Interactive Sound Propagation with Bidirectional Path Tracing

Chunxiao Cao* Zhong Ren* Carl Schissler† Dinesh Manocha† Kun Zhou*‡

*State Key Lab of CAD&CG, Zhejiang University †University of North Carolina at Chapel Hill

Abstract

We introduce *Bidirectional Sound Transport (BST)*, a new algorithm that simulates sound propagation by bidirectional path tracing using multiple importance sampling. Our approach can handle multiple sources in large virtual environments with complex occlusion, and can produce plausible acoustic effects at an interactive rate on a desktop PC. We introduce a new metric based on the signal-to-noise ratio (SNR) of the energy response and use this metric to evaluate the performance of ray-tracing-based acoustic simulation methods. Our formulation exploits temporal coherence in terms of using the resulting sample distribution of the previous frame to guide the sample distribution of the current one. We show that our sample redistribution algorithm converges and better balances between early and late reflections. We evaluate our approach on different benchmarks and demonstrate significant speedup over prior geometric acoustic algorithms.

Keywords: sound propagation, bidirectional path tracing

Concepts: •**Computing methodologies** → **Physical simulation;** *Ray tracing;*

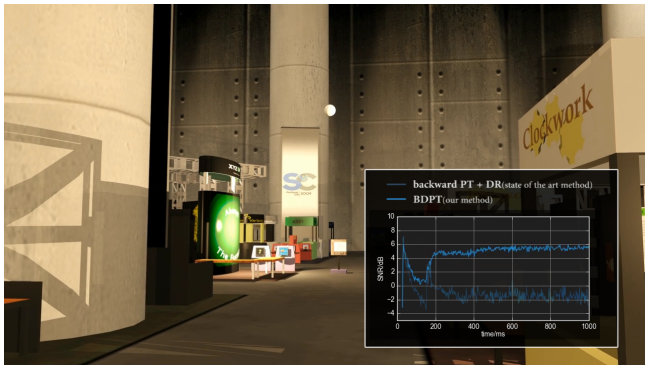


Figure 1: *The "Tradeshow" benchmark rendered by our BST and the backward path tracing method with diffuse rain in a similar propagation time of 18ms/frame. Based on bidirectional path tracing with multiple importance sampling and temporal sample distribution optimization, our method is able to generate energy response with up to a 7dB higher signal-to-noise ratio.*

1 Introduction

The rapid development of consumer virtual reality hardware in recent years has sparked renewed interest in complex virtual environments and in generating high-quality user experiences that involve the use of multiple senses. It is known that a greater correlation between sound and visual rendering can significantly enhance the sense of immersion. While there has been remarkable progress for realistic visual rendering, the generation of high-quality acoustic effects at interactive rates remains a major challenge.

One of the main goals of sound rendering is to simulate the propagation of sound to take into account the environment, source locations and the listener's position. In practice, sound propagation results are combined with spatialized audio rendering for immersive experiences. Prior work in sound simulation algorithms can be classified into two broad categories. The wave-based methods directly solve the acoustic wave equation using numerical methods. The geometric acoustic (GA) algorithms model the propagation of sound based on the concepts of ray tracing. Wave-based algorithms [Brebbia and Ciskowski 1991; Thompson 2006] can accurately simulate all acoustic effects, including diffraction and scattering. However, they are computationally expensive, involve considerable precomputation, and are inefficient for high frequency simulation [Savioja 2010] and limited to static scenes [Mehra et al. 2013; Raghuvanshi and Snyder 2014]. In contrast, GA methods use ray tracing to compute specular and diffuse reflections and can also handle dynamic scenes [Lentz et al. 2007; Schissler et al. 2014]. There is considerable work on extending ray tracing algorithms to approximate edge diffraction [Tsingos et al. 2001] and handle complex environments with multiple sources [Schissler and Manocha 2014]. However, many issues arise in terms of using the resulting methods for interactive sound propagation. Current GA algorithms tend to use trial-and-error methods to select the ray budget – an insufficient number of rays can lead to aliasing artifacts in sound rendering. A larger ray budget slows down the performance, and interactive simulation of higher order reflections or dynamic late reverberation can be challenging. Stochastic methods like path tracing are better at handling diffuse reflections, but their convergence can be a problem, especially for late reverberation. Besides, current GA algorithms do not address issues related to quality differences between early and late reflections, and lack a proper method to balance between them.

