(180m)^3$	950 m^2	625 TB	17 TB	9 GB	56 MB
Christmas	$(180m)^3$	2953 m^2	625 TB	17 TB	27 GB	65 MB

Table IV. : *Runtime memory requirements per source, for FDTD [Taflove and Hagness 2005], ARD [Raghuvanshi et al. 2009], BEM/FMM-BEM [Liu et al. 2009], and our ESM technique with error thresholds $\sigma = 15\%$, $\eta = 1\%$ at maximum simulation frequency $\nu_{\max} = 1018Hz$. Refer to Section 6.3 and Appendix 8.3 for more details.*

Our precomputation step is computationally heavy and takes a few hours to run on a CPU-cluster (Table III). But this step is trivially parallel and could be performed easily on cheap and widely available cloud computing resources, such as Amazon EC2. Moreover, our current implementation is in MATLAB and we expect 10x improvement with an optimized C++ implementation.

[1] To solve a dense linear system of size m x n $(m > n)$, the cost is mn^2

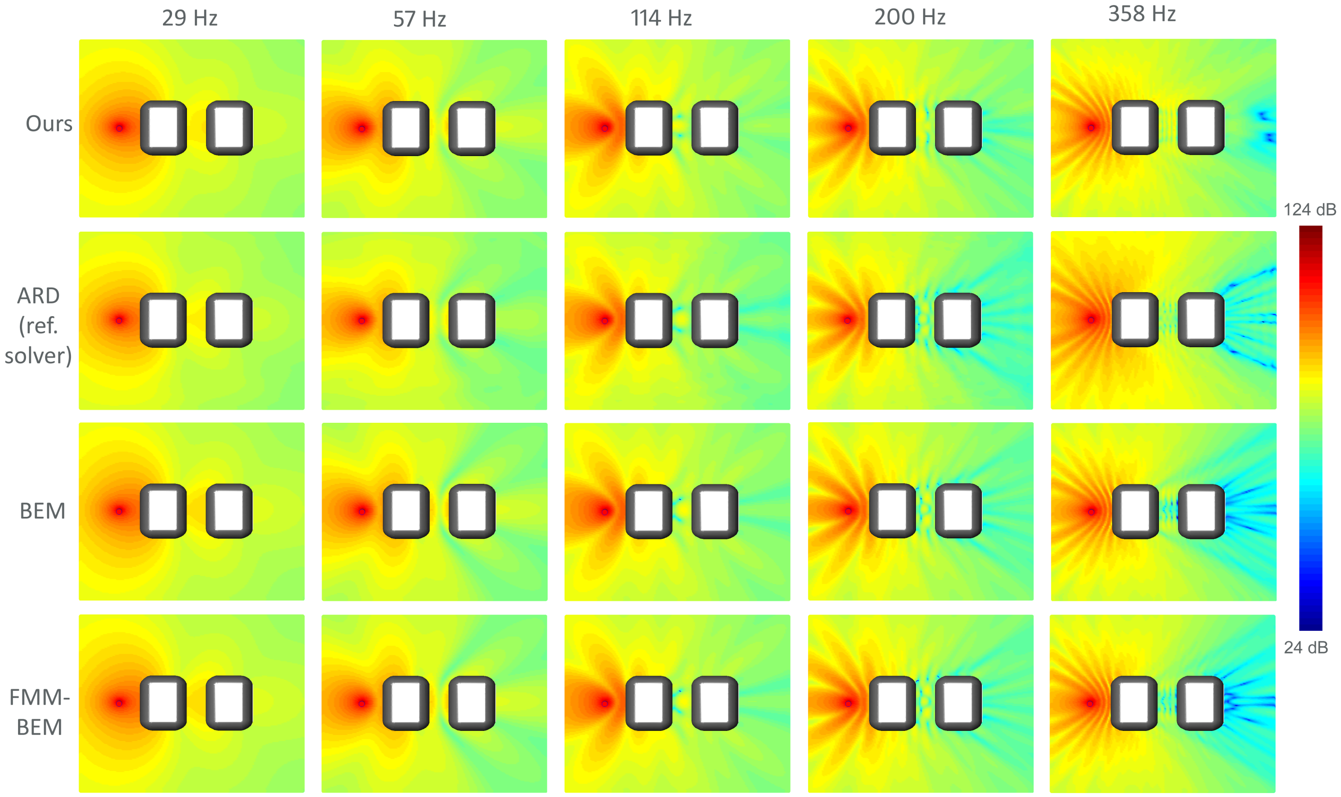


Figure 8: *Comparison between the magnitude of the total pressure field (in Sound Pressure Level SPL, units dB) computed by our ESM, the reference wave-solver ARD, BEM and FMM-BEM techniques for the two parallel walls scene on a XY cutview grid of listeners. The red point denotes the position of the sound source. The error (defined in Figure 6 caption) between the ARD-ESM fields is $< 3\%$, BEM-ESM fields is $< 5\%$ and FMM BEM-ESM fields is $< 5\%$ for the frequencies shown.*

6.4 Comparison with Prior Interactive Techniques

Our usage of equivalent sources for sound propagation is in a similar vein to prior work [James et al. 2006], where the authors represent arbitrary outgoing radiation field from a single, geometrically complex object. Our work differs primarily in three regards: First, we model *mutual interactions* between objects in arbitrary scenes using inter-object transfer functions, accounting for high-order interactions, such as echoes and multiple diffraction. Secondly, we model acoustic scattering from objects (as opposed to radiation), which requires an approximation of both the *incoming and outgoing* pressure fields for an object. Finally, our outgoing equivalent sources are chosen to satisfy multiple outgoing scattered fields as opposed to a single radiation field.

The problem of real-time acoustic scattering has been previously addressed using GPUs [Tsingos et al. 2007]. First-order scattering effects are incorporated, but acoustic interactions between objects are not modeled. In contrast, our work can handle all orders of interactions between the objects using inter-object transfer functions.

