

# Direct-to-Indirect Acoustic Radiance Transfer

Lakulish Antani, Anish Chandak, Micah Taylor and Dinesh Manocha

**Abstract**—We present an efficient algorithm for simulating diffuse reflections of sound in a static scene. Our approach is built on the latest advances in precomputed light transport techniques for visual rendering and uses them to develop an improved acoustic radiance transfer technique. We precompute a direct-to-indirect acoustic transfer operator for the scene, and use it to map direct sound incident on the surfaces of the scene to multi-bounce diffuse indirect sound, which is then gathered at the listener to compute the final impulse response. Our algorithm decouples the transfer operator from the source position so we can efficiently update the acoustic response at the listener when the source moves. We highlight its performance on various benchmarks and observe significant speedups over prior methods based on acoustic radiance transfer.

**Index Terms**—sound propagation, radiosity, virtual reality, precomputed transport



## 1 INTRODUCTION

Sound rendering or auditory displays can augment graphical rendering and provide the user with an enhanced spatial sense of presence. Some of the driving applications of sound rendering include acoustic design of architectural models or outdoor scenes, walkthroughs of large CAD models with sounds of machine parts or moving people, urban scenes with traffic, computer games, etc. The computation of sound propagation paths takes into account the knowledge of sound sources, listener locations, the 3D model of the environment, and material absorption and scattering properties.

The modeling of sound propagation effects needs to account for different wave propagation phenomena such as specular reflections, scattering (including diffuse reflections and edge diffraction), interference, etc. In this paper, we mainly focus on the modeling of diffuse reflections, which are considered important for modeling sound propagation [1]. Many objective [2] and perceptual [3] studies have been conducted to ascertain the importance of diffuse reflections for sound propagation. Further, it is computationally challenging to model high orders of diffuse reflection. Thus, due to its importance and computational challenge, modeling diffuse reflections for sound propagation is an active area of interest in many sound rendering applications.

At a broad level, sound propagation algorithms can be classified into numerical and geometric methods. Numerical methods attempt to directly compute numerical solutions to the acoustic wave equation. However, the complexity of these methods is proportional to the volume of the scene and the fourth power of the maximum frequency of sound simulated, and can be very slow for large acoustic spaces or high frequency sound sources. In terms of geometric sound propagation, two standard methods used to simulate diffuse sound reflections are based

on ray (or volume) tracing and radiance transfer. Our approach is motivated by recent developments in global illumination based on precomputed light transport algorithms [4], [5], [6]. Specifically, our work is inspired by direct-to-indirect transfer algorithms for visual rendering, that map direct light incident on the surfaces of a scene to indirect light on the surfaces of the scene after multiple bounces.

**Main Results** We present a new algorithm for modeling diffuse reflections of sound based on the direct-to-indirect transfer approach. This formulation decouples the precomputation of acoustic radiance transfer operators from both the source and the listener positions, and can efficiently update the acoustic response at the listener whenever the source moves.

The algorithm uses an SVD approximation of the transfer operator to perform higher-order diffuse reflections. We show that this allows us to reduce the memory requirements and increase the performance of our algorithm. Our algorithm decouples the transfer operator from the source position, and thus accelerates the computations as compared to the state-of-the-art.

We highlight the performance of our algorithm on various models. In practice, it is much faster than prior methods based on radiance transfer. To the best of our knowledge, it is the first approach that can perform diffuse reflections in static scenes with moving sources at almost interactive rates.

The rest of this paper is organized as follows. Section 2 covers some background material and gives a brief survey of related work. Section 3 gives a broad overview of our approach. Section 4 discusses our direct-to-indirect acoustic radiance transfer algorithm, and we describe its implementation in Section 5. Section 6 presents some experimental results and compares the performance of our algorithm with prior methods.

## 2 RELATED WORK

In this section, we give a brief overview of prior algorithms for radiance transfer computation and diffuse reflections.

**Numerical Acoustics** The propagation of sound in a medium is governed by the acoustic wave equation, a second-order partial differential equation [7]. Several techniques (e.g., finite difference time-domain method) are known for directly solving the wave equation using numerical methods [8], [9] and accurately modeling sound propagation in a scene. Modeling diffuse reflections is essentially a matter of specifying appropriate boundary conditions to the numerical solver and performing the simulation on a grid fine enough to capture the detailed “roughness” of the surfaces that results in acoustic wave scattering [10]. However, despite recent advances [9], these methods can be rather slow and are mainly limited to simple static sources. Precomputation-based methods have recently been developed [11] that use a numerical simulator to compute the acoustic response of a scene from several sampled source positions; at runtime these responses are interpolated given the actual source position. These methods are fast, but require large amounts of precomputed data.

**Geometric Acoustics** Most sound propagation techniques used in practical applications model the acoustic effects of an environment in terms of linearly propagating rays or 3D volumes. These geometric acoustics techniques are not as accurate as numerical methods in terms of solving the wave equation, and cannot easily model all kinds of propagation effects, but allow efficient simulation of early reflections.

Methods based on ray tracing [12], [13] are able to model both diffuse and specular reflections of sound. Since early specular reflections provide the listener with important perceptual cues regarding the direction of sound, specialized techniques have been developed for modeling specular reflections, which include volume tracing [14], [15] and the image source method [16], [17]. For static scenes, which frequently arise in architectural acoustics and virtual environments, radiance transfer methods can be used to simulate reflections from surfaces with arbitrary BRDFs [18], [19]. Many techniques have also been designed to simulate edge diffraction [20], [21], [22].

**Precomputed Light Transport** The radiosity algorithm [23] is the classic example of an algorithm which precomputes light transport effects in a scene. However, the classic radiosity algorithm uses a full solution that needs to be recomputed every time the light source moves. In contrast, *precomputed radiance transfer* (PRT) algorithms essentially decouple light transport effects from the light source configuration.