In order to address these problems, we first propose the use of the signal-to-noise ratio (SNR) of the energy response of sound sources as the metric to assess the quality of stochastic sound propagation simulation. We then show how to use general bidirectional path tracing and multiple importance sampling in sound propagation simulation and maximize the SNR of the energy response. The result is a novel sound propagation algorithm, called *Bidirectional Sound Transport (BST)*, which is able to handle multiple sources in large virtual environments and provides higher than state-of-the-art quality stochastic sound propagation algorithms with a similar computation cost.

Some of our novel contributions include:

- We propose a new metric for assessing the quality of stochastic sound propagation. With the new metric as a target function, we present a numerical method to calculate the optimal sample distribution. The convergence properties of our method are proved.

*ccx4graphics@gmail.com, renzhong@zju.edu.cn, kunzhou@acm.org

†schissle@cs.unc.edu, dm@cs.unc.edu

‡Corresponding author

SA '16 Technical Papers, December 05-08, 2016, Macao

ISBN: 978-1-4503-4514-9/16/12

DOI: http://dx.doi.org/10.1145/2980179.2982431

- We introduce a new algorithm that simulates the sound propagation by bidirectional path tracing (BDPT) with multiple importance sampling (MIS). The algorithm is able to provide better balance between early and late responses and produces higher quality sound rendering results than the state-of-the-art techniques. To the best of our knowledge, this is the first application of BDPT and MIS to sound propagation and has actually been shown to be able to provide superior rendering results to backward or forward path tracing. BST generates results of more stable quality as the sound sources move around and it is therefore not necessary to conservatively reserve ray budget for difficult sampling cases.
- We analyze the different sound propagation algorithms using the SNR criterion and provide some insights that may be used to design improved sound propagation algorithms.

2 Related Work

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

2.1 Interactive Sound Propagation

Most earlier work in sound propagation was driven by architectural acoustics and noise reduction, and the main focus was on the development of offline algorithms. Over the last few decades, the development of interactive ray tracing algorithms and their variants, along with the hardware capabilities of commodity processors, has resulted in realtime sound propagation algorithms. Some algorithms exploit frame-to-frame coherence or tuning techniques to use fewer rays [Schissler et al. 2014]. A recent survey of GA algorithms is given in [Savioja and Svensson 2015].

There is considerable work on the development of precomputation-based methods for interactive sound propagation in static scenes. These include algorithms based on GA or wave-based methods that store the precomputed impulse responses, transfer functions, or sound pressure fields. In order to handle the large size of these representations, many compression algorithms have been proposed [Antani et al. 2012; Raghuvanshi et al. 2010; Raghuvanshi and Snyder 2014; Mehra et al. 2013] that reduce the runtime memory overhead.

Sound propagation can also be accelerated by computing appropriate simplifications of an acoustic environment. Due to the wave property of sound, geometry details become less important, especially for lower frequencies. The simplest techniques use proxy objects like boxes [Antani and Manocha 2013] for the scene objects. Other algorithms compute levels-of-details (LODs) corresponding to different frequency bands [Pelzer and Vorländer 2010; Schissler et al. 2014]. In order to deal with a high number of sources in the scene, clustering is frequently used to accelerate the computations [Tsingos et al. 2004]. Other techniques exploit frame-to-frame coherence to use only a few rays [Schissler et al. 2014].

2.2 Path Tracing

Due to its ability to handle complex scenes and generate unbiased results, path tracing has been widely used in visual and sound rendering. In order to simulate sound propagation, a traditional path tracer emits random paths from each sound source. The paths interact with the scene objects in various ways, and their contribution is accumulated in the final result once the path intersects with the listener. The listener is typically represented by a detection sphere. In practice, the convergence of path tracing depends on the number of sources, the complexity of the scene and the size of the detection sphere. A detailed discussion can be found in [Vorländer 1988].