A recent technique for interactive acoustics based on wave simulation was proposed in Raghuvanshi et al. [2010], which relies on sampling the volume of the scene, and uses a perceptual compression specific to indoor scenes. The runtime memory requirement of their technique (per source) on our scenes (assuming a spatial sampling of $1m$) is 187 MB for the parallel walls and 1.8 GB for the reservoir scene. This technique is complimentary to our approach; it works best in indoor spaces with a lot of geometric clutter but limited volume, while our technique is better suited to outdoor spaces with well-separated objects. In fact, it would be quite natural to integrate this method with ours, with the indoor and outdoor propagation models coupled through transport operators defined on doors and windows.

7. CONCLUSIONS AND DISCUSSION

We have presented a novel wave-based sound propagation algorithm that captures acoustic effects such as high-order diffraction and scattering, using an equivalent source formulation. As a result, our technique can perform accurate sound propagation on large, open scenes in real-time, has a small memory footprint, and allows flexible efficiency-to-accuracy tradeoffs. Compared to directly storing and convolving wave-solver solutions for auralization, we reduce the memory usage more than 100 times.

Our approach is currently limited to static scenes, due to the computational cost of recomputing inter-object transfers as objects move. We would like to combine our approach with Fast Multipole Method (FMM)[Liu et al. 2009; Gumerov and Duraiswami 2009] to accelerate inter-object transfer evaluations using progressive far-field approximations. Moreover, real-time performance could be achieved by further using GPU-based dense linear solvers. The incoming field strength computation for a moving source is similar to inter-object transfer. Thus, the combination of FMM and GPU-based computations could enable dynamic sources along with a moving listener. Also, our current runtime system does not model Doppler effect, which we would like to address in future work.

Currently, our sound sources emit a pre-recorded audio clip. An interesting direction of future research would be to integrate acoustic radiators based on mechanical physical models [Zheng and James 2010; Chadwick et al. 2009] as potential sound sources, thus enabling physically-based real-time sound synthesis and propagation.