This is performed by computing a linear operator that defines how a variable light source configuration affects the radiances at sample points on the surfaces in the scene. PRT techniques can support both distant [4], [24] and local [25] source configurations.

*Direct-to-indirect transfer* algorithms [5], [6] are one class of precomputed light transport algorithms. These algorithms compute linear operators which map direct light incident on the surface samples to multi-bounce indirect light at the samples. These algorithms are designed to handle diffuse reflections, and some of them can also support limited glossy reflections. In order to reduce the storage and processing requirements, these techniques project the radiance function over the surface of the scene into a hierarchical basis, such as Haar wavelets [26] or the spectral mesh basis [27]. Our approach is based on applying these ideas to sound propagation.

## 3 OVERVIEW

We present a direct-to-indirect acoustic radiance transfer technique to accelerate the radiance transfer computations for sound propagation so that the first few orders of diffuse reflection can be computed efficiently, for static scenes with moving sources.

### 3.1 Sound Rendering vs. Visual Rendering

There are some key differences between the nature of light and sound waves. In this section, we give a brief overview of these differences. For further details, we refer the reader to surveys of sound rendering algorithms [28]. With light transport simulation, we are mainly concerned with the *steady-state* values of radiance over the surface of the scene. This is because light travels fast enough that transient radiance values are not observed and can be ignored. However, the speed of sound in air is much slower (340 m/s for sound as compared to  $3 \times 10^8$  m/s for light), and hence it is important to compute *time-varying* radiances over the surface.

Another key difference between light and sound is that the wavelengths of sound waves are much larger than the wavelengths of light waves, and are comparable to the sizes of obstacles in typical architectural and game scenes. Therefore, diffraction plays an important role in sound propagation, and it must be modeled in order to generate plausible sounds. In the rest of the paper, we limit ourselves to modeling diffuse reflections; our approach can be combined with other algorithms for computing specular reflection, edge diffraction, etc.

The basic sound propagation problem is: Given the signal emitted by a source (i.e., a time-varying pressure wave), compute the signal heard by a listener after modeling the reflections, diffractions and interferences with the environment. This is typically performed using *impulse responses* (IRs). An IR describes

the sound heard at the listener if the source emits a unit impulse at  $t = 0$ . Under the assumptions of room acoustics [29], the sound heard by the listener for an arbitrary source sound can be computed by convolving the source sound with the IR at the listener’s location. Therefore, for the remainder of this paper, we shall be concerned with computing IRs given the source and listener positions and a geometric model of the scene along with the material properties.

### 3.2 Acoustic Rendering Equation

As a geometric approximation to the acoustic wave equation, the propagation of sound in a scene can be modeled using an extension of the standard graphics rendering equation [30], called the *acoustic rendering equation* [18]:

$$L(x', \omega) = L_0(x', \omega) + \int_S R(x, x', \omega) L\left(x, \frac{x' - x}{|x' - x|}\right) dx \quad (1)$$

where  $L$  is final outgoing radiance,  $L_0$  is emitted radiance and  $R$  is the *reflection kernel*, which describes how radiance at point  $x$  influences radiance at point  $x'$ :

$$R(x, x', \omega) = \rho(x, x', \omega)G(x, x')V(x, x')P(x, x') \quad (2)$$

Here,  $\rho$  is the BRDF of the surface at  $x$ ,  $G$  is the form factor between  $x$  and  $x'$ ,  $V$  is the point-to-point visibility function, and  $P$  is a *propagation term* [18] that takes into account the effect of propagation delays. The latter is unique to sound rendering as visual rendering algorithms neglect propagation delays due to the high speed of light.

Note that the radiances used in Equation 1 are functions of time – typically impulse responses. The time variable  $t$  is hidden in Equation 1 for the sake of brevity. This added dimension of time complicates the storage and processing requirements of sound propagation algorithms based on the acoustic rendering equation. Visual rendering algorithms typically model glossy reflections using a directionally-varying radiance function, which can be represented using spherical harmonics or some other directional basis [4], [26]. In order to model sound reflections using time-varying radiances, a basis such as the Fourier basis is typically used [19].

### 3.3 Impulse Response Representation

In this section, we describe the methods we use to represent impulse responses in the frequency domain. The goal of our algorithm is to compute impulse responses at surface samples and the listener. The impulse response at a point is a function of time, denoted by  $f(t)$ . In order to apply attenuation and delay using a unified formulation [19], we represent our IRs in the frequency domain using the Fourier transform [31].

For a continuous function  $f : [0, T] \rightarrow \mathbb{R}$ , the Fourier transform is defined as (upto a scale factor):

$$\mathcal{F}(f(t)) = F(\omega) = \int_0^T f(t)e^{i\omega t} dt \quad (3)$$

and the inverse Fourier transform is defined as (upto a scale factor):

$$\mathcal{F}^{-1}(F(\omega)) = \hat{f}(t) = \int_{-\infty}^{\infty} F(\omega)e^{i\omega t} d\omega \quad (4)$$

Here,  $F$  is the Fourier transform of  $f$ . The inverse Fourier transform is exact (i.e.,  $\hat{f}(t) = f(t)$ ) only for periodic functions. However, the domain of  $f$  is finite, whereas the domain of  $F$  is infinite. For functions defined over a finite interval  $[0, T]$  (such as IRs), the Fourier transform implicitly assumes that  $f(t)$  is periodic with period  $T$ , and can be extrapolated over the entire real line.

