

SynCoPation: Interactive Synthesis-Coupled Sound Propagation

Atul Rungta , Carl Schissler , Ravish Mehra , Chris Malloy , Ming Lin *Fellow, IEEE* , and Dinesh Manocha *Fellow, IEEE*

Department of Computer Science, University of North Carolina at Chapel Hill

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



Fig. 1. Our interactive sound synthesis-propagation technique has been integrated in the UnityTM game engine. We demonstrate sound effects generated by our system on a variety of scenarios: (a) Cathedral, (b) Tuscany, and (c) Game scene. In Cathedral scene, the bell sounds are synthesized and propagated in the indoor space at interactive rates; In Tuscany scene, the chime sounds are synthesized and propagated in an outdoor space; In the last scene, sounds generated by the barrel hitting the ground are synthesized and propagated at interactive rates.

Abstract— Recent research in sound simulation has focused on either sound synthesis or sound propagation, and many standalone algorithms have been developed for each domain. We present a novel technique for coupling sound synthesis with sound propagation to automatically generate realistic aural content for virtual environments. Our approach can generate sounds from rigid-bodies based on the vibration modes and radiation coefficients represented by the single-point multipole expansion. We present a mode-adaptive propagation algorithm that uses a perceptual Hankel function approximation technique to achieve interactive runtime performance. The overall approach allows for high degrees of dynamism - it can support dynamic sources, dynamic listeners, and dynamic directivity simultaneously. We have integrated our system with the Unity game engine and demonstrate the effectiveness of this fully-automatic technique for audio content creation in complex indoor and outdoor scenes. We conducted a preliminary, online user-study to evaluate whether our Hankel function approximation causes any perceptible loss of audio quality. The results indicate that the subjects were unable to distinguish between the audio rendered using the approximate function and audio rendered using the full Hankel function in the Cathedral, Tuscany, and the Game benchmarks.

Index Terms—Sound Synthesis, Sound Propagation, Physically-based Modeling

1 INTRODUCTION

Realistic sound simulation can increase the sense of presence for users in games and VR applications [10, 36]. Sound augments both the visual rendering and tactile feedback, provides spatial cues about the environment, and improves the overall immersion in a virtual environment, e.g., playing virtual instruments [29, 35, 32, 49] or walking interaction [12, 21, 39, 47, 44]. Current game engines and VR systems

tend to use pre-recorded sounds or reverberation filters, which are typically manipulated using digital audio workstations or MIDI sequencer software packages, to generate the desired audio effects. However, these approaches are time consuming and unable to generate appropriate auditory cues or sound effects that are needed for virtual reality. Further, many sound sources have a very pronounced directivity patterns which get propagated into the environment. And as these sources move, so do their directivities. Thus, it is important to model these time-varying, dynamic directivities propagating in the environment to make sure the audio-visual correlation is maintained and the presence not disrupted.

Recent trend has been on development of physically-based sound simulation algorithms to generate realistic effects. At a broad level, they can be classified into sound synthesis and sound propagation algorithms. Sound synthesis techniques [46, 22, 50, 7, 51, 45] model the generation of sound based on vibration analysis of the object resulting in modes of vibration that vary with frequency. However, these techniques only model sound propagation in free-space and do not account for the acoustics effects caused by interaction of sound waves with the objects in the environment. On the other hand, sound propagation techniques [13, 48, 17, 18] model the interaction of sound waves with the objects in environment, but assume pre-recorded or pre-synthesized audio clips as input. Therefore, current sound simulation algorithms ignore the dynamic interaction between the processes of sound synthesis, emission (radiation), and propagation, resulting in inaccurate (or non-plausible) solutions for the underlying physical

- Atul Rungta, Carl Schissler, Ming Lin, and Dinesh Manocha are with Department of Computer Science University of North Carolina at Chapel Hill E-mail: {rungta, schissle, lin, dm}@cs.unc.edu.
- Ravish Mehra is with the Department of Computer Science University of North Carolina at Chapel Hill E-mail: ravish.mehra07@gmail.com.
- Chris Malloy E-mail: chris.p.malloy@gmail.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/

processes. For example, consider the case of a kitchen bowl falling from a countertop; the change in the directivity of the bowl with different hit positions and the effect of this time-varying, mode-dependent directivity on the propagated sound in the environment is mostly ignored by the current sound simulation techniques. Similarly, for a barrel rolling down the alley, the sound consists of multiple modes, where each mode has a time-varying radiation and propagation characteristic that depends on the hit positions on the barrel along with the instantaneous position and orientation of the barrel. Moreover, the interaction of the resulting sound waves with the walls of the alley cause resonances at certain frequencies and damping at others. Current sound simulation techniques model the barrel as a sound source with either static, mode-independent directivity, and model the resulting propagation in the environment with a mode-independent acoustic response or model the time-varying directivity of the barrel but propagate those in free-space only [7]. Due to these limitations, artists and game audio-designers have to manually design sound effects corresponding to these different scenarios, which can be very tedious and time-consuming [28].

Main Results: In this paper, we present the first coupled synthesis-propagation algorithm which models the entire process of sound simulation starting from the surface vibration of rigid objects, radiation of sound waves from these surface vibrations, and interaction of the resulting sound waves with the virtual environment for interactive applications. The key insights of our work is the use of a single-point multipole expansion (SPME) to couple the radiation and propagation characteristics of a source for each vibration mode. Mathematically, a single-point multipole corresponds to a single radiating source placed inside the object; this expansion significantly reduces the computational cost of the propagation stage compared to a multi-point multipole expansion. Moreover, we present a novel interactive mode-adaptive sound propagation technique that uses ray tracing to compute the per-mode impulse responses for a source-listener position. We also describe a novel perceptually-driven Hankel function approximation scheme that reduces the computational cost of this mode-adaptive propagation to enable interactive performance for virtual environments. The main benefits of our approach include:

1. Per-mode coupling of synthesis and propagation through the use of single-point multipole expansion.
2. Interactive mode-adaptive propagation technique based on perceptually-driven Hankel function approximation.
3. High degree of dynamism to model dynamic surface vibrations, sound radiation and propagation for moving sources and listeners.

Our technique performs end-to-end sound simulation from first principles and enables automatic sound effect generation for interactive applications, thereby reducing the manual effort and the time spent by artists and game-audio designers. Our system can automatically model the complex acoustic effects generated in various dynamic scenarios such as (a) swinging church bell inside a reverberant cathedral, (b) swaying wind chimes on the balcony of Tuscany countryside house, (c) a metal barrel falling downstairs in an indoor game scene and (d) orchestra playing music in a concert hall, at 10fps or faster on a multi-core desktop PC. We have integrated our technique with the UnityTM game engine and demonstrated complex sound effects enabled by our coupled synthesis-propagation technique in different scenarios (see Fig. 1).

Furthermore, we evaluated the effectiveness of our perceptual Hankel approximation algorithm by performing a preliminary user-study. The study was an online one where the subjects were shown snippets of three benchmarks (Cathedral, Tuscany, and Game) with audio delivered through headphones/earphones and rendered using our perceptual Hankel approximation and using no approximation. The subjects were asked to judge the similarity between the two sounds for the three benchmarks. Initial results show that the subjects were unable to distinguish between the two sounds indicating that our Hankel

approximation doesn't compromise on the audio quality in a perceptible way.

2 RELATED WORK AND BACKGROUND

In this section, we give an overview of sound synthesis, radiation, and propagation and survey some relevant work.