Recently, fast algorithms for complex scenes have been proposed based on backward path tracing for many-source scenes [Schissler and Manocha 2014]. While the complexity of forward path tracing scales linearly according to the number of sources, the cost of backward tracing scales sub-linearly. In practice, only a small portion of emitted rays will eventually hit the detection sphere, and the hit probability depends heavily on the size and structure of scene geometry. This characteristic is observed in forward as well as backward path tracing. In order to achieve the best performance in a certain scene, the user must make a trade between speed and accuracy by adjusting the size of the detection sphere radius [Taylor et al. 2012]. Another major disadvantage of backward path tracing is its inability to handle ideal point sources. It is important to support such point sources as they can be used to approximate complex sound sources [Li et al. 2015]. One simple approach is to approximate the point sources with smaller-sized spheres, but this can decrease the efficiency of backward path tracing.

Some of these issues with path tracing can be addressed using the concept of diffuse rain (DR) [Schröder 2011], which can be viewed as a counterpart of next event estimation [Lafortune 1996] or the VPL technique [Keller 1997; Dachsbacher et al. 2014] in graphics rendering. DR requires no detection sphere, but instead builds connections between the sources and the hit points on the emitted paths. For scenes with no occlusion, this method guarantees the validity of every connection. The original diffuse rain algorithm is designed for spherical sources, but it can also support point sources accurately. Moreover, the validity of the connection is independent of the size of the scene objects, which makes this technique attractive for a large variety of scenes.

3 Bidirectional Sound Transport

In this section, we give an overview of our approach, describe the new metric and use that for our BST algorithm. Our formulation is based on the theory of GA, and sound propagation is modeled as energy transport. The main objective of a GA simulator is to compute the energy response $\mathrm{ER}(\mathbf{x}, \omega, t)$ of every sound source, where $\mathbf{x}$, ω, and t represent the listener position, the incident direction, and the propagation delay, respectively. In order to perform sound rendering, the resulting algorithm performs auralization that computes the impulse response (IR) from $\mathrm{ER}()$ and convolves that with the sound source. In the rest of the paper, we mainly deal with the computation of the energy response.

3.1 Background

In this paper, we will use most of the terminology presented in [Siltanen et al. 2007], with a few necessary modification for the convenience of analysis in a path tracing framework.

Acoustic Transport Equation For sound propagation, a time-dependent transport equation is proposed by [Siltanen et al. 2007]. We use a reformulated version of that equation in the rest of the paper:

$$L(\mathbf{x}' \to \mathbf{x}, t) = L_0(\mathbf{x}' \to \mathbf{x}, t) + \int_{\Omega} L_s(\mathbf{x}' \to \mathbf{x}, t) * M(\mathbf{x}'' \leftrightarrow \mathbf{x}', t) dA_{\mathbf{x}''}, \quad (1)$$

where $L_0(\mathbf{x}' \to \mathbf{x}, t)$ represents the radiance emitted by $\mathbf{x}'$, and

$$L_s(\mathbf{x}' \to \mathbf{x}, t) = L(\mathbf{x}'' \to \mathbf{x}', t) G(\mathbf{x}'' \leftrightarrow \mathbf{x}') \rho(\mathbf{x}'' \to \mathbf{x}' \to \mathbf{x}). \quad (2)$$

We use the following symbols in these equations. The time-variant radiance is usually denoted as $L(\mathbf{x}, \omega, t)$, where $\mathbf{x}$, ω, and t repre-

sent the position, direction, and time, respectively. For the convenience of analysis in a path tracing framework, we use $L(\mathbf{x}' \rightarrow \mathbf{x}, t)$ to represent $L(\mathbf{x}', \omega(\mathbf{x}' \rightarrow \mathbf{x}), t)$ so that the relationship between the transportation formula and path nodes is expressed clearly. The BSDF (bidirection scattering distribution function) $\rho(\mathbf{x}'' \rightarrow \mathbf{x}' \rightarrow \mathbf{x})$ and the geometry term $G(\mathbf{x}'' \leftrightarrow \mathbf{x}')$ are represented in a similar manner. BSDF represents the acoustic property of the scene geometry, and the geometry term explains the energy dispersion and occlusion during the propagation. A is the area measure defined on the scene geometry Ω.