One way to interpret this is that the Fourier transform is a linear transform over infinite-dimensional function space. It transforms functions from the canonical basis (where the basis vectors are Dirac delta functions of the form  $\delta(t - t_i)$ ) to a sinusoidal basis (with basis vectors of the form  $e^{i\omega_i t}$ ). However, this transformation involves a projection into the subspace of periodic functions. Therefore the inverse Fourier transform reconstructs a periodic function which matches the original function in the interval  $[0, T]$ .

In this vector space interpretation of the Fourier transform, Equation 3 is a dot product. This suggests the way to compute the Fourier transform of a function sampled at  $N$  discrete points, or the Discrete Fourier Transform (DFT):

$$F_k = \sum_{i=0}^{N-1} f_i e^{-i\frac{2\pi}{N}ki} \quad (5)$$

and the corresponding inverse DFT:

$$f_i = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{i\frac{2\pi}{N}ki} \quad (6)$$

Here, the  $f_i$ s denote samples of  $f(t)$  at equi-spaced points of time, and  $F_k$ s denote samples of  $F(\omega)$  at equi-spaced values of normalized frequency. Computing the  $N$  dot products of  $N$ -dimensional vectors in Equation 5 would take  $O(n^2)$  time. We compute the DFT using the Fast Fourier Transform (FFT) [32] algorithm, which can compute the DFT of a function with  $N$  samples in  $O(n \log n)$  time by exploiting correlations between elements of the Fourier basis vectors.

The linearity property of the Fourier transform implies that attenuations and accumulation of IRs can be performed easily:

$$\mathcal{F}(af_1(t) + bf_2(t)) = a\mathcal{F}(f_1(t)) + b\mathcal{F}(f_2(t)) \quad (7)$$

Another useful implication is that unlike in time-domain, in frequency-domain delays can also be applied using a scale factor:

$$\mathcal{F}(f(t - \Delta t)) = e^{-i\omega\Delta t}\mathcal{F}(f(t)) \quad (8)$$

Note that care must be taken to ensure that the delays align on sample boundaries, otherwise the inverse Fourier transform will contain non-zero imaginary parts.

If we model a unit impulse emitted by the source at time  $t = 0$  (i.e., the signal emitted by the source has all Fourier coefficients set to 1), then computing the acoustic radiance transfer using the above expressions for delay and attenuation results in a frequency-domain signal. Computing the inverse Fourier transform of this signal using the method described by Siltanen et al [19] yields a periodic function which is approximately equal to the time-domain IR at the listener within the interval  $[0, T]$  for some user-specified maximum IR time  $T$ , which is also the period of the function. It is important to note that this method does not compute the *steady-state* acoustic response, but the time-varying impulse response. The key to this is the frequency-domain delay equations described above.

The Fourier transform lets us store IRs at all the sample points as a collection of  $N$  column vectors, one for each Fourier coefficient. This also allows us to store transfer operators as a collection of  $N$  matrices, and allows us to express direct-to-indirect transfer as  $N$  matrix-vector products.

## 4 ALGORITHM

Our algorithm uses the direct-to-indirect transfer formulation for modeling diffuse reflections of sound. The overall approach is as follows (see Figure 1):

- 1) Choose a number of sample points on the scene surfaces. We discuss our sampling approach in Section 5.
- 2) Given the source position, compute the direct impulse response at each sample. This is performed at runtime using ray tracing.
- 3) Given the direct responses, compute the “indirect” response at each sample (for some user-specified number of diffuse reflections). This is expressed as a matrix-vector multiplication, where the transfer matrix is precomputed. Note that this formulation requires that the scene be static, otherwise the transfer matrix would need to be recomputed every frame.
- 4) Given the direct and indirect responses, compute the final IR at the listener position. This is performed at runtime using raytracing.

### 4.1 Direct-to-Indirect Acoustic Radiance Transfer

Let us assume that the surface is discretized into  $p$  samples. The transfer operator can then be computed

in terms of the direct impulse responses at all samples to impulses emitted from every other sample. Since these are time-varying impulse responses, we can use Fourier coefficients to represent the signals in the frequency domain. Let there be  $f$  Fourier coefficients per surface sample. We then perform our computations on each frequency band (each band corresponding to one Fourier coefficient) independently. From the definition of the Fourier basis functions, we see that the Fourier coefficients have frequencies  $\omega_k = \frac{2\pi}{f}k$ , for  $k = 0 \dots f - 1$ .

For each frequency  $\omega_k$ , we define acoustic radiance vectors of the form  $\mathbf{l}(\omega_k)$ , which contain  $p$  elements that represent the  $k^{\text{th}}$  Fourier coefficients of the IRs at each patch. For the sake of brevity, we shall omit the parameter  $\omega_k$  from the equations in the rest of the paper as it may be obvious from the context. All of the computations we describe in the rest of this section are repeated for each frequency  $\omega_k$ .

If we take the Neumann series expansion of Equation 1 and express it in matrix form, we get:

$$\mathbf{l}_{n+1}(\omega_k) = \mathbf{T}(\omega_k)\mathbf{l}_n(\omega_k) \quad (9)$$

where  $\mathbf{l}_n(\omega_k)$  is the  $k^{\text{th}}$  Fourier coefficient of the IRs at each surface sample after  $n$  reflections. The *transfer matrix*  $\mathbf{T}(\omega_k)$  can be used to compute the effect of one diffuse reflection. The  $(i, j)^{\text{th}}$  element of  $\mathbf{T}(\omega_k)$  describes how the  $k^{\text{th}}$  Fourier coefficient at sample  $j$  affects the  $k^{\text{th}}$  Fourier coefficient at sample  $i$  after one diffuse reflection. The entries of  $\mathbf{T}$  can be computed by shooting rays and computing visibility and form factors between samples. The propagation terms are complex numbers which are computed for each Fourier coefficient using the distances between samples [19].