2.1 Sound Synthesis for rigid-bodies

Given a rigid body, sound synthesis techniques solve the modal displacement equation

$$\mathbf{K}\mathbf{d} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{f}, \quad (1)$$

where \mathbf{K} , \mathbf{C} , and \mathbf{M} are the stiffness, damping, and mass matrices, respectively and \mathbf{f} represents the (external) force vector. This gives a discrete set of mode shapes $\hat{\mathbf{d}}_i$, their modal frequencies ω_i , and the amplitudes $q_i(t)$. The vibration's displacement vector is given by:

$$\mathbf{d}(\mathbf{t}) = \mathbf{U}\mathbf{q}(\mathbf{t}) \equiv [\hat{\mathbf{d}}_1, \dots, \hat{\mathbf{d}}_M]\mathbf{q}(\mathbf{t}), \quad (2)$$

where M is total number of modes and $\mathbf{q}(\mathbf{t}) \in \mathfrak{R}^M$ is the vector of modal amplitude coefficients $q_i(t)$ expressed as a bank of sinusoids:

$$q_i(t) = a_i e^{-d_i t} \sin(2\pi f_i t + \theta_i), \quad (3)$$

where f_i is the modal frequency (in Hz.), d_i is the damping coefficient, a_i is amplitude, and θ_i is the initial phase.

[2] introduced modal analysis approach to synthesizing sounds. [46] introduced a measurement-driven method to determine the modes of vibration and their dependence on the point of impact for a given shape. Later, [22] were able to model arbitrarily shaped objects and simulate realistic sounds for a few of these objects at interactive rates. This approach is called the *modal analysis* and requires an expensive precomputation, but achieves interactive runtime performance. The number of modes generated tend to increase with the geometric complexity of the objects. [27] used a system of spring-mass along with perceptually motivated acceleration techniques to generate realistic sound effects for hundreds of objects in real time. [31] developed a contact model to capture multi-level surface characteristics based on [27]. Recent work on modal synthesis also uses the single point multipole expansion [51].

2.2 Sound Radiation and Propagation

Sound propagation in frequency domain is described using the Helmholtz equation

$$\nabla^2 p + \frac{\omega^2}{c^2} p = 0, \quad \mathbf{x} \in \Omega, \quad (4)$$

where $p = p(\mathbf{x}, \omega)$ is the complex-valued pressure field, ω is the angular frequency, c is the speed of sound in the medium, and ∇^2 is the Laplacian operator. To simplify the notation, we hide the dependence on angular frequency and represent the pressure field as $p(\mathbf{x})$. Boundary conditions are specified on the boundary of the domain $\partial\Omega$ by either the Dirichlet boundary condition that specifies the pressure on the boundary $p = f(\mathbf{x})$ on $\partial\Omega$, the Neumann boundary condition that specifies the velocity of the medium $\frac{\partial p(\mathbf{x})}{\partial n} = f(\mathbf{x})$ on $\partial\Omega$, or a mixed boundary condition that specifies $Z \in \mathbb{C}$, so that $Z \frac{\partial p(\mathbf{x})}{\partial n} = f(\mathbf{x})$ on $\partial\Omega$. The boundary condition at infinity is also specified using the *Sommerfeld radiation condition* [25]

$$\lim_{r \rightarrow \infty} \left[\frac{\partial p}{\partial r} + i \frac{\omega}{c} p \right] = 0, \quad (5)$$

where $r = \|\mathbf{x}\|$ is the distance of point \mathbf{x} from the origin.

Equivalent Sources: The uniqueness of the acoustic boundary value problem guarantees that the solution of the free-space Helmholtz equation along with the specified boundary conditions is unique inside Ω [23]. The unique solution $p(\mathbf{x})$ can be found by expressing the solution as a linear combination of *fundamental solutions*. One choice

of fundamental solutions is based on *equivalent sources*. An equivalent source $q(\mathbf{x}, \mathbf{y}_i)$ is the solution of the Helmholtz equation subject to the Sommerfeld radiation condition. Here \mathbf{x} is the point of evaluation, \mathbf{y}_i is the source position and $\mathbf{x}_i \neq \mathbf{y}_i$. The equivalent source can be expressed as:

$$q(\mathbf{x}, \mathbf{y}_i) = \sum_{l=0}^{L-1} \sum_{m=-l}^l c_{ilm} \varphi_{lm}(\mathbf{x}, \mathbf{y}_i) = \sum_{k=1}^{L^2} e_{ik} \varphi_k(\mathbf{x}, \mathbf{y}_i), \quad (6)$$

where k is a generalized index for (l, m) , φ_k are multipole functions, and c_{ilm} is the strength of multipoles. Multipoles are given as a product of two functions:

$$\varphi_{lm}(\mathbf{x}, \mathbf{y}_i) = \Gamma_{lm} h_l^{(2)}(kd_i) \psi_{lm}(\theta_i, \phi_i), \quad (7)$$

where (d_i, θ_i, ϕ_i) is the vector $(\mathbf{x} - \mathbf{y}_i)$ expressed in spherical coordinates, $h_l^{(2)}$ is the spherical *Hankel* function of the second kind, k is the wavenumber given by $\frac{\omega}{c}$, $\psi_{lm}(\theta_i, \phi_i)$ are the complex-valued spherical harmonics functions, and Γ_{lm} is the normalizing factor for the spherical harmonics.

2.2.1 Sound Radiation

The Helmholtz equation is the mathematical way to model sound radiation from vibrating rigid bodies. Boundary element method is a widely used method for solving acoustic radiation problems [9] but has a major drawback in terms of high memory requirements. An efficient technique known as the Equivalent source method (ESM) [23] exploits the uniqueness of the solutions to the acoustic boundary value problem. ESM expresses the solution field as a linear combination of equivalent sources of various orders (monopoles, dipoles, etc.) by placing these simple sources at variable locations inside the object and matching the boundary conditions on the object's surface, guaranteeing the correctness of solution. The pressure at any point in Ω due to N equivalent sources located at $\{\mathbf{y}_i\}_{i=1}^N$ can be expressed as a linear combination:

$$p(\mathbf{x}) = \sum_{i=1}^N \sum_{l=0}^{L-1} \sum_{m=-l}^l c_{ilm} \varphi_{lm}(\mathbf{x}, \mathbf{y}_i). \quad (8)$$

This compact representation of the pressure $p(\mathbf{x})$ makes it possible to evaluate the pressure at any point of the domain in an efficient manner. This is also known as the *multi-point multipole expansion*. Typically, this expansion uses a large number of low-order multipoles ($L = 1$ or 2) placed at different locations inside the object to represent the pressure field. [14] use this multi-point expansion to represent the radiated pressure field generated by a vibrating object. Another variant of this, is the *single-point multipole expansion* represented as

$$p(\mathbf{x}) = \sum_{l=0}^{L-1} \sum_{m=-l}^l c_{lm} \varphi_{lm}(\mathbf{x}, \mathbf{y}). \quad (9)$$

discussed in [23]. In this expansion, only a single multipole of high order is placed inside the object to match outgoing radiation field.

2.2.2 Geometric Sound Propagation

Geometric sound propagation techniques use the simplifying assumption that the wavelength of sound is much smaller than the features on the objects in the scene. As a result, these methods are most accurate for high frequencies and approximately model low-frequency effects like diffraction and scattering as separate phenomena. Commonly used techniques are based on image source methods and ray tracing. Recently, there has been a focus on computing realistic acoustics in real-time using algorithms designed for fast simulation. These include beam tracing [13] and ray-based algorithms [16, 40] to compute specular and diffuse reflections and can be extended to approximate edge diffraction. Diffuse reflections can also be modeled using acoustic rendering equation [37, 4]. In addition, frame-to-frame coherence of the sound field can be utilized to achieve a significant speedup [34].

2.3 Coupled Synthesis-Propagation

Ren et al. [29] presented an interactive virtual percussion instrument system that used modal synthesis as well as numerical sound propagation for modeling a small instrument cavity. However, the coupling proposed in this system did not incorporate a time-varying, mode-dependent radiation and propagation characteristic of the musical instruments. Additionally, this system only modeled propagation inside the acoustic space of the instrument and not the full 3D environment. Furthermore, the volume of the underlying acoustic spaces (instruments) in [29] was rather small in comparison to the typical scenes shown in this paper (see Fig. 1).

3 OVERVIEW

In this section, we provide an overview of our mode-adaptive, coupled synthesis-propagation technique (see Figure 2).