The media term, $M(\mathbf{x}'' \leftrightarrow \mathbf{x}', t)$, accounts for energy absorption and time delay caused by the propagation media through the convolution with L_s (the asterisk $*$ is the convolution symbol). For homogeneous media it can be written as:

$$M(\mathbf{x}'' \leftrightarrow \mathbf{x}', t) = e^{-\alpha|\mathbf{x}''-\mathbf{x}'|}\delta(t - \frac{|\mathbf{x}''-\mathbf{x}'|}{c}), \tag{3}$$

where c is the speed of sound, α is the absorption factor, and $\delta(t)$ is the Dirac delta function. This formulation of the rendering equation does not account for diffraction effects.

It should be noticed that the energy response and the time-variant radiance could be expressed in the same form. In actuality, we have

$$\mathrm{ER}(\mathbf{x}' \rightarrow \mathbf{x}, t) = L(\mathbf{x}' \rightarrow \mathbf{x}, t), \tag{4}$$

with $f(\mathbf{x}'' \rightarrow \mathbf{x}' \rightarrow \mathbf{x}) = 1$. One can regard the energy response as the radiance of an infinitely-small sphere at the position of the listener. We will always use $L(\mathbf{x}' \rightarrow \mathbf{x}, t)$ instead of $\mathrm{ER}(\mathbf{x}' \rightarrow \mathbf{x}, t)$ in our analysis.

The transport equation can be rewritten in the operator form:

$$L = L_0 + TL, \tag{5}$$

where T is an operator defined as

$$TL = \int_\Omega L_s(\mathbf{x}' \rightarrow \mathbf{x}, t) * M(\mathbf{x}'' \leftrightarrow \mathbf{x}', t) dA_{\mathbf{x}''}. \tag{6}$$

The rendering equation can be solved with the Neumann series expansion [Kreyszig 1978]:

$$L = (\mathrm{Id} - T)^{-1} L_0 = \sum_{i=0}^{\infty} T^i L_0, \tag{7}$$

which turns Eq. (5) into an infinite sum of integrals that can be evaluated with various numerical methods.

Monte-Carlo Path Tracing Monte-Carlo integration is a popular method that evaluates complex integrals in a probabilistic way. Given a σ-finite measure μ_1 defined on measurable space $(\Sigma, \mathcal{X})$, Monte-Carlo integration evaluates the integral $\int_\Sigma f(x) d\mu_1$ by constructing a probability measure μ_2 and sampling randomly from the probability space $(\Sigma, \mathcal{X}, \mu_2)$. If we let $p(x)$ be the Radon-Nikodym derivative $d\mu_2/d\mu_1$ [Royden and Fitzpatrick 1988], we have

$$E[\frac{f(X)}{p(X)}] = \int_\Sigma f(x) \frac{d\mu_1}{d\mu_2} d\mu_2 = \int_\Sigma f(x) d\mu_1, \tag{8}$$

where $E[X]$ is the expected value of random variable X.

When evaluating $T^i L_0$ with the Monte-Carlo method, μ_1 is the product measure A^i defined on product measurable space Ω^i, commonly referred to as the "path space of bounce i". μ_2 is the probability measure of path generation. A path X is a sample from Ω^i, $f(X)$ is the energy impulse generated by sound propagation along path X, and $p(X)$ is its "probability of generation."

Bidirectional Path Tracing To form the paths used for the Monte Carlo integration in Eq. (8), a bidirectional path tracer [Lafortune and Willems 1993; Veach and Guibas 1995] first generates subpaths from both the source side (*forward subpaths*) and the listener (*backward subpaths*), and then builds connections between the subpath nodes to construct the final paths. These two steps are referred to as the *trace* step and the *connect* step, respectively.

Given a subpath $\mathbf{x}_0 \cdots \mathbf{x}_k$, its generation probability can be expressed in the form below:

$$\begin{aligned} p(\mathbf{x}_0 \cdots \mathbf{x}_k) =& p_g(\mathbf{x}_0 \rightarrow \mathbf{x}_1) G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1) \cdot \\ & \prod_{i=1}^{k-1} p_f(\mathbf{x}_{i-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{x}_{i+1}) G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}). \end{aligned} \tag{9}$$

Here $p_g(\mathbf{x}' \rightarrow \mathbf{x})$ is the probability density for the source to generate a path in the direction $\mathbf{x}' \rightarrow \mathbf{x}$, and $p_f(\mathbf{x}'' \rightarrow \mathbf{x}' \rightarrow \mathbf{x})$ is the probability density for the outgoing path to be in the direction $\mathbf{x}' \rightarrow \mathbf{x}$ when given the incident path direction $\mathbf{x}'' \rightarrow \mathbf{x}'$. These probability density functions depend on the implementation of path emitters and the sound materials.