The above matrix-vector multiplication needs to be performed once per frequency coefficient for each order of reflection at runtime. However, even for scenes of moderate complexity, the number of surface samples,  $p$ , can be very large. Since  $\mathbf{T}$  is a  $p \times p$  matrix and  $\mathbf{l}_n$  is a  $p \times 1$  vector, this step takes  $O(p^2)$  time per frequency coefficient per order of reflection, which can quickly become quite expensive. We use the singular value decomposition (SVD) to compute a rank  $k$  approximation of  $\mathbf{T}$ . This allows us to reduce the complexity to  $O(pk)$ . Next, we show that this approximation can allow us to further accelerate higher-order reflections.

### 4.2 Multiple Bounces and Runtime Complexity

We use the SVD approximation to reduce the complexity of the matrix-vector multiplications to  $O(pk)$  per bounce. However, for multiple bounces, the cost of computing transfer matrices that represent additional bounces can be further reduced to  $O(k^2)$  by precomputing appropriate matrices. Suppose the direct IRs at each surface sample (for a given frequency)

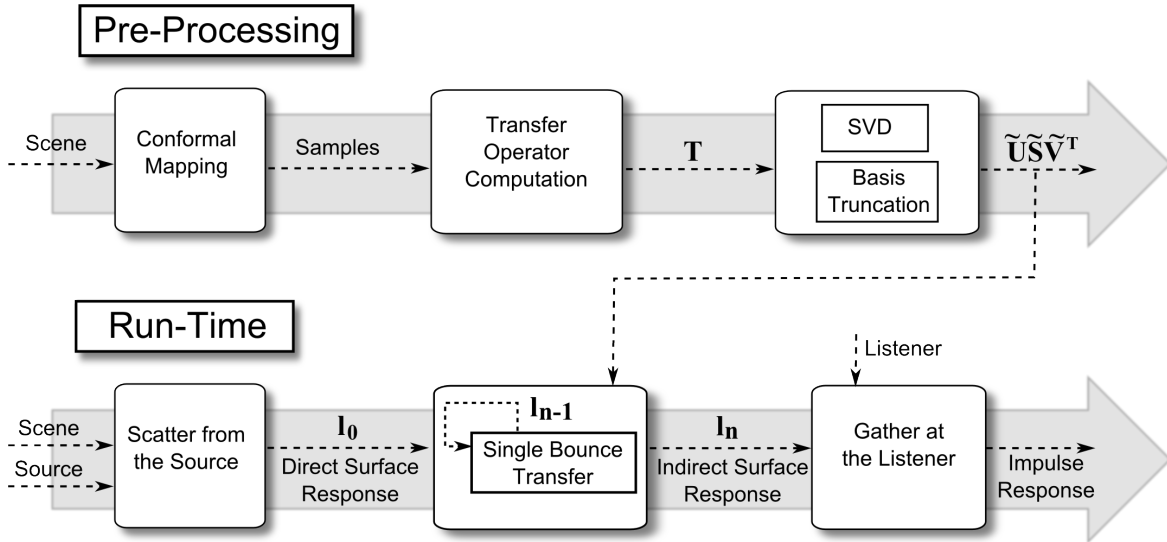


Fig. 1. Overview of our algorithm. In a precomputation step, we sample the surfaces on the scene, and compute a one-bounce transfer operator for these samples ( $\mathbf{T}$ ). We then use the SVD to compute the modes of the transfer operator. At runtime, we shoot rays from the source (which may move freely) and compute direct IRs at the surface samples. We then apply the transfer operator (with a user-specified number of modes retained) repeatedly to quickly obtain the multi-bounce indirect IRs at the surface samples. We then compute the final IR at the listener position in a final gathering step.

are stored in a vector  $\mathbf{l}_0$ . Further suppose we have a rank  $k$  approximation of  $\mathbf{T}$ , given by  $\tilde{\mathbf{T}} = \tilde{\mathbf{U}}\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T$ , where  $\tilde{\mathbf{U}}$  is a  $p \times k$  matrix,  $\tilde{\mathbf{S}}$  is a  $k \times k$  diagonal matrix and  $\tilde{\mathbf{V}}^T$  is a  $k \times p$  matrix. Then the first-order impulse response at each surface sample is given by:

$$\begin{aligned} \tilde{\mathbf{T}}\mathbf{l}_0 &= \tilde{\mathbf{U}}\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T\mathbf{l}_0 \\ &= \tilde{\mathbf{U}}\mathbf{b} \end{aligned}$$

where  $\mathbf{b} = \tilde{\mathbf{S}}\tilde{\mathbf{V}}^T\mathbf{l}_0$  is  $\mathbf{l}_0$  projected into the span of the first  $k$  right singular vectors of  $\mathbf{T}$ . The second-order response is:

$$\begin{aligned} \tilde{\mathbf{T}}\tilde{\mathbf{T}}\mathbf{l}_0 &= \tilde{\mathbf{U}}(\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T\tilde{\mathbf{U}})\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T\mathbf{l}_0 \\ &= \tilde{\mathbf{U}}\mathbf{D}\mathbf{b} \end{aligned}$$

where  $\mathbf{D} = \tilde{\mathbf{S}}\tilde{\mathbf{V}}^T\tilde{\mathbf{U}}$  is essentially the one-bounce operator in the  $k$ -dimensional subspace spanned by the singular vectors corresponding to the top  $k$  singular values of  $\mathbf{T}$ . The cost of multiplying  $\mathbf{b}$  by  $\mathbf{D}$  is simply  $O(k^2)$ . Notice that the third-order response can be written as  $\tilde{\mathbf{U}}\mathbf{D}^2\mathbf{b}$ , and so on. This allows us to compute higher-order responses using a  $k \times k$  matrix instead of a  $p \times p$  matrix. Thus we can quickly compute the first few orders of diffuse reflection at runtime; the approximation error due to the SVD is compensated by the gain in performance.