The overall technique can be split into two main stages: *preprocessing* and *runtime*. In the preprocessing stage, we start with the vibration analysis of each rigid object to compute its modes of vibrations. This step is performed using the finite element analysis of the object mesh to compute displacements (or shapes), frequencies, and amplitudes of all the modes of vibration. The next step is to compute the sound radiation field corresponding to each mode. This is done by using the mode shapes as the boundary condition for the free-space Helmholtz equation and solving it using the state-of-the-art boundary element method (BEM). This step computes the outgoing radiation field corresponding to each vibration mode. To enable interactive evaluation at runtime, the outgoing radiation fields are represented compactly using the *single-point multipole expansion* [23]. This representation significantly reduces the runtime computational cost for sound propagation by limiting the number of multipole sources to one per mode instead of hundreds or even thousands per mode in the case of multi-point multipole expansion [23, 14]. This completes our preprocessing step. The coefficients of the single-point multipole expansion are stored for runtime use.

At runtime, we use a mode-adaptive sound propagation technique that uses the single-point multipole expansion as the sound source for computing sound propagation corresponding to each vibration mode. In order to achieve interactive performance, we use a novel perceptually-driven Hankel function approximation. The sound propagation technique computes the impulse response corresponding to the instantaneous position for source-listener pair for each vibration mode. High modal frequencies are propagated using the geometric sound propagation techniques. Low modal frequencies can be propagated using the wave-based techniques. Hybrid techniques combine geometric and wave-based techniques to perform sound propagation in the entire frequency range. The final stage of the pipeline takes the impulse response for each mode, convolves it with that mode's amplitude, and sums it for all the modes to give the final audio at the listener.

We now describe each stage of the pipeline in detail.

Modal Analysis: We adopt a finite element method [22] to precompute the modes of vibration of an object. In this step, we first discretize the object into a tetrahedral mesh and solve the modal displacement equation (Eq. 1) analytically under the Raleigh-damping assumption (i.e. damping matrix \mathbf{C} can be written as a linear combination of stiffness \mathbf{K} and mass matrix \mathbf{M}). This facilitates the diagonalization of the modal displacement equation, which can then be represented as a generalized eigenvalue problem and solved analytically as system of decoupled oscillators. The output of this step is the vibration modes of the object along with the modal displacements, frequencies, and amplitudes. [30] showed that the Raleigh damping model is a suitable geometry-invariant sound model and is therefore a suitable choice for our damping model.

Sound Radiation: This step computes the sound radiation characteristic of the vibration modes of each object by solving the free-space Helmholtz equation [14]. The modal displacements of each mode serves as the boundary condition for the Helmholtz equation. The boundary element method (BEM) is then used to solve the Helmholtz

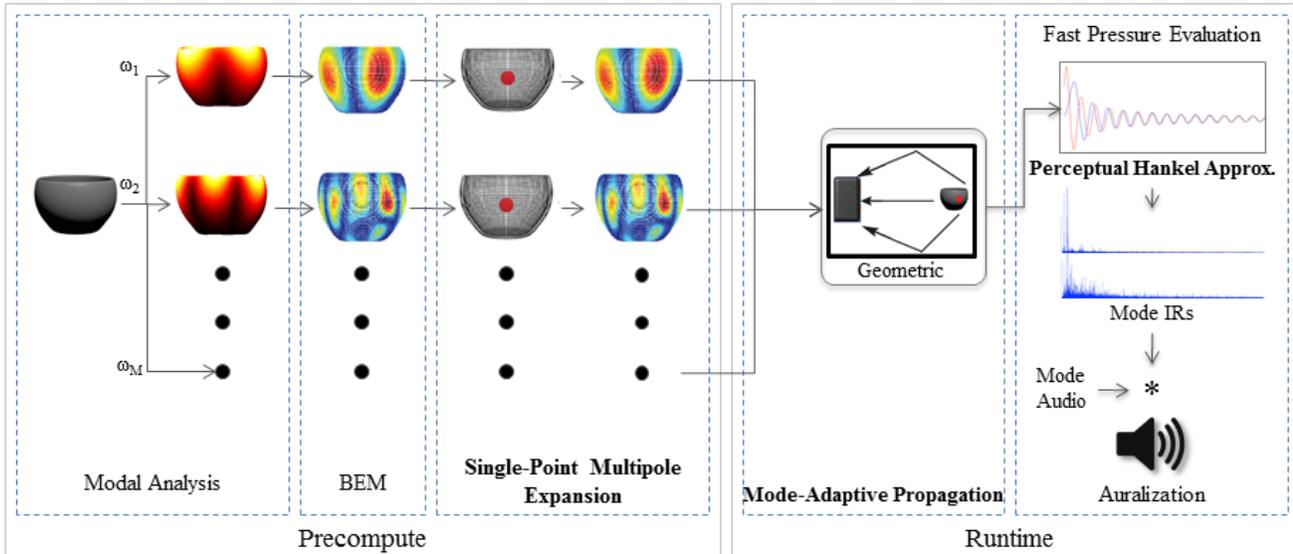


Fig. 2. Overview of our coupled synthesis and propagation pipeline for interactive virtual environments. The first stage of precomputation comprises the modal analysis. The figures in red show the first two sounding modes of the bowl. We then calculate the radiating pressure field for each of the modes using BEM, place a single multipole at the center of the object, and approximate the BEM evaluated pressure. In the runtime part of the pipeline, we use the multipole to couple with an interactive propagation system and generate the final sound at the listener. We present a new perceptual Hankel approximation algorithm to enable interactive performance. The stages labeled in bold are the main contributions of our approach.

equation and resulting outgoing radiation field is computed on an offset surface around the object. This outgoing pressure field can be efficiently represented by using either the single-point or multi-point multipole expansion.

Single-point Multipole fitting A key aspect of our approach is to represent the radiating sound fields for each vibrating mode in a compact basis by fitting the single-point multipole expansion, instead of a multi-point expansion. This representation makes it possible to use just one point source position for *all the vibration modes*. This formulation makes it possible to perform interactive modal sound propagation (Eq. 9).

Mode-Adaptive Sound Propagation: The main idea of this step is to perform sound propagation for each vibration mode of the object independently. The single-point multipole representation calculated in the previous step is used as the sound source in this step. By performing a mode-adaptive propagation, our technique models the mode-dependent radiation and propagation characteristic of sound simulation. The modal frequencies generated for the objects in our scenes tend to be high (i.e., more than 1000Hz). Ideally, we would like to use wave-based propagation algorithms [17, 18], as they are regarded more accurate. However, the complexity of wave-based methods increase as a fourth power of the frequency, and therefore they can very high time and storage complexity. We use a mode-adaptive sound propagation based on geometric methods.

Geometric Propagation: Given single-point multipole expansions of the radiation fields of a vibrating object, we use a geometric acoustic algorithm based on ray-tracing to propagate the field in the environment. In particular, we extend the interactive ray-tracing based sound propagation algorithm [34, 33] to perform mode-aware propagation. As discussed above, we use a single source for all the modes and trace rays from this source into the scene. Then, at each listener position, the acoustic response is computed for each mode by using the pressure field induced by the rays and scaled by the mode-dependent radiation filter corresponding to the single-point multipole expansion for that mode. In order to handle low-frequency effects, current geometric propagation algorithm use techniques based on uniform theory of diffraction. While they are not as accurate as wave-based methods, they can be used to generate plausible sound effects for virtual environments.

Auralization: The last stage of the pipeline involves computing the

final audio corresponding to all the modes. We compute this by convolving the impulse response of each mode with the mode’s amplitude and summing the result:

$$q(\mathbf{x}, t) = \sum_{i=1}^M q_i(t) * p^{o_i}(\mathbf{x}, t), \quad (10)$$

where $p^{o_i}(\mathbf{x}, t)$ is the acoustic response of the i th mode with angular frequency ω_i computed using sound propagation, $q_i(t)$ is the amplitude of the i th mode computed using modal analysis, \mathbf{x} is the listener position, M is the number of modes, and $*$ is the convolution operator.