To simplify the expression, we denote the geometry term $G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1})$ as $G_{i,i+1}$, probability density $p_f(\mathbf{x}_{i-1} \rightarrow \mathbf{x}_i \rightarrow \mathbf{x}_{i+1})$ as V_i $(i > 0)$ and $p_g(\mathbf{x}_i \rightarrow \mathbf{x}_{i+1})$ as V_0, and rewrite Eq. (9):

$$p(\mathbf{x}_0 \cdots \mathbf{x}_k) = \prod_{i=0}^{k-1} V_i G_{i,i+1}. \tag{10}$$

When connecting the s-th node of a forward path and the t-th node of a backward path, we form a complete path of bounce $s+t$. Such a connection is referred to as a (s,t)-connection. For a path $X = \mathbf{x}_0 \cdots \mathbf{x}_{s+t+1}$, its probability of generation by a (s,t)-connection could be written as

$$\begin{aligned} p_{s,t}(\mathbf{x}_0 \cdots \mathbf{x}_{s+t+1}) &= \prod_{k=0}^{s} V_k G_{k,k+1} \prod_{k=s+1}^{s+t+1} G_{k-1,k} V_k \\ &= \frac{\prod_{k=0}^{s+t+1} V_k \prod_{k=0}^{s+t} G_{k,k+1}}{V_s G_{s,s+1} V_{s+1}}. \end{aligned} \tag{11}$$

Combined with $f(X)$, we have $f(X)/p_{s,t}(X)$ as an estimator of $T^{s+t}L_0$. We refer to such an estimator as *estimator* (s,t). It's worth noting that the nominator $V_s G_{s,s+1} V_{s+1}$ is only related to the connection segment $\mathbf{x}_s \mathbf{x}_{s+1}$, and how the connection is formed can have a major influence on the estimator.

Multiple Importance Sampling Given a forward subpath $\mathbf{x}_0 \cdots \mathbf{x}_k$ and a backward subpath $\mathbf{y}_k \cdots \mathbf{y}_0$, there are multiple possible connections between them to form a path and evaluate $T^i L_0$. MIS combines these different *estimators* as:

$$T^i L_0 = E\left[\frac{1}{N} \sum_{j \geq 0, i-j \geq 0} \frac{1}{c_j} \sum_{k=0}^{n_j} \frac{w_{j,i-j}(X_{j,k}) f(X_{j,k})}{p_{j,i-j}(X_{j,k})} \right], \tag{12}$$

where N is the total number of samples for evaluating $T^i L_0$, n_j is the number of samples for bounce j, and $X_{j,k}$, for $k = 1 \cdots n_j$, is the number of samples generated by the $(j, i-j)$-connection. The symbol c_j in Eq. (12) is the probability that a sample belongs to the *estimator* $(j, i-j)$.

A carefully designed weighting function can significantly decrease the variance of the estimation. We use the balance heuristic:

$$w_{k,i-k}(X) = \frac{c_k p_{k,i-k}(X)}{\sum_{j \geq 0, i-j \geq 0} c_j p_{j,i-j}(X)}. \tag{13}$$

Other weighting functions, like the power heuristic, may outperform the balance heuristic in cases where some estimation strategies are much better than others. This could happen in visual rendering when the scene contains glossy objects. However, in the context of sound propagation, the contribution of specular reflection is much less obvious, and the balance heuristic works well.

Sound Rendering vs. Visual Rendering Because sound propagation in GA is similar to light propagation, many algorithms and sampling analyses from visual rendering may be extended to sound simulation. However, there are some key differences between light and sound waves, and we therefore need to develop significant variants.

The spatial resolution of the human auditory system is far lower than that of the visual system. For the localization of a single sound source, the most optimistic error estimation is no less than 0.75 degrees for the azimuth angle and that of the elevation angle is even larger [Blauert 1997]. In comparison, the angular resolution of the human visual system is about $0.01-0.02$ degrees at the foveal area of the retina [DeValois 1988]. These characteristics have a considerable influence on the quality assessment of the final rendering and the choice of a simulation algorithm. For example, an important advantage of photon mapping over path tracing (for visual rendering) is its low-frequency noise distribution. However, human auditory systems are not sensitive to the spatial frequency of noise, and this advantage is lost in the context of sound propagation.