## 5 IMPLEMENTATION

In this section, we discuss some important aspects of our implementation. Our implementation is CPU-based, and uses Microsoft DirectX 9 for visualization,

and Intel Math Kernel Library (MKL) for the matrix operations.

### 5.1 Approximations

Our algorithm allows for the following user-controlled approximations: *surface samples*, both in number and location; *frequency samples*, i.e. the number of Fourier coefficients used to represent IRs; and *number of modes* retained after computing the SVD of the transfer operator.

In our implementation, we compute a surface parameterization which maps the scene primitives to the unit square (essentially a  $uv$  texture mapping). This parameterization is computed using Least Squares Conformal Mapping (LSCM) [33]. We allow the user to specify the texture dimensions; each texel of the resulting texture is mapped to a single surface sample using an inverse mapping process. The number of texels mapped to a given primitive is weighted by the area of the primitive, to ensure a roughly even distribution of samples. We chose the LSCM algorithm for this purpose since our modeling tools (Blender<sup>1</sup>) have an implementation built-in; it can be replaced with any other technique for sampling the surfaces as long as the number of samples generated on a primitive is proportional to its area.

Our implementation allows the user to vary the number of Fourier coefficients used to represent the IRs. It has been shown [19] that 1K Fourier coefficients

1. <http://www.blender.org>

can provide an acceptable compromise between performance and quality, and therefore we use 1K Fourier coefficients for all our experiments.

We measure the error caused by the SVD approximation of the transfer operator in terms of the Frobenius norm. Figure 2 plots the Frobenius norm error against the number of modes retained for the transfer operator ( $\mathbf{T}$ ). The figure clearly shows that we could use a very small number of modes to compute IRs with diffuse reflections at runtime.

## 5.2 Audio Processing

As described in Section 3 the pressure signal at a source is convolved with an IR from the source to a listener to compute the final audio at the listener. The algorithm presented in Section 4 computes a frequency domain energy IR with 1K Fourier coefficients. The pressure IR is computed from the energy IR [34] and upsampled to encode the desired propagation delay in the IR [19].

**Moving sources and listeners:** In typical virtual environments applications, the source and listener are moving and the audio is streaming from a source in chunks of audio samples (called audio frames). The size of the audio frames is determined by the allowed latency for the application. We choose audio frames of 4800 samples at sampling rate of 48KHz leading to a 100ms latency in our audio output. For a static source and listener, computing the final audio is trivial and amounts to convolving each audio frame with the IR to compute output audio frames. For moving sources and listeners, IRs evolve over time which could lead to discontinuities in the final audio when using different IRs for two adjacent audio frames. In order to minimize such discontinuity artifacts, windowing [35] is applied at the source frame and the listener frame when the source and listener are moving respectively. The windowing applied by our audio processing step is similar to Siltanen et. al. [19].

## 6 RESULTS

In this section, we present some experimental results on the performance and quality achieved by our implementation of our algorithm. All of our tests were performed on an Intel Xeon X5560 workstation with 4 cores (each operating at 2.80 GHz) and 4GB of RAM running Windows Vista. MKL parallelizes our matrix operations over all 4 cores of the test machine. Therefore, the timings we report are for all 4 cores. We have benchmarked our implementation on three scenes whose complexity is typical of scenes encountered in acoustics applications. Figure 3 shows these scenes along with some details.

For comparison, we chose the state-of-the-art frequency acoustic radiance transfer algorithm [19]. To the best of our knowledge, the only other algorithms for simulating diffuse reflections of sound are

naïve time-domain radiance transfer and path tracing. Naïve time-domain radiance transfer would require a large amount of memory, and since the frequency-domain radiance transfer approach [19] is superior to it in this regard, we chose not to compare against the time-domain radiance transfer approach. Path tracing can be used for dynamic scenes, however, since the scene would have to be traversed millions of times per frame, we choose not to compare our algorithm with path tracing, since we restrict ourselves to static scenes.

The frequency-domain acoustic radiance transfer method [19] works by computing the transfer operator (without any SVD approximation) and iteratively applying it to the direct acoustic response until the solution converges. In this sense, it is similar to radiosity, and can be considered to compute the acoustic equivalent of global illumination. In order to perform a fair comparison, we restrict it to computing the same orders of reflection as our algorithm.

Table 1 summarizes the performance characteristics of the precomputation and runtime stages of our algorithm. The complexity of the runtime steps depends on the number of modes retained during the SVD approximation; the table clearly highlights this dependency. As can be seen from the table, our algorithm allows a very efficient update of the impulse responses when the source position changes in the run-time step as compared to the state-of-the-art techniques [19]. Note that we precompute a one-bounce transfer operator, and use the approach described in Section 4.2 to handle multiple orders of reflection at runtime. Depending on the application, we could also precompute a multi-bounce operator and apply it directly at runtime, thus improving our performance further. However, we have used the more flexible approach of varying the orders of reflection at runtime in our implementation. This way, the speedups we demonstrate are *lower bounds* on the speedups possible for our implementation; precomputing a multi-bounce transfer operator would result in even greater performance gains.

Table 2 shows the benefit of the SVD in reducing the memory required to store the transfer operator. The table shows that without SVD, which is essentially a form of lossy compression, the transfer operators may be too large to be used on everyday hardware. For the uncompressed (“reference”) case, the transfer operator size is  $n \times n$ , for each Fourier coefficient (1K in our case). For the compressed (“50 Modes”) case, the transfer operator size is  $n \times k$  for  $\tilde{\mathbf{U}}$ ,  $k \times k$  for  $\mathbf{D}$  and  $k \times n$  for  $\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T$ , where  $k$  is the number of modes retained. In the table,  $k = 50$ , and  $n$  is the number of surface samples in the scene.