4 COUPLED SYNTHESIS-PROPAGATION

In this section, we discuss in detail the single-point multipole expansion and the mode-adaptive sound propagation.

4.1 Single-Point Multipole Expansion

There are two types of multipole expansions that can be used to represent radiating sound fields: *single-point* and *multi-point*. In a single-point multipole expansion (SPME), a single multipole source of high order is placed inside the object to represent the sound field radiated by the object. On the other hand, multi-point multipole expansion places a large number of low order multipoles at different points inside the object to represent the sound field. Both SPME and MPME are two different representations of the outgoing pressure field and do not restrict the capabilities of our approach in terms of handling near-field and far-field computations.

To perform sound propagation using a multipole expansion, the number of sound sources that need to be created depend on the number of modes and the number of multipoles in each mode. In case of a single-point expansion, the number of sound sources is equal to M where M is the number of modes since the number of multipoles in each expansion is 1. In case of multi-point multipole expansion, the number of sound sources is equal to $\sum_{i=1}^M N_i$ where N_i is the number of multipoles in i th mode. The number of multipoles at each mode vary with the square of the mode frequency. This results in thousands of sound sources for multi-multipole expansion. The computational complexity of a sound propagation technique (wave-based or geometric) varies with the number of sound sources. As a result, we selected SPME in our approach. However, it is possible that there are some

cases where low-order MPME could be more efficient than a single and very high-order SPME. However, in the benchmarks used in the paper, SPME results in efficient runtime performance.

Previous sound propagation approaches have proposed the use of source clustering to reduce the computation required for scenes with many sources [43]. However, these techniques cannot be used to cluster multipoles as the clustering disrupts the phase of the multipoles, producing error in the sound radiation. Therefore, we chose to use a single-point multipole expansion to enable interactive sound propagation at runtime.

The output of this stage is the set of coefficients of the single-point multipole expansion (Eq. 9) for each mode (for example, coefficients c_{lm}^ω for mode ω).

4.2 Mode-adaptive Sound Propagation

We now propose a *position invariant* method of computing the sound propagation for each mode of the vibrating object. This approach brings down the number of sound sources to be propagated from M to just one. This is achieved by placing the SPME for all the modes at exactly the same position. Given a ray-tracing based geometric technique, this implies that instead of tracing rays for each mode separately, we trace rays from only a single source position. These rays are emitted from the source in different directions, get reflected/diffracted/scattered/absorbed in the scene, and reach the listener with different pressure values. Mode-dependent impulse response is computed for each mode by multiplying the pressure values produced by the traced rays with the corresponding SPME weights for each ray. We describe this approach in detail as follows:

Sound propagation is split into two computations: *mode-independent* and *mode-dependent* computations.

Mode-independent: We make use of the ray-based geometric technique of [34] to compute sound propagation paths in the scene. This system combines path tracing with a cache of diffuse sound paths to reduce the number of rays required for an interactive simulation. The approach begins by tracing a small number (e.g., 500) of rays uniformly in all directions from each sound source. These rays strike the surfaces and are reflected recursively up to a specified maximum reflection depth (e.g., 50). The reflected rays are computed using vector-based scattering [8], where the resulting rays are a linear combination of the specularly reflected rays and random Lambertian-distributed rays. The listener is modeled as a sphere the same size as a human head. At each ray-triangle intersection, the visibility of the listener sphere is sampled by tracing a few additional rays towards the listener. If some fraction of the rays are not occluded, a path to the listener is produced. A path contains the following output data: The total distance the ray traveled d , along with the attenuation factor α due to reflection and diffraction interactions. Diffracted sound is computed separately using the UTD diffraction model [42]. The frequency dependent effects are computed using a vector of frequency attenuation coefficients given the mode's frequency for both diffraction and reflection. This step remains the same for all the modes since the position of the source remains the same (across all the modes) as described above.

Mode-dependent: Given the output of the geometric propagation system, we can evaluate the mode-dependent acoustic response for a mode with angular frequency ω as:

$$p^\omega(\mathbf{x}, t) = \sum_{r \in R} |p_r^\omega(\mathbf{x})| w_r \delta(t - d_r/c), \quad (11)$$

where w_r is the contribution from a ray r in a set of rays R , d_r is the distance traveled by the ray r , c is the speed of sound, δ is the delta function, and $p_r^\omega(\mathbf{x})$ is the pressure contribution generated by the ray r for mode ω computed using the single-point multipole expansion:

$$p_r^\omega(\mathbf{x}) = \alpha_r \sum_{l=0}^{L-1} \sum_{m=-l}^{m=l} c_{lm}^\omega \varphi_{lm}^\omega(d_r, \theta_r, \phi_r), \quad (12)$$

where φ_{lm}^ω is the multipole, k is wavenumber of the mode ($k = \omega/c$), (θ_r, ϕ_r) is the direction of emission of ray r from the source, and α_r

is the attenuation factor. We switch between $h_l^{(2)}(kd_r)$ and its approximate variant $\tilde{h}_l^{(2)}(kd_r)$ based on the distance d_r in a mode-dependent manner as described next.

These mode-dependent acoustic responses are used in the auralization step as described in Section 3.

4.3 Hankel Approximation

The spherical Hankel function of the second kind, $h_l^{(2)}(kd)$, describes the radially-varying component of the radiation field of a multipole of order l . It is a complex-valued function of the distance d from the multipole position and the wave number $k = \omega/c$. This function itself is a linear combination of the spherical Bessel functions of the first and second kind, $j_l(kd)$ and $y_l(kd)$: $h_l^{(2)}(kd) = j_l(kd) - iy_l(kd)$. [1]. These Bessel functions are often evaluated to machine precision using a truncated infinite power series.

While this computation of the Bessel functions is accurate, it is also slow when the functions need to be evaluated many times. Within sound propagation algorithm, both Bessel functions need to be evaluated for each mode and each sound path through the scene. The number of paths in a reflective scene (e.g. cathedral) can easily exceed 10^5 , and the number of modes for the sounding objects is around 20 to 40, resulting in millions of Bessel function evaluations per frame. The Hankel function is also amenable to computation using recurrence relation(s). One such relation is given as:

$$h_{l+1}^{(2)}(kd) = \frac{2l+1}{kd} h_l^{(2)}(kd) - h_{l-1}^{(2)}(kd) \quad (13)$$

Unfortunately, computing the Hankel function using this recurrence relation has similar runtime costs as evaluating the Bessel functions, and can become a bottleneck for interactive applications. If the Hankel function is used directly, its evaluation for all modes and paths can take seconds.

Another possibility is to precompute a table for different values, and perform table lookup at runtime. However, such an approach is not practical, since Hankel is a 2D function (l, kd). For a table, the granularity of the arguments would have to be extremely fine, given the high numeric sensitivity of the function. Although, it would be easy to store the values of l and k as they're known beforehand, the value of d can have a large range, even for a small scene. This is because the value of d depends on the distance a ray travels as it reaches the listener position which could include multiple bounces in the environment.

Perceptual Hankel Approximation: We present an approximation technique for evaluation of the Hankel function for interactive applications. Our approach uses a perceptually-driven error threshold to switch between the full function evaluation and the approximation. We use the approximation function given by [17]:

$$h_l^{(2)}(kd) \approx \tilde{h}_l^{(2)}(kd) = i^{l+1} \frac{e^{-ikd}}{kd}. \quad (14)$$

This approximation converges to $h_l^{(2)}(kd)$ for large values of kd , but does not match well near the multipole. For this reason, we apply this approximation only in the *far field*, where the value of the distance d is greater than a threshold distance $d_{\tilde{}}$. Overall, the approximation works well even for small scenes since the reflected rays can take a long path before they reach the listener and be in the *far field*.