In contrast, temporal information is essential for acoustic simulation. The perception of sound and light are fundamentally different in the way temporal information is processed. Different sound receptors in our cochlea respond to different sound frequencies, which implies that the sound information received by the brain is not temporal, but tempo-spectral. This allows human auditory system to extract more temporal information from sound. The critical flicker fusion frequency of human eyes is generally below 60Hz [Mahneke 1957], while the upper bound of the human audible frequency range is around 20000Hz.

The aforementioned difference has its influence on the design of quality metric and propagation algorithms. In visual rendering, the quality metric is generally defined as a function of position. In sound rendering, however, one would expect the metric to be a function of time. Therefore, controlling the quality in the temporal dimension is an important task for sound propagation. For GA algorithms that involve stochastic sampling, quality control of the temporal dimension requires the modification of the temporal sample distribution. However, unlike sampling direction, the time delay of a sample depends on its propagation distance, which cannot be manipulated directly. Therefore, we need an indirect method to control the temporal distribution of samples.

3.2 SNR Metric for Stochastic Sound Propagation

In general, there is no well-accepted standard for evaluating the quality of acoustic rendering. Current assessment methods usually involve investigation of user experiences [Rychtáriková et al. 2011; Nicol et al. 2014] and/or real world measurements [Pelzer et al. 2011]. However, capturing measurements for all kinds of environments can be very challenging and expensive, and the current perceptual evaluations of sound are restricted to a few specific environments. A key issue in the design of a good simulation algorithm is to have an appropriate quality metric. Such a metric should be relatively easy to evaluate and should also correlate well with perceptual evaluation. Because our sound simulation algorithm deals with computing an energy response and uses the characteristics of the acoustic transport equation, it's natural to use the accuracy of the energy response as the quality metric. As a result, we propose the use of the signal-to-noise ratio (SNR) of the energy response as the quality metric in our formulation. In particular, the SNR of a random variable X is defined as

$$\mathrm{SNR}[X] = \frac{E[X]}{\sigma[X]}, \mathrm{SNR}_{\mathrm{dB}}[X] = 10\log_{10}(\frac{E[X]}{\sigma[X]}). \tag{14}$$

In practice, when we calculate the SNR of the energy response, we render many frames with different random seeds under the same conditions. The results are integrated over different time spans (referred to as "bins") and in all incident directions. The SNR is calculated with the mean value of these integrals as $E[X]$ and the sample variance is represented as $\sigma^2[X]$. These SNR values are arranged in the time domain as the "SNR curve" as shown in various figures in the rest of the paper for quality comparison. In most benchmarks, the size of the bins is 3ms.

As compared to other quality criteria like energy response variance and spectrogram variance, energy response SNR has many important advantages. It has a simple formulation and does not require expensive operations like FFT to evaluate. Moreover, it is independent of the source intensity. In sound propagation, it is very important to differentiate between early and late reflections, and energy response SNR can easily handle such cases.

3.3 Bidirectional Sound Transport Algorithm

In this section, we present our novel sound propagation algorithm based on BDPT.

Existing GA algorithms generate paths from the source or listener to evaluate the Monte Carlo integration (Eq. (8)). To improve the validity of generated paths, diffuse rain [Schröder 2011] is used to build connections between the source/listener with the hit points on the emitted paths. In the view of bidirectional path tracing, this technique builds connections between source/listener side subpaths with the listener/sources to form a complete path, and can be regarded as a special case of BDPT. On the other hand, since the paths generated in this way have zero bounces on either source or listener side, there is only one estimator for each $T^i L_0$ and the benefit of MIS cannot be exploited. As a result, in scenarios where different paths have a radiance with a large difference, diffuse rain may generates energy responses with low SNR.