Table 3 compares the runtime performance of our method to the performance of acoustic radiance transfer [19]. The timings in the table describe the time

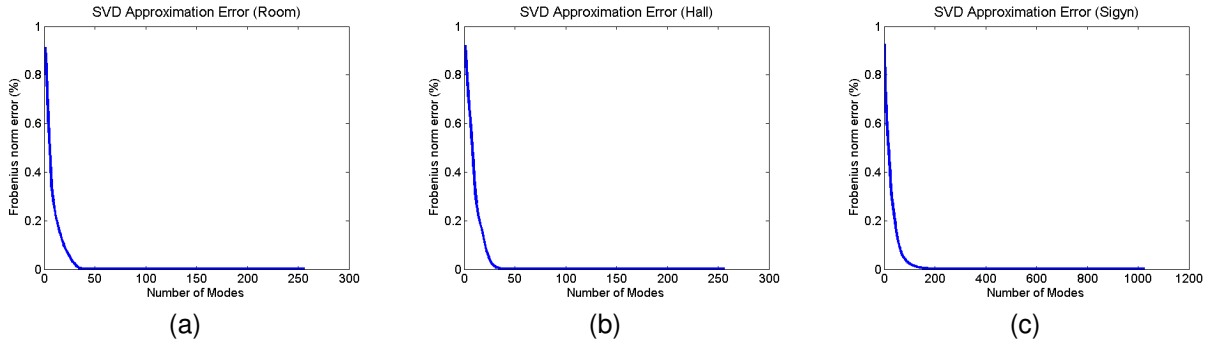


Fig. 2. SVD approximation error for transfer operators. For each benchmark scene, the plots show the relative Frobenius norm of rank- $k$  approximations of  $\mathbf{T}$  (for one value of  $\omega$ ) for all possible values of  $k$ . From left to right: (a) Room, (b) Hall, (c) Sigyn.



Fig. 3. Benchmark scenes. From left to right: (a) Room (252 samples), (b) Hall (177 samples), (c) Sigyn (1024 samples).

Scene	Precomputation Time		Modes	Runtime		
	$\mathbf{T}$	SVD		Initial Scatter	Transfer Operator	Final Gather
Room	14.2 s	94.5 s	10	43.2 ms	24.0 ms	33.7 ms
			25	45.8 ms	43.8 ms	35.0 ms
			50	42.4 ms	84.3 ms	36.4 ms
Hall	13.1 s	93.1 s	10	37.8 ms	26.8 ms	31.5 ms
			25	37.1 ms	45.5 ms	30.2 ms
			50	36.6 ms	79.7 ms	31.2 ms
Sigyn	6.31 min	50.9 min	50	164.1 ms	218.1 ms	109.9 ms

TABLE 1

Performance characteristics of our algorithm. For each scene, we present the precomputation time required by our algorithm for 1K Fourier coefficients. Under precomputation time, we show the time required to compute the transfer operator,  $\mathbf{T}$ , and the time required to compute its SVD approximation. We also compare running times for varying numbers of modes from the SVD. The table shows the time spent at runtime in initial shooting from the source, applying the transfer operator, and gathering the final IR at the listener position.

Scene	Samples	Reference	50 Modes
Room	177	250.6	161.6
Hall	252	508.0	221.6
Sigyn	1024	8388.6	839.2

TABLE 2

Memory requirements of the transfer operators computed by our algorithm with (column 4) and without (column 3) SVD compression. Note that since the entries of each matrix are complex numbers, each entry requires 8 bytes of storage. All sizes in the table are in MB.

required to update the IRs at the listener position when the source position is changed. The table clearly shows the advantage of our approach. Since our precomputed transfer operator is decoupled from the source position, moving the source does not require us to recompute the transfer operator; this fact allows us to update the source position much faster than a straightforward acoustic radiance transfer technique.

Figure 4 shows some impulse responses computed by our algorithm, as compared with the reference acoustic radiance transfer case. As the figure shows, reducing the number of modes has very little effect on the overall shape of the IRs. Coupled with the memory savings demonstrated in Table 2 and perfor-

Scene	Orders	Radiance Transfer	Direct-to-Indirect Transfer (50 modes)
Room	2	11.7 s	131.8 ms
	5	11.8 s	154.4 ms
	10	12.0 s	179.3 ms
Hall	2	10.6 s	116.5 ms
	5	10.7 s	147.3 ms
	10	10.9 s	170.5 ms
Sigyn	2	185.3 s	468.5 ms
	5	186.7 s	491.2 ms
	10	188.7 s	512.8 ms

TABLE 3

Comparison of our approach with a straightforward Acoustic Radiance Transfer approach [19]. We compare the time required by our algorithm to update the source position and recompute the IR at the listener position after a varying number of diffuse reflections. Since the Acoustic Radiance Transfer approach does not decouple the transfer operator from the source position, it needs to perform a costly recomputation of the transfer operator each time the source moves. On the other hand, our algorithm quickly updates the direct IR at all surface samples, then applies the precomputed transfer operator. This allows our approach to handle moving sources far more efficiently than the state-of-the-art.

mance advantage demonstrated in Table 3, we see that using the SVD allows us to significantly reduce memory requirements and increase performance without significant adverse effects on the IRs computed. Of course, the best way to demonstrate the benefit of our approach is by comparing audio clips; for this we refer the reader to the accompanying video.