We determine this distance threshold independently for each mode frequency ω and its corresponding wave number k so that a perceptual error threshold is satisfied. We derive the error threshold for each mode from the absolute threshold of hearing at the mode's frequency. If the pressure error from the approximation is less than the threshold of hearing, the difference in pressure is unable to be perceived by a human listener [24]. The threshold of hearing can be well-approximated by the analytic function [41]:

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4. \text{ (dB SPL)}, \quad (15)$$

SPL stands for Sound Pressure Level and is measured in decibels (dB).

In a preprocessing step, we evaluate this function at each mode’s frequency to determine a per-mode error threshold, and then determine the distance threshold $d_{\tilde{h}_i}$ where the approximation is perceptually valid for the mode. This information is computed and stored for each sounding object. At runtime, when the pressure contribution for each path i is computed, we use the original Hankel $h_i^{(2)}(k_id_i)$ when $d_i < d_{\tilde{h}_i}$ and the approximation $\tilde{h}_i^{(2)}(k_id_i)$ when $d_i \geq d_{\tilde{h}_i}$.

We would like to note that although the approximation to Hankel function specified in Eq 14 is standard, the novelty of our approach lies in the way we use it. As described above, we use perceptually-driven thresholds to decide when to automatically switch to the approximate version. We also did a user-evaluation to make sure the perceptually-motivated approximation doesn’t cause any loss of quality in our context. The details of the evaluation are presented in the Section 5.

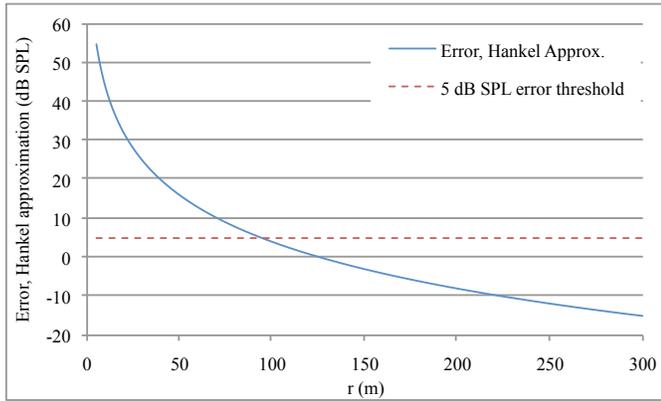


Fig. 3. The error between the Hankel function approximation $\tilde{h}_i^{(2)}(kd)$ and the original function $h_i^{(2)}(kd)$ decreases at increasing values of d for order $l = 6$ and mode frequency 1000Hz. An error threshold of $\epsilon = 5$ dB SPL is overlaid. For this case, the approximation threshold distance is chosen to be $d_{\tilde{h}_i} = 93\text{m}$. All sound paths for this mode frequency with $d > 93\text{m}$ use this approximation.

Error Threshold Preprocessing: Given a perceptual error threshold such as $\epsilon = 5$ dB SPL, we use a brute-force approach to determine the smallest value of $d_{\tilde{h}_i}$ for which the error of the approximation is less than ϵ for all distances $d > d_{\tilde{h}_i}$. We have included Figure 3 that shows an example of how the error shrinks at increasing values of d . Our approach starts at the multipole position and samples the error value at $\lambda/10$ to avoid aliasing. The method stops when d reaches a point past the end of the longest expected impulse response (e.g., 1000m). The final value for $d_{\tilde{h}_i}$ is chosen to be the last d sample where the error dropped below ϵ .

The result of applying this approximation is that our sound propagation system is able to handle pressure computation for interactive scenes that are much more complex and with many more sound paths than with the original Hankel formulation. In addition, the error due to our approach is small and not perceptible by a human listener.

Near-field vs. far-field: As mentioned in Sec 4.1, Equivalent Source theory states that if the pressure on the offset surface is matched by matching the appropriate boundary condition, the pressure field is valid in the near-field as well as far-field. We use the perceptual Hankel approximation for far-field computation, but we don’t truncate the order of the multipole anywhere. In particular, we use the exact multipole formulation everywhere with the following difference: the Hankel function part of multipole is approximated in the far-field but the expansion is never truncated anywhere in the domain. So the only difference in the computation of near and far-fields is in terms of Hankel computation.

5 USER-EVALUATION OF HANKEL APPROXIMATION

In order to evaluate the accuracy of our chosen thresholds, we performed an online user-study with 3 benchmark: the Cathedral, Tuscany, and the Game benchmark. Given the scope of our experiments, an online study was the best choice as it offered the subjects convenience of taking the study as per their convenience and at a pace they were comfortable with. This also eased the process of keeping their identities confidential. We generated the audio for these scenes using the perceptual Hankel approximation and the full Hankel computation. The Tuscany benchmark has the Unity in-game, static soundscape playing and was left that way to make scene appear more natural and have a better audio-visual correlation. For the study, we consider the full Hankel computation to be the *base* method while the approximated-Hankel was considered as *our* method.

Participants The study was taken by 29 subjects all within the age group of 18 and 50 with 18 males and 11 females. The mean age of all the participants was 27.3 and all of them reported normal hearing. The subjects were recruited by sending out emails to the departments, colleagues, and friends. The subjects were not paid for their participation.

Procedure The participants were given instructions on the study and asked to fill out a questionnaire on their background. The subjects were required to have a headphone/earphone before they could take part in the study. There was one test scene to help them calibrate their headphones/earphones and make sure they’re oriented correctly (right channel on right ear, left channel on left). We designed four cases: *base vs. base*, *our vs. base*, *base vs. our*, and *our vs. our* for each of the three scenes. In total, twelve video pairs were generated for the benchmarks (4 cases x 3 benchmarks). We performed an online survey where subjects were presented the four cases in a random order and asked to answer a single question, “Compared to the audio in the left video, how similar is the audio in the right video?”. The choice of the question was motivated by [26, 3] where the authors use a similar question and a similar scale to measure similarity between two stimuli. Our hypothesis was: Sound produced by *our* method would be indistinguishable from the *base* method. If our hypothesis is validated, it would *indicate* that our Hankel approximation is perceptually equivalent to full Hankel computation. The subjects were then presented the 12 benchmarks in a random order and asked to rate the similarity on a scale on 1 to 11 with 1 being the audio in the two videos is very different and 11 being the audio in the two videos is virtually the same. There was no repetition of stimuli to make sure there was no learning between subsequent iterations given the low number of stimuli present. The study had no time constraints and the participants were free to take breaks in-between the benchmarks as long as the web-session did not expire. After presenting the 12 benchmarks, the subjects were given the opportunity to leave open (optional) comments. Although, it is difficult to ascertain the average time it took the

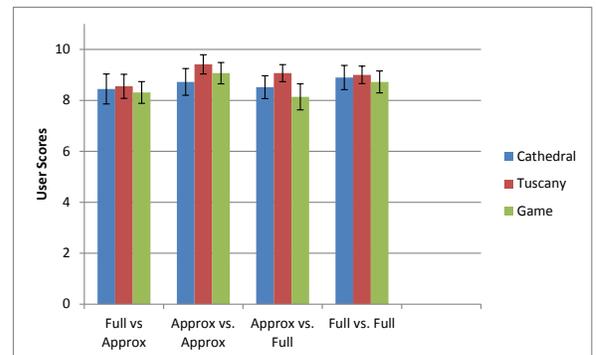


Fig. 4. Mean and standard errors of the subjects’ scores on the user-study. Full refers to sound computed using Full Hankel, while Approx refers to sound computed using our perceptual approximation. The response is to the question, “Compared to the audio in the left video, how similar is the audio in the right video?”

Scene	Full vs. Approx		Approx vs. Approx		Approx vs. Full		Full vs. Full	
	Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper
Cathedral	-0.4021	1.2054	-0.3587	1.0754	-0.3064	0.9184	-0.3259	0.9769
Tuscany	-0.3246	0.9729	-0.2572	0.7710	-0.2298	0.6890	-0.2350	0.7043
Game	-0.2919	0.8751	-0.2856	0.8562	-0.3504	1.0502	-0.2935	0.8798

Table 1. Equivalence test results for the three scenes. The equivalence interval was ± 2.2 while the confidence level was 95%

subjects to finish the study, in our experience, the study took around 15-20 minutes on average.