The computational complexity and runtime memory requirement of our technique scale linearly with number of frequency samples which in turn scales linearly with the scene size (number of frequency samples $\propto$ length of impulse response $\propto$ scene size). Thus, our technique can easily handle scenes that are hundreds of meters wide. However, for massive outdoor scenes that span kilometers, the runtime memory requirement would become too high (GBs per source). We plan to address this in future by using FMM-based far field approximations to reduce the number of equivalent sources.

Our precomputation depends heavily on the maximum simulation frequency thereby limiting it to 1-2 kHz. This behavior is consistent with other wave-based techniques like BEM and FDTD, which are also computationally limited to a few kHz. Geometric approximations become quite accurate for outdoor scenes at higher frequencies because buildings and terrain have much larger dimensions than the wavelength of $17cm$ at 2kHz. Thus, hybridization of our technique with geometric methods could lead to accurate wideband propagation techniques for open scenes. Hybridization is an active area of research in acoustics [Southern et al. 2011].

REFERENCES

ABRAMOWITZ, M. AND STEGUN, I. 1964. *Handbook of Mathematical Functions*, Fifth ed. Dover, New York.

ALLEN, J. B. AND BERKLEY, D. A. 1979. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America 65,* 4, 943–950.

ANTANI, L., CHANDAK, A., TAYLOR, M., AND MANOCHA, D. 2012. Direct-to-indirect acoustic radiance transfer. *IEEE Transactions on Visualization and Computer Graphics 18,* 2, 261–269.

CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. In *SIGGRAPH Asia '09*. ACM, New York, NY, USA, 1–10.

CHENG, A. AND CHENG, D. 2005. Heritage and early history of the boundary element method. *Engineering Analysis with Boundary Elements 29,* 3 (Mar.), 268–302.

DOICU, A., EREMIN, Y. A., AND WRIEDT, T. 2000. *Acoustic and Electromagnetic Scattering Analysis Using Discrete Sources*, 1st ed. Academic Press.

FAIRWEATHER, G. 2003. The method of fundamental solutions for scattering and radiation problems. *Engineering Analysis with Boundary Elements 27,* 7 (July), 759–769.

FUNKHOUSER, T., CARLBOM, I., ELKO, G., PINGALI, G., SONDHI, M., AND WEST, J. 1998. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proc. of ACM SIGGRAPH*. 21–32.

GUMEROV, N. A. AND DURAISWAMI, R. 2009. A broadband fast multipole accelerated boundary element method for the three dimensional Helmholtz equation. *The Journal of the Acoustical Society of America 125,* 1, 191–205.

HOBSON, E. W. 1955. *The Theory of Spherical and Ellipsoidal Harmonics*. Cambridge University Press, New York, NY, USA.

JAMES, D. L., BARBIČ, J., AND PAI, D. K. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06. ACM, New York, NY, USA, 987–995.

KROKSTAD, A., STROM, S., AND SORSDAL, S. 1968. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration 8,* 1 (July), 118–125.

KROPP, W. AND SVENSSON, P. U. 1995. Application of the time domain formulation of the method of equivalent sources to radiation and scattering problems. *Acta Acustica united with Acustica 81,* 6, 528–543.

LENTZ, T., SCHRÖDER, D., VORLÄNDER, M., AND ASSENMACHER, I. 2007. Virtual reality system with integrated sound field simulation and reproduction. *EURASIP J. Appl. Signal Process. 2007,* 1, 187.

LIU, Q. H. 1997. The PSTD algorithm: A time-domain method combining the pseudospectral technique and perfectly matched layers. *The Journal of the Acoustical Society of America 101,* 5, 3182.

LIU, Y., SHEN, L., AND BAPAT, M. 2009. Development of the fast multipole boundary element method for acoustic wave problems. In *Recent Advances in Boundary Element Methods*. Springer, 287–303.