## 7 CONCLUSION

We have described a precomputed direct-to-indirect transfer approach to solving the acoustic rendering equation in the frequency domain for diffuse reflections. We have demonstrated that our approach is able to efficiently simulate early diffuse reflections for a moving source and listener in static scenes. In comparison with existing methods, our approach offers a significant performance advantage when handling moving sources.

### 7.1 Limitations

Our approach has some limitations. Since it is a precomputed acoustic radiance transfer algorithm, it cannot be used for scenes with dynamic objects. In such situations, algorithms based on path tracing are the best available choice. However, in many applications, including games and virtual environments, scenes are entirely or mostly static, with relatively few moving parts, and hence our algorithm can still be used to model reflections within the static portions of the scene.

Our algorithm performs matrix-vector multiplications on large matrices at runtime. The size of the matrix still depends on the surface size and complexity of the scene. Therefore, our method is useful mainly for scenes of low to medium complexity.

### 7.2 Future Work

Our algorithm uses 1K Fourier coefficients per IR, and this significantly increases our memory requirements. It is crucial to develop an approach to reduce these storage requirements in order to make it feasible to implement on hardware such as video game consoles. It might be possible to use a representation based on Raghuvanshi et al’s precomputed numerical simulation algorithm [11].

In most complex scenes, each surface sample may influence only a subset of all samples in the scene, due to occlusion effects. This observation motivates us to subdivide the scene into cells separated by portals. We could compute transfer operators for each cell independently, and model the interchange of sound energy at the portal boundaries. Cells and portals have been previously used to model late reverberation [36], and would be a promising research direction for diffuse reflections.

The acoustic response over the surfaces of the scene typically are smooth functions. Therefore, it would be beneficial to exploit the spatial coherence of IRs by projecting the transfer operator into basis functions defined over the surfaces of the scene. Furthermore, it might be interesting to investigate the possibility of applying direct-to-indirect transfer techniques to the problem of non-diffuse reflections or edge diffractions.

## REFERENCES

- [1] B.-I. Dalenbäck, M. Kleiner, and P. Svensson, “A macroscopic view of diffuse reflection,” *J. Audio Eng. Soc.*, vol. 42, pp. 793–807, 1994.
- [2] B.-I. Dalenbäck, “The Importance of Diffuse Reflection in Computerized Room Acoustic Prediction and Auralization,” in *Proceedings of the Institute of Acoustics (IOA)*, vol. 17, no. 1, 1995, pp. 27–33.

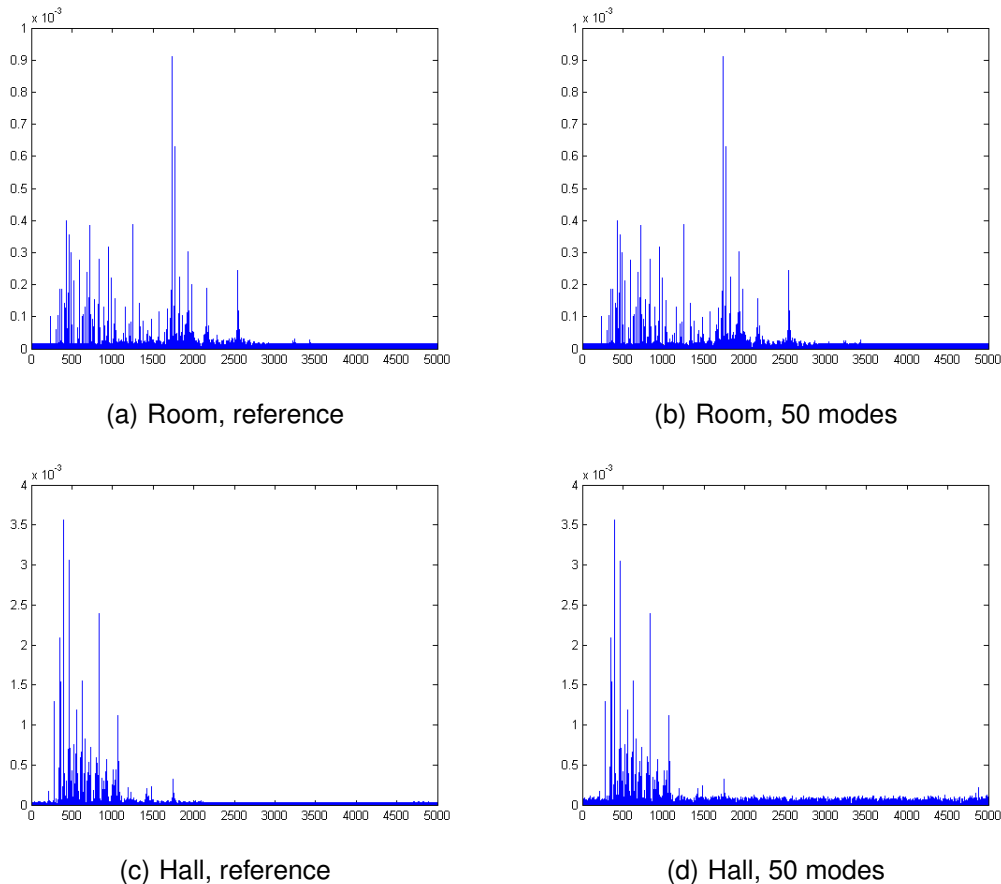


Fig. 4. Comparison of second order diffuse IRs computed by our system with and without SVD compression, for some of our benchmark scenes.