Results and Discussion The questions posed to participants of the study include mixed cases between audio generated using the full Hankel and approximate Hankel functions as well as cases where either the full or approximate Hankel function was used to generate both audio samples in a pair. Our hypothesis is thus that the subjects are going to rate the full vs. approximate similar to what they rate full vs. full, which would indicate that users are unable to perceive a difference between results generated using the full functions and those generated using their approximation. The mean values and the standard errors are shown in the Fig 4. The figure shows how close the mean scores are for the full vs. approximate test as compared to the full vs. full test.

The responses were analyzed using the non-parametric Wilcoxon signed-rank test on the full vs. full and approximate vs. approximate data to ascertain whether their population mean ranks differ. The Wilcoxon signed-rank test failed to show significance for all the three benchmarks: Cathedral ($Z = -0.035$, $p = 0.972$), Tuscany ($Z = -1.142$, $p = 0.254$), and Game ($Z = 0.690$, $p = 0.49$) indicating that the population means do not differ for all the three benchmarks. The responses were also analyzed using the non-parametric Friedman test. The Friedman test, too, failed to show significance for the benchmarks: Cathedral ($\chi^2(1) = 0.048$, $p = 0.827$), Tuscany ($\chi^2(1) = 2.33$, $p = 0.127$), Game ($\chi^2(1) = 0.053$, $p = 0.819$).

The responses were further analyzed using confidence interval approach to show equivalence between the groups. The equivalence interval was chosen to be the $\pm 20\%$ of our 11-point rating scale, i.e., ± 2.2 . The confidence level was chosen to be 95%. Table 1 shows that the lower and upper values of the confidence intervals lie within our equivalence intervals indicating that the groups are equivalent.

6 IMPLEMENTATION AND RESULTS

In this section, we describe the implementation details of our system. All the runtime code was written in C++ and timed on a 16-core workstation with Intel Xeon E5 CPUs with 64 GB of RAM running Windows 7 64-bit. In the preprocessing stage, the eigen decomposition code was written in C++, while the single-point multipole expansion was written in MATLAB.

Preprocessing: We used finite element technique to compute the stiffness matrix \mathbf{K} which takes the tetrahedralized model, Young’s modulus, and the Poisson’s ratio of the sounding object and compute the stiffness matrix for the object. Next, we compute the eigenvalue decomposition of the system using Intel’s MKL library (DSYEV) and calculate the modal displacements, frequencies, and amplitudes in C++. The code to find the multipole strengths was written in MAT-

Scene	#Tri.	#Paths	#S	#M	Time		
					Prop.	Pres.	Tot
Sibenik	77083	30850	1	15	52.2	57.9	110.1
Game	100619	58363	1	5	69.5	22.7	92.2
Tuscany	98274	9232	3	14	62.2	16.8	79
Auditor.	12373	13742	3	17	82.5	12.5	95

Table 2. We show the performance of our runtime system (mode-adaptive propagation). The number of modes for Tuscany and Auditorium is the sum over all sources used. The number of modes and number of paths were chosen to give a trade-off for speed vs. quality. All timings are in milliseconds. We show the breakdown between ray-tracing based propagation (Prop.) and pressure (Pres.) computation and the total (Tot) time per frame on a multi-core PC. #S is the number of sources and #M is the number of modes.

LAB, the Helmholtz equation was solved using the FMM-BEM (Fast-multipole BEM) method implemented in FastBEM software package. Our current implementation is not optimized. It takes about 1-15 hours on our current benchmarks.

Sound Propagation: We use a fast, state-of-the-art geometric ray tracer [34] to get the paths for our pressure computation. This technique is capable of handling very high orders of diffuse and specular reflections (e.g., 10 orders of specular reflections and 50 orders of diffuse reflections) and still maintain interactive performance. The ray tracing system scales linearly with the number of cores keeping the propagation time low enough for the entire frame to be interactive (see Table 2).

Spherical Harmonic computation: The number of spherical harmonics computed per ray varies as $O(L^2)$, making naive evaluation too slow for an interactive runtime. We used a modified version of available fast spherical harmonic code [38] to compute the pressure contribution of each ray. The available code computes only the real spherical harmonics by making extensive use of SSE (Streaming SIMD Extension). We find the complex spherical harmonics from the real ones following a simple observation:

$$Y_l^m = \frac{1}{\sqrt{2}}(Y_l^m + iY_l^{-m}) \quad m > 0, \quad (16)$$

$$Y_l^m = \frac{1}{\sqrt{2}}(Y_l^m - iY_l^{-m})(-1)^m \quad m < 0. \quad (17)$$

Since our implementation uses the recurrence relation to compute the associated Legendre polynomials along with extensive SIMD usage, it makes it faster than the GSL implementation and significantly faster other implementation such as BOOST.

Approximate Hankel Function: As mentioned in Section 4, the Hankel function is approximated when the listener is sufficiently far away from the listener. The approximate Hankel function $\tilde{h}_l^{(2)}(kd) = i^{l+1} \frac{e^{-ikd}}{kd}$ reduces to computing $\sin(kd)$ and $\cos(kd)$. In order to accelerate this computation further, we use a lookup table for computing sines and cosines, improving the approximate Hankel computation by a factor of about four, while introducing minimal error as seen in Section 7.3. The lookup table for the sines and cosines make no noticeable perceptual difference in the quality of sound.

Parallel computation of mode pressure: In order to make the system scalable, we parallelize over the number of paths in the scene rather than the number of modes. Parallelizing over the number of modes would not be beneficial if $number\ of\ cores > number\ of\ modes$. Since the pressure computation for each ray is done independent of the other, the system parallelizes easily over the paths in the scene. We use OpenMP for the parallelization on a multi-core machine. Further, the system is configured to make extensive use of SIMD allowing it to process 4 rays at once. Refer to Table 2 for a breakdown of time spent on pressure computation and propagation for the different scenes.

Real-Time Auralization: The final audio for the simulations is rendered using a streaming convolution technique [11]. Once the audio is rendered, it can be played on the usual output devices such as headphones or multi-channel stereo. Although, headphones would give the best results in terms of localization. All audio rendering is performed at a sampling rate of 44.1 kHz.

6.1 Results

We now describe the different scenarios we used to test our system.

Cathedral: This scene serves as a way to test the effectiveness of our method in a complex indoor environment. We show a modal object (Bell) that has impulses applied to it. As the listener moves about in the scene the intensity of sound varies depending on the distance of the listener from the bell. Further, since the cathedral corresponds to an indoor environment, effects such as reflections and late reverberation coupled with modal sounds become apparent.

Tuscany: The Tuscany scene provides a means to test the indoor/outdoor capabilities of our system. The modal object (Three bamboo chimes) is placed on the balcony with the wind providing the

Object	#Tris	Dim. (m)	#Modes	Freq. Range (Hz)	Order
Bell	14600	0.32	20	480 - 2148	13-36
Barrel (Auditorium)	7410	0.6	20	397 - 2147	13-37
Barrel (Game)	7410	1.03	9	370 - 2334	8-40
Chime - Long	3220	0.5	4	780 - 2314	7-19
Chime - Medium	3220	0.4	6	1135 - 3958	10-24
Chime - Short	3220	0.33	4	1564 - 3495	10-15
Bowl	20992	0.35	20	870 - 5945	8-36
Drum	7600	0.72	13	477 - 1959	8-28
Drum stick	4284	0.23	7	1249 - 3402	7-15
Trash can	7936	0.60	5	480 - 1995	11-17

Table 3. We show the characteristics of SPME for different geometries and materials.