One natural way of solving the above problem is to use general BDPT for sound propagation in combination with MIS. For sound propagation, however, there is a complexity of temporal sample distribution. Paths of different lengths correspond to different delay values and contribute to different parts of the energy response:

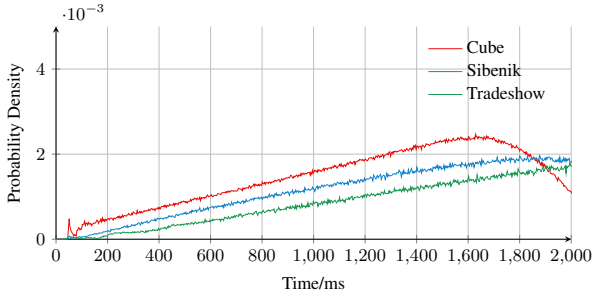
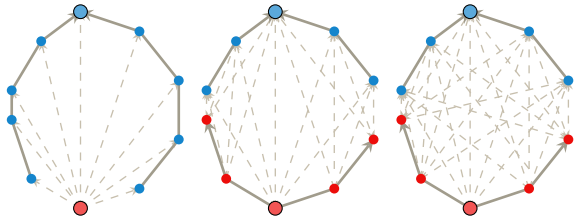


Figure 2: *Temporal sample distribution of the original BDPT algorithm in different benchmarks (see Sec. 5 for more details). When straightforwardly applied to sound propagation, the original BDPT favors a certain part of the response due to the way subpaths are connected.*

short paths for early response and long paths for late response. The natural BDPT implementation favors a certain part of the response, which corresponds to path lengths that can be formed by more possibilities of connecting source and listener side subpaths. See Fig. 2 for an illustration of this effect in some test scenes.

Direct manipulation of path lengths is difficult because the subpaths and connections are generated stochastically. In the following, we propose an algorithm to indirectly control the temporal distribution of samples. The basic idea is to exploit the coherence between bounce number and path length, and allocate samples among the path spaces of different bounces in every frame. We also exploit the frame coherence and use the temporal sample distribution and variance of the previous frame to derive the temporal sample distribution of the current frame.

Path Generation Traditionally in bidirectional path tracing, forward subpaths and backward subpaths are generated in pairs, and connections are built between the subpaths that belong to the same pair. In our approach, we use a connecting strategy that is similar to [Pajot et al. 2011; Popov et al. 2015]. There is no restriction on the correspondence of forward and backward subpaths in terms of building connections, and any kind of connection between the forward and backward subpath nodes is permitted. We illustrate the difference in path generation of diffuse rain, original BDPT and our method in Fig. 3.



(a) backward PT+DR (b) traditional BDPT (c) our method

Figure 3: *Diagrams of connection schemes in different path tracing algorithms, including backward path tracing combined with DR, traditional BDPT, and our algorithm. Path nodes are colored in red for backward subpaths and blue for forward subpaths.*

Sample Allocation among Integrals The samples generated by the path tracer can be categorized by the number of bounces along the path. Such an interaction is represented as T in our operator-form transport equation (5), and the samples of bounce i are used to evaluate $T^i L_0$ in Eq. (7). With our new connection scheme, the number of potentially valid paths is considerably larger, and we will use only a small portion of them to produce the final result. We need a scheme to determine the number of samples that are used to evaluate each $T^i L_0$.

We will consider the natural BDPT sample distribution first. Suppose that we have a forward subpath $\mathbf{x}_0 \cdots \mathbf{x}_k$ and a backward subpath $\mathbf{y}_k \cdots \mathbf{y}_0$. By connecting $\mathbf{x}_s$ and $\mathbf{y}_t$,we have a sample of bounce $s+t$. For any $k_0 \leq k$ (as the case $k_0 > k$ will cause problems for MIS), there are k_0+1 combinations for s and t that satisfy $s+t=k_0$, which means that the number of samples increases linearly with path bounces. Thus the natural sample distribution is more favorable towards the late responses.