- [3] R. R. Torres, M. Kleiner, and B.-I. Dalenbäck, "Audibility of "diffusion" in room acoustics auralization: An initial investigation," *Acta Acustica united with Acustica*, vol. 86, pp. 919–927(9), November/December 2000.
- [4] P.-P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2002, pp. 527–536.
- [5] M. Hašan, F. Pellacini, and K. Bala, "Direct-to-indirect transfer for cinematic relighting," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006, pp. 1089–1097.
- [6] J. Lehtinen, M. Zwicker, E. Turquin, J. Kontkanen, F. Durand, F. X. Sillion, and T. Aila, "A meshless hierarchical representation for light transport," in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*. New York, NY, USA: ACM, 2008, pp. 1–9.
- [7] P. Svensson and R. Kristiansen, "Computational modelling and simulation of acoustic spaces," in *22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, June 2002.
- [8] R. Ciskowski and C. Brebbia, *Boundary Element methods in acoustics*. Computational Mechanics Publications and Elsevier Applied Science, 1991.
- [9] N. Raghuvanshi, N. Galoppo, and M. C. Lin, "Accelerated wave-based acoustics simulation," in *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*. New York, NY, USA: ACM, 2008, pp. 91–102.
- [10] F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*. Springer Verlag AG, 1998.
- [11] N. Raghuvanshi, J. Snyder, R. Mehra, M. Lin, and N. Govindaraju, "Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes," in *Proc. SIGGRAPH 2010 (To appear)*, 2010.
- [12] M. Vorlander, "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*, vol. 86, no. 1, pp. 172–178, 1989. [Online]. Available: <http://link.aip.org/link/?JAS/86/172/1>
- [13] B. Kapralos, M. Jenkin, and E. Miliotis, "Sonel mapping: acoustic modeling utilizing an acoustic version of photon mapping," in *In IEEE International Workshop on Haptics Audio Visual Environments and their Applications (HAVE 2004)*, 2004, pp. 2–3.
- [14] A. Chandak, C. Lauterbach, M. Taylor, Z. Ren, and D. Manocha, "Ad-frustum: Adaptive frustum tracing for interactive sound propagation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1707–1722, 2008.
- [15] M. T. Taylor, A. Chandak, L. Antani, and D. Manocha, "Re-sound: interactive sound rendering for dynamic virtual environments," in *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*. New York, NY, USA: ACM, 2009, pp. 271–280.
- [16] 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*, 1998, pp. 21–32.
- [17] A. Chandak, L. Antani, M. Taylor, and D. Manocha, "Fastv: From-point visibility culling on complex models," *Computer Graphics Forum*, vol. 28, pp. 1237–1246(10), 2009. [Online]. Available: <http://www.ingentaconnect.com/content/bpl/cgf/2009/00000028/00000004/art00020>
- [18] S. Siltanen, T. Lokki, S. Kiminki, and L. Savioja, "The room acoustic rendering equation," *The Journal of the Acoustical Society of America*, vol. 122, no. 3, pp. 1624–1635, September 2007.
- [19] S. Siltanen, T. Lokki, and L. Savioja, "Frequency domain acoustic radiance transfer for real-time auralization," *Acta Acustica united with Acustica*, vol. 95, pp. 106–117(12),

2009. [Online]. Available: <http://www.ingentaconnect.com/content/dav/aaau/2009/00000095/00000001/art00010>
- [20] U. P. Svensson, R. I. Fred, and J. Vanderkooy, "An analytic secondary source model of edge diffraction impulse responses," *Acoustical Society of America Journal*, vol. 106, pp. 2331–2344, Nov. 1999.
- [21] M. Taylor, A. Chandak, Z. Ren, C. Lauterbach, and D. Manocha, "Fast edge-diffraction for sound propagation in complex virtual environments," in *EAA Auralization Symposium*, 2009.
- [22] D. Schröder and A. Pohl, "Real-time hybrid simulation method including edge diffraction," in *EAA Auralization Symposium*, 2009.
- [23] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 213–222, 1984.
- [24] Y.-T. Tsai and Z.-C. Shih, "All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 967–976, 2006.
- [25] A. W. Kristensen, T. Akenine-Möller, and H. W. Jensen, "Pre-computed local radiance transfer for real-time lighting design," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM, 2005, pp. 1208–1215.
- [26] J. Kontkanen, E. Turquin, N. Holzschuch, and F. Sillion, "Wavelet radiance transport for interactive indirect lighting," in *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*, W. Heidrich and T. Akenine-Möller, Eds. Eurographics, jun 2006. [Online]. Available: <http://artis.imag.fr/Publications/2006/KTHS06>
- [27] R. Wang, J. Zhu, and G. Humphreys, "Precomputed radiance transfer for real-time indirect lighting using a spectral mesh basis," in *Proceedings of the Eurographics Symposium on Rendering*, 2007.
- [28] T. Funkhouser, N. Tsingos, and J.-M. Jot, "Survey of methods for modeling sound propagation in interactive virtual environment systems," *Presence*, 2003.
- [29] H. Kuttruff, *Room Acoustics*. Elsevier Science Publishing Ltd., 1991.
- [30] J. T. Kajiya, "The rendering equation," in *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1986, pp. 143–150.
- [31] R. N. Bracewell, *The Fourier transform and its applications*. McGraw-Hill, 2000.
- [32] E. O. Brigham, *The Fast Fourier Transform and its applications*. Prentice-Hall, 1988.
- [33] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 362–371, 2002.
- [34] H. K. Kuttruff, "Auralization of Impulse Responses Modeled on the Basis of Ray-Tracing Results," *Journal of the Audio Engineering Society*, vol. 41, no. 11, pp. 876–880, November 1993.
- [35] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, January 1999.
- [36] E. Stavrakis, N. Tsingos, and P. Calamia, "Topological sound propagation with reverberation graphs," *Acta Acustica/Acustica - the Journal of the European Acoustics Association (EAA)*, 2008. [Online]. Available: <http://www-sop.inria.fr/revs/Basilic/2008/STC08>