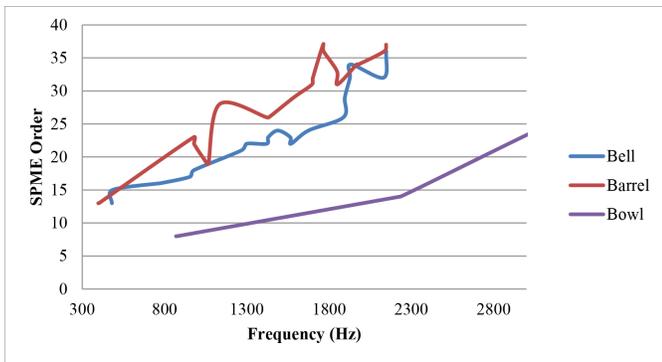


Fig. 5. The order required by the Single-Point multipole generally increases with increasing modal frequency. We show the results for the objects used in our simulations. It is possible for the same modal frequency (for different objects) to have different order multipole owing to difference in geometries of these objects. The plot shows the SPME order required for approximating the radiation pattern of different objects as a function of their increasing modal frequencies.

impulses. As the listener goes around the house and moves inside, the propagated sound of the chimes changes depending on the position of the listener in the environment. The sound is much less in intensity outside owing to most of the propagated sound being lost in the environment and increases dramatically when the listener goes in.

Game Scene: This demo showcases the effectiveness of our system in a game like environment containing, both, an indoor and a semi-outdoor environment. We use a metal barrel as our sounding object and let the listener interact with it. Initially, the barrel rolls down a flight of stairs in the indoor part of the scene. The collisions with the stairs serve as input impulses and generate sound in an enclosed environment, with effects similar to that in the Cathedral scene. The listener then picks up the barrel and rolls it out of the door and follows it. As soon as the barrel exits the door, the environment outside is a semi-outdoor one, the reverberation characteristics change, demonstrating the ability of our system to handle modal sounds with different environments in a complex game scene.

Auditorium: This scene showcases the ability of our system to support multiple sound sources and propagate them inside an environment. We use a metal barrel, bell (from the Cathedral), a toy wooden drum, a drum stick, and a trash can lid to form a garage band. The *instruments* play a joyful percussive piece and provide the listener with the sound from a particular seat in the auditorium. (Fig. 6)

6.2 Analysis

Fig 5 shows the different orders of Single-Point Multipoles needed for the different objects as function of their modal frequencies. We choose an error threshold based on [17] as our error threshold ϵ when computing the co-efficients of SPME for a particular mode. The order of the SPME is iterated till the error drops below ϵ . We used $\epsilon = 0.15$ for each mode. (Fig. 7)

We have included a table (Table 4) that shows the performance improvement we get in various scenes with our Perceptual-Hankel approximation. The results were computed on a single thread. The first three scenes had the listener moving around in the scene and being at different distances from the sounding object. This indicates that the listener moves in and out of the near-field of the object (Refer to the supplemental video). And as the table indicates, approximation is still at least 3x faster than full Hankel computation without loss in quality.

Scenario	#Paths	F-Hankel(ms)	P-Hankel(ms)	Speed-up
Sibenik	42336	7837.72	1794.5	4.37
Game	55488	5391.6	754.5	7.14
Tuscany	6575	225.73	69.75	3.23
Auditorium	11889	1395	284.75	4.9

Table 4. The speed-up obtained using the Perceptual-Hankel approximation. We achieve at least 3 – 7x speed-up with no loss in the perceptual quality of sound. Here, F-Hankel stands for Full-Hankel while P-Hankel stands for Perceptual-Hankel. The results for Tuscany and Auditorium are averaged over all the sources.

Table 2 shows that we can achieve interactive performance (10 fps) using our system. The number of modes and number of rays in the scene can be controlled in order to get the best performance vs. quality balance. Table 5 shows the case for the Cathedral scene. The bell has 20 computed modes with about 44k rays on the one end and 13k rays with 1 mode on the other. The framework can be customized to suit the needs of a particular scenario to offer the best quality/cost ratio. Further, owing to the scalable nature of our system, more number of cores scales the performance almost linearly.

#Paths	Prop. Time	1 mode	5 modes	10 modes	15 modes	20 modes
44148	84.23	3.23	15.46	31.9	60.8	152.5
30850	52.27	2.2	11.2	29.9	57.9	144.1
22037	37.8	2	10.5	31.3	61	127.9
13224	25	1.6	9.4	27.8	53.7	102.7

Table 5. The table shows how controlling the number of rays and the number of modes can influence the timing in the Cathedral scene with a bell. This can help one customize the system to provide the best quality/performance ratio for a particular scenario. The total time taken is propagation time + time for chosen number of modes. All times are reported in milliseconds.

7 LIMITATIONS, CONCLUSION AND FUTURE WORK

We present the first coupled sound synthesis-propagation algorithm that can generate realistic sound effects for computer games and virtual reality, by combining modal sound synthesis, sound radiation, and sound propagation. The radiating sound fields are represented in a compact basis using a single-point multiple expansion. We perform sound propagation using this source basis via a fast ray-tracing technique to compute the impulse responses using perceptual Hankel approximation. The resulting system has been integrated and we highlight the performance in many indoor and outdoor scenes. Our user-



Fig. 6. The Auditorium Music Scene. This scene includes multiple sources playing a musical composition.

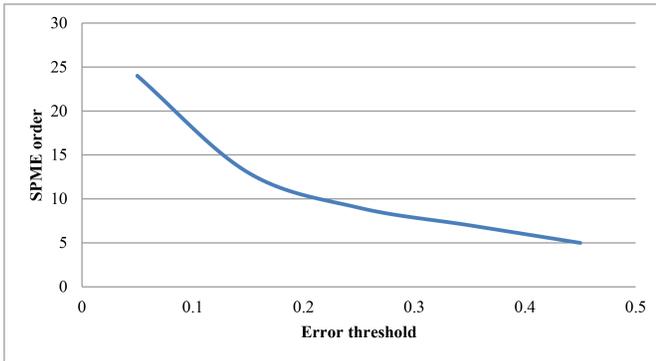


Fig. 7. For an increasing error threshold ϵ , the order of the multipole decreases almost quadratically. This demonstrates our SPME algorithm provides a very good approximation.

study demonstrates that perceptual Hankel approximations doesn't degrade sound quality and results in interactive performance. To the best of our knowledge, ours is the first system that successfully combines these methods and can handle a high degree of dynamism in term of source radiation and propagation in complex scenes.

Our approach has some limitations. Our current implementation is limited to rigid objects and modal sounds. Moreover, the time complexity tends to increase with the mode frequency. Our single-point multipole expansion approach can result in high orders of multipoles. The geometric sound propagation algorithm may not be able to compute the low frequency effects (e.g. diffraction) accurately in all scenes. Moreover, the wave-based sound propagation algorithm involves high pre-computation overhead and is limited to static scenes.

There are several avenues for future work. In addition to overcoming these limitations, we can further integrate other acceleration techniques, such as mode compression, mode culling etc [27] for use in more complex indoor and outdoor environments and generate other sound effects in large virtual environments (e.g. outdoor valley). It would also be useful to consider the radiation efficiency of each mode and use more advanced compression techniques [28]. It would be useful to accelerate the computations using iterative algorithms like Arnoldi's [5]. Integrating non-rigid synthesized sounds, e.g., liquid sounds [20] into our framework would be an interesting direction of future research. Our system is fully compatible with binaural rendering techniques such as HRTF-based (Head Related Transfer Function) rendering and it is our strong belief that using such techniques would improve the degree of presence that our system currently provides. [6, 15]. To this end, we would like to incorporate fast HRTF extraction methods such as [19] and evaluate the benefits. Our current user-evaluation can be expanded in multiple ways that might reveal interesting perceptual metrics which might further help optimize the system. Finally, we would like to use these approaches in VR applications and evaluate their benefits.

8 ACKNOWLEDGMENT

The authors would like to thank Alok Meshram, Nic Morales, and Priyadarshi Sharma 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 NSF grants 1320644 and 1456299 (under subcontract to Impulsive Inc.) and Link Foundation Fellowship in Advanced Simulation and Training.