MANOCHA, D., CALAMIA, P., LIN, M. C., MANOCHA, D., SAVIOJA, L., AND TSINGOS, N. 2009. Interactive sound rendering. In *ACM SIGGRAPH 2009 Courses*. ACM, New York, NY, USA, 15:1–15:338.

MEHRA, R., RAGHUVANSHI, N., SAVIOJA, L., LIN, M. C., AND MANOCHA, D. 2012. An efficient gpu-based time domain solver for the acoustic wave equation. *Applied Acoustics 73,* 2, 83 – 94.

OCHMANN, M. 1995. The source simulation technique for acoustic radiation problems. *Acustica 81*, 512–527.

OCHMANN, M. 1999. The full-field equations for acoustic radiation and scattering. *The Journal of the Acoustical Society of America 105,* 5, 2574–2584.

PAVIC, G. 2006. A technique for the computation of sound radiation by vibrating bodies using multipole substitute sources. *Acta Acustica united with Acustica 92*, 112–126(15).

PIERCE, A. D. 1989. *Acoustics: An Introduction to Its Physical Principles and Applications*. Acoustical Society of America.

RAGHUVANSHI, N., NARAIN, R., AND LIN, M. C. 2009. Efficient and Accurate Sound Propagation Using Adaptive Rectangular Decomposition. *IEEE Transactions on Visualization and Computer Graphics 15,* 5, 789–801.

RAGHUVANSHI, N., SNYDER, J., MEHRA, R., LIN, M. C., AND GOVINDARAJU, N. K. 2010. Precomputed Wave Simulation for Real-Time Sound Propagation of Dynamic Sources in Complex Scenes. *SIGGRAPH 2010 29,* 3 (July).

SAKAMOTO, S., USHIYAMA, A., AND NAGATOMO, H. 2006. Numerical analysis of sound propagation in rooms using finite difference time domain method. *Journal of the Acoustical Society of America 120,* 5, 3008.

SAVIOJA, L. 2010. Real-Time 3D Finite-Difference Time-Domain Simulation of Mid-Frequency Room Acoustics. In *13th International Conference on Digital Audio Effects (DAFx-10)*.

SILTANEN, S., LOKKI, T., KIMINKI, S., AND SAVIOJA, L. 2007. The room acoustic rendering equation. *The Journal of the Acoustical Society of America 122,* 3 (Sept.), 1624–1635.

SILTANEN, S., LOKKI, T., AND SAVIOJA, L. 2009. Frequency domain acoustic radiance transfer for real-time auralization. *Acta Acustica united with Acustica 95,* 1, 106–117.

SOUTHERN, A., SILTANEN, S., AND SAVIOJA, L. 2011. Spatial room impulse responses with a hybrid modeling method. In *Audio Engineering Society Convention 130*.

SVENSSON, U. P., FRED, R. I., AND VANDERKOOY, J. 1999. An analytic secondary source model of edge diffraction impulse responses . *Acoustical Society of America Journal 106*, 2331–2344.

TAFLOVE, A. AND HAGNESS, S. C. 2005. *Computational Electrodynamics: The Finite-Difference Time-Domain Method, Third Edition*, 3 ed. Artech House Publishers.

TAKALA, T. AND HAHN, J. 1992. Sound rendering. *SIGGRAPH Comput. Graph. 26,* 2 (July), 211–220.

TARALDSEN, G. AND JONASSON, H. 2011. Aspects of ground effect modeling. *The Journal of the Acoustical Society of America 129,* 1, 47–53.

TAYLOR, M. T., CHANDAK, A., ANTANI, L., AND MANOCHA, D. 2009. RESound: interactive sound rendering for dynamic virtual environments. In *MM '09: ACM Multimedia*. ACM, New York, NY, USA, 271–280.

THOMPSON, L. L. 2006. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America 119,* 3, 1315–1330.

THOMPSON, L. L. AND PINSKY, P. M. 2004. *Acoustics*. John Wiley & Sons, Ltd.

TSINGOS, N. 2009. Pre-computing geometry-based reverberation effects for games. In *35th AES Conference on Audio for Games*.