REFERENCES

- [1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Number 55. Courier Dover Publications, 1972.
- [2] J.-M. Adrien. The missing link: Modal synthesis. In *Representations of musical signals*, pages 269–298. MIT Press, 1991.
- [3] K. M. Aldrich, E. J. Hellier, and J. Edworthy. What determines auditory similarity? the effect of stimulus group and methodology. *The Quarterly Journal of Experimental Psychology*, 62(1):63–83, 2009.
- [4] L. Antani, A. Chandak, L. Savioja, and D. Manocha. Interactive sound propagation using compact acoustic transfer operators. *ACM Trans. Graph.*, 31(1):7:1–7:12, Feb. 2012.
- [5] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951.
- [6] D. Begault. 3-d sound for virtual reality and multimedia, academic press. Boston, MA, 1994.
- [7] J. N. Chadwick, S. S. An, and D. L. James. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. In *ACM Transactions on Graphics (TOG)*, volume 28, page 119. ACM, 2009.
- [8] C. Christensen and G. Koutsouris. Odeon manual, chapter 6. 2013.
- [9] R. D. Ciskowski and C. A. Brebbia. *Boundary element methods in acoustics*. Computational Mechanics Publications Southampton, Boston, 1991.
- [10] Durlach. Virtual reality scientific and technological challenges. Technical report, National Research Council, 1995.
- [11] G. P. Egelmeers and P. C. Sommen. A new method for efficient convolution in frequency domain by nonuniform partitioning for adaptive filtering. *IEEE Transactions on signal processing*, 44(12):3123–3129, 1996.
- [12] K. Franinovic and S. Serafin. *Sonic Interaction Design*. MIT Press, 2013.
- [13] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proc. of ACM SIGGRAPH*, pages 21–32, 1998.
- [14] D. L. James, J. Barbič, and D. K. Pai. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics (TOG)*, 25(3):987–995, 2006.
- [15] P. Larsson, D. Vastfjall, and M. Kleiner. Better presence and performance in virtual environments by improved binaural sound rendering. In *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002.
- [16] 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.
- [17] R. Mehra, L. Antani, S. Kim, and D. Manocha. Source and listener directivity for interactive wave-based sound propagation. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):567–575, 2014.
- [18] R. Mehra, A. Rungta, A. Golas, M. Lin, and D. Manocha. Wave: Interactive wave-based sound propagation for virtual environments. *Visualization and Computer Graphics, IEEE Transactions on*, 21(4):434–442, 2015.
- [19] A. Meshram, R. Mehra, H. Yang, E. Dunn, J.-M. Franm, and D. Manocha. P-hrtf: Efficient personalized hrtf computation for high-fidelity spatial sound. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 53–61. IEEE, 2014.
- [20] W. Moss, H. Yeh, J.-M. Hong, M. C. Lin, and D. Manocha. Sounding liquids: Automatic sound synthesis from fluid simulation. *ACM Transactions on Graphics (TOG)*, 29(3):21, 2010.
- [21] R. Nordahl, S. Serafin, and L. Turchet. Sound synthesis and evaluation of interactive footsteps for virtual reality applications. *Proc. of IEEE VR*, pages 147–153, 2010.
- [22] J. F. O'Brien, C. Shen, and C. M. Gatchalian. Synthesizing sounds from rigid-body simulations. In *The ACM SIGGRAPH 2002 Symposium on Computer Animation*, pages 175–181. ACM Press, July 2002.
- [23] M. Ochmann. The full-field equations for acoustic radiation and scattering. *The Journal of the Acoustical Society of America*, 105(5):2574–2584, 1999.
- [24] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000.
- [25] A. D. Pierce et al. *Acoustics: an introduction to its physical principles and applications*. McGraw-Hill New York, 1981.
- [26] T. A. Polk, C. Behensky, R. Gonzalez, and E. E. Smith. Rating the similarity of simple perceptual stimuli: asymmetries induced by manipulating exposure frequency. *Cognition*, 82(3):B75–B88, 2002.
- [27] N. Raghuvanshi and M. C. Lin. Interactive sound synthesis for large scale environments. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 101–108. ACM, 2006.
- [28] N. Raghuvanshi and J. Snyder. Parametric wave field coding for pre-

- computed sound propagation. *ACM Transactions on Graphics (TOG)*, 33(4):38, 2014.
- [29] Z. Ren, R. Mehra, J. Copusky, and M. C. Lin. Tabletop ensemble: touch-enabled virtual percussion instruments. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 7–14. ACM, 2012.
- [30] Z. Ren, H. Yeh, R. Klatzky, and M. C. Lin. Auditory perception of geometry-invariant material properties. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):557–566, 2013.
- [31] Z. Ren, H. Yeh, and M. C. Lin. Synthesizing contact sounds between textured models. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 139–146. IEEE, 2010.
- [32] D. Rocchesso, S. Serafin, F. Behrendt, N. Bernardini, R. Bresin, G. Eckel, K. Franinovic, T. Hermann, S. Pauletto, P. Susini, and Y. Visell. Sonic interaction design: Sound, information and experience. *Proc. of ACM SIGCHI*, pages 3969–3972, 2008.
- [33] C. Schissler and D. Manocha. Interactive sound propagation and rendering for large multi-source scenes. Technical report, Department of Computer Science, University of North Carolina at Chapel Hill, 2015.
- [34] C. Schissler, R. Mehra, and D. Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Trans. Graph.*, 33(4):39:1–39:12, July 2014.
- [35] S. Serafin. *The Sound OF Friction: Real-Time Models, Playability and Musical Applications*. PhD thesis, Stanford University, 2004.
- [36] R. D. Shilling and B. Shinn-Cunningham. Virtual auditory displays. *Handbook of virtual environment technology*, pages 65–92, 2002.
- [37] S. Siltanen, T. Lokki, S. Kiminki, and L. Savioja. The room acoustic rendering equation. *The Journal of the Acoustical Society of America*, 122(3):1624–1635, September 2007.
- [38] P.-P. Sloan. Efficient spherical harmonic evaluation. *Journal of Computer Graphics Techniques (JCGT)*, 2(2):84–83, September 2013.
- [39] F. Steinicke, Y. Visell, J. Campos, and A. Lcuyer. *Human Walking in Virtual Environments: Perception, Technology, and Applications*. 2015.
- [40] M. Taylor, A. Chandak, L. Antani, and D. Manocha. Resound: interactive sound rendering for dynamic virtual environments. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 271–280, New York, NY, USA, 2009. ACM.
- [41] E. Terhardt. Calculating virtual pitch. *Hearing research*, 1(2):155–182, 1979.
- [42] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proc. of ACM SIGGRAPH*, pages 545–552, 2001.
- [43] N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. *ACM Trans. Graph.*, 23(3):249–258, 2004.
- [44] L. Turchet. Designing presence for real locomotion in immersive virtual environments: an affordance-based experiential approach. *Virtual Reality*, 19(3-4):277–290, 2015.
- [45] L. Turchet, S. Spagnol, M. Geronazzo, and F. Avanzini. Localization of self-generated synthetic footstep sounds on different walked-upon materials through headphones. *Virtual Reality*, pages 1–16, 2015.
- [46] K. van den Doel, P. G. Kry, and D. K. Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *Proc. of ACM SIGGRAPH*, pages 537–544, 2001.
- [47] Y. Visell, F. Fontana, B. L. Giordano, R. Nordahl, S. Serafin, and R. Bresin. Sound design and perception in walking interactions. 67(11):947–959, 2009.
- [48] M. Vorländer. Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm. *The Journal of the Acoustical Society of America*, 86(1):172–178, 1989.
- [49] D. Young and S. Serafin. Playability evaluation of a virtual bowed string instrument. *Proc. of Conference on New Interfaces for Musical Expression*, pages 104 – 108, 2003.
- [50] C. Zheng and D. L. James. Harmonic fluids. *ACM Trans. Graph.*, 28(3):1–12, 2009.
- [51] C. Zheng and D. L. James. Toward high-quality modal contact sound. *ACM Transactions on Graphics (TOG)*, 30(4):38, 2011.