TSINGOS, N., DACHSBACHER, C., LEFEBVRE, S., AND DELLEPIANE, M. 2007. Instant Sound Scattering. In *Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering)*.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *SIGGRAPH '01*. ACM, New York, NY, USA, 545–552.

TSINGOS, N. AND GASCUEL, J. D. 1997. A general model for the simulation of room acoustics based on hierachical radiosity. In *ACM SIGGRAPH 97*. SIGGRAPH '97. ACM, New York, NY, USA.

VORLANDER, M. 1989. Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm. *Journal of the Acoustical Society of America 86,* 1, 172–178.

WATERMAN, P. C. 2009. T-matrix methods in acoustic scattering. *The Journal of the Acoustical Society of America 125,* 1, 42–51.

YEE, K. 1966. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation 14,* 3 (May), 302–307.

ZHENG, C. AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers*. ACM, New York, NY, USA, 1–13.

ZIENKIEWICZ, O. C., TAYLOR, R. L., AND NITHIARASU, P. 2006. *The finite element method for fluid dynamics*, 6 ed. Butterworth-Heinemann.

8. APPENDIX

8.1 Two-object Steady State Field

We describe in detail the way we compute the equivalent source strengths for a scene composed of two objects. Consider a scene with objects A and B and a sound source s. Let the incoming field multipoles for A and B be Φ_A^{in} and Φ_B^{in}, respectively. Similarly, let the multipoles for the outgoing field for A and B be Φ_A^{out} and Φ_B^{out}, respectively. The scattering matrices for A and B are T_A and T_B, respectively. Let the interaction matrices for the objects be G_A^B and G_B^A, respectively. First of all, we express the incoming field produced by sound source s on objects A and B in terms of their incoming field multipoles :

$$s_A^{in} = \sum_{i=1}^{Q}\sum_{k=1}^{M^2} a_{ik}\varphi_{ik}^{in} = S_A^{tr}\Phi_A^{in}; \;\; s_B^{in} = \sum_{i=1}^{Q}\sum_{k=1}^{M^2} b_{ik}\varphi_{ik}^{in} = S_B^{tr}\Phi_B^{in}.$$

Now assume that the steady state outgoing field of object A and B is P_A^{out} and P_B^{out} respectively.

$$P_A^{out} = \sum_{j=1}^{P}\sum_{h=1}^{N^2} c_{jh}^{A}\varphi_{jh}^{out} = C_A^{tr}\Phi_A^{out}; \qquad (20)$$

$$P_B^{out} = \sum_{j=1}^{P}\sum_{h=1}^{N^2} c_{jh}^{B}\varphi_{jh}^{out} = C_B^{tr}\Phi_B^{out}. \qquad (21)$$

The outgoing field of one object becomes the incoming field for the other object. Exploiting the linearity of the inter-object transfer function and (14), we find the incoming field for B produced by the outgoing field of A as

$$\widehat{P}_B^{in} = g_A^B(C_A^{tr}\Phi_A^{out}) = C_A^{tr}G_A^B\Phi_B^{in}.$$

Similarly, we find the incoming field for A produced by the outgoing field of B as

$$\widehat{P}_A^{in} = g_B^A(C_B^{tr}\Phi_B^{out}) = C_B^{tr}G_B^A\Phi_A^{in}.$$

The total incoming fields on objects A and B are given by

$$\begin{aligned} P_A^{in} &= s_A^{in} + \widehat{P}_A^{in} = S_A^{tr}\Phi_A^{in} + C_B^{tr}G_B^A\Phi_A^{in}; \\ P_B^{in} &= s_B^{in} + \widehat{P}_B^{in} = S_B^{tr}\Phi_B^{in} + C_A^{tr}G_A^B\Phi_B^{in}. \end{aligned}$$

Applying the linearity of per-object transfer function f and using (9), we get outgoing pressure P_{out}^A and P_{out}^B due to the scattering of incoming fields by the objects as

$$P_A^{out} = f(P_A^{in}) = (S_A^{tr}T_A + C_B^{tr}G_B^AT_A)\Phi_A^{out}, \qquad (22)$$

$$P_B^{out} = f(P_B^{in}) = (S_B^{tr}T_B + C_A^{tr}G_A^BT_B)\Phi_B^{out}. \qquad (23)$$

In steady state, this outgoing pressure should match the outgoing pressure we started with. Equating (22) with (20), and (23) with (21), we get

$$\begin{aligned} C_A^{tr} &= S_A^{tr}T_A + C_B^{tr}G_B^AT_A; \\ C_B^{tr} &= S_B^{tr}T_B + C_A^{tr}G_A^BT_B. \end{aligned}$$

Combining the above two equations, and rearranging, we obtain

$$\begin{bmatrix} C_A \\ C_B \end{bmatrix} = \begin{bmatrix} T_A^{tr} & 0 \\ 0 & T_B^{tr} \end{bmatrix} \left(\begin{bmatrix} 0 & (G_B^A)^{tr} \\ (G_A^B)^{tr} & 0 \end{bmatrix} \begin{bmatrix} C_A \\ C_B \end{bmatrix} + \begin{bmatrix} S_A \\ S_B \end{bmatrix} \right).$$

In other words,

$$(\mathbf{I} - \mathbf{TG})\mathbf{C} = \mathbf{TS}, \qquad (24)$$

which is a linear system $Ax = b$. We solve this linear system to get the outgoing equivalent source strengths $\mathbf{C}$. At runtime, the outgoing scattered field at any listener position $\mathbf{x}$ is given by

$$p(\mathbf{x}) = C_A^{tr}\Phi_A^{out}(\mathbf{x}) + C_B^{tr}\Phi_B^{out}(\mathbf{x}). \qquad (25)$$

The total pressure field becomes

$$p(\mathbf{x}) = C_A^{tr}\Phi_A^{out}(\mathbf{x}) + C_B^{tr}\Phi_B^{out}(\mathbf{x}) + s(\mathbf{x}). \qquad (26)$$

8.2 Multiple Objects Steady State Field

For a scene with κ objects, $A_1, A_2, ..., A_\kappa$, equation (24) remains the same except the vectors and matrices are generalized for κ objects. The total pressure field becomes

$$p(\mathbf{x}) = \sum_{j=1}^{\kappa} C_{A_j}^{tr}\Phi_{A_j}^{out}(\mathbf{x}) + s(\mathbf{x}). \qquad (27)$$

8.3 Computational Complexity

BEM. The storage requirements of BEM depends on the total surface area S of the objects in the scene and the number of frequency samples $\nu_{\max}/\Delta\nu$. Assuming BEM places τ samples per wavelength (usually $\tau = 12$), the number of BEM elements placed on the object's surface at frequency sample ν_i is equal to $S\tau^2\nu_i^2/c^2$. The total number of BEM elements for all the frequency samples is equal to $S\tau^2\nu_{\max}^3/(3c^2\Delta\nu)$, where each element is specified by its position (3 floats) and four complex amplitudes corresponding to pressure and its gradient (2 floats each). Total memory requirement of storing the simulation results becomes

$$11S\tau^2\nu_{\max}^3/(3c^2\Delta\nu).$$

ARD and FDTD. The runtime memory requirements of ARD and FDTD are equal to the number of grid cells in the spatial discretization of the entire volume of the scene and the number of timesteps in the simulation. Assuming volume of the scene to be V, the grid size h, the maximum frequency $\nu_{\max}$, the speed of sound c, and the number of samples per wavelength τ (equal to 3 for ARD and 10 for FDTD), the number of grid cells are $(\tau\nu_{\max}/c)^3V$. The total number of time samples to store is at least twice the number of samples in the frequency domain. The total memory requirement of storing the simulation results for these techniques is thus

$$2\tau^3\nu_{\max}^4V/(c^3\Delta\nu).$$