We use the SNR criterion to guide an optimal sample distribution. We use M sample bins, N integrals and S samples. We denote the contribution from the n-th integral to the m-th sample bin as X_{mn}, the variance of a single sample contribution from the n-th integral to the m-th bin as σ^2_{mn}, and the sample probability for the n-th integral as x_n. All the random variables X_{mn} are supposed to be mutually independent. In this case, the sum of the energy response SNR is given as

$$\begin{aligned}
&\sum_{m=1}^{M} 10\log_{10}\frac{E[\sum_{n=1}^{N} X_{mn}]}{\sigma[\sum_{n=1}^{N} X_{mn}]} \\
&= \sum_{m=1}^{M} 10\log_{10} E[\sum_{n=1}^{N} X_{mn}] - \sum_{m=1}^{M} 5\log_{10}\sigma^2[\sum_{n=1}^{N} X_{mn}] \quad (15)\\
&= \sum_{m=1}^{M} 10\log_{10} E[\sum_{n=1}^{N} X_{mn}] - \sum_{m=1}^{M} 5\log_{10}(\sum_{n=1}^{N}\frac{\sigma^2_{mn}}{Sx_n}).
\end{aligned}$$

Notice that $\sum_{m=1}^{M} 10\log_{10} E[\sum_{n=1}^{N} X_{mn}]$ is a constant. We need to minimize the target function

$$\sum_{m=1}^{M} 5\log_{10}(\sum_{n=1}^{N}\frac{\sigma^2_{mn}}{Sx_n}) \quad (16)$$

to maximize the total SNR. In order to optimize the target function above, we devised an iterative algorithm based on the Lagrange Multiplier Method [Bertsekas and Nedic 2003]. The iteration step of this algorithm is shown below:

$$x_{n,i+1} = \frac{\alpha}{M}\sum_{m=1}^{M}\frac{\frac{\sigma^2_{mn}}{x_{n,i}}}{\sum_{k=1}^{N}\frac{\sigma^2_{mk}}{x_{k,i}}} + (1-\alpha)x_{n,i}. \quad (17)$$

Here α is a positive value smaller than 1. In practice, we usually choose a value between 0.4 and 0.7 (see our supplementary material). To avoid division by zero, $x_{n,0}$ must be all positive and there must be at least one $\sigma_{mn} > 0$ for every m and n. Sample bins and integrals that cannot satisfy this restriction can be ignored without any consequence. The convergence proof of this algorithm is present in our supplementary material.

In our implementation, the iteration is executed once during each frame, and the resultant distribution of the previous frame is used as the initial value for the current frame. This is equivalent to the repetitive iteration when the sound environment is fully static. In dynamic scenes, the distribution of the previous frame is used as a good initial guess.

The iterative algorithm described above requires an estimation of σ^2_{mn}. This estimation must be reevaluated constantly to address the changes of the sound environment. However, the variance estimation can be inaccurate due to the computation budget or low sample quality, which can lead to audible defects. In order to ameliorate

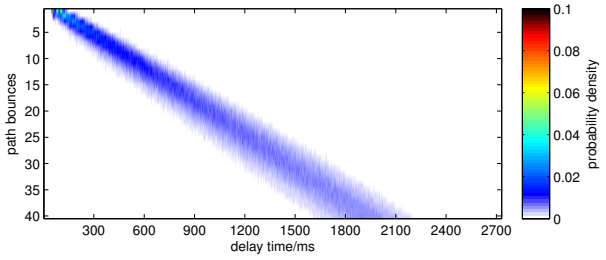


Figure 4: *Relationship between path bounces and the temporal distribution of samples. The data is generated by our BST renderer in a cube room scene. The correlation between time and path bounces is evident. We could alter the temporal sample distribution by controlling the number of samples for each bounce.*

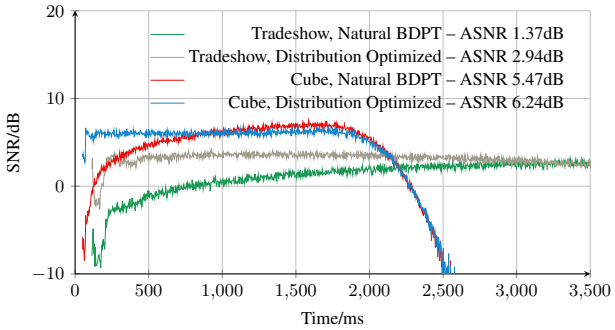


Figure 5: *The effect of optimized sample distribution reflected by SNR curves in different sound environments. With our optimized sample distribution, the quality of early and late responses is better balanced, resulting in improved overall quality, measured in average SNR (ASNR).*

the estimation quality, we exploit the temporal coherence and combine the estimation of the current frame with the results from the previous frame.

To control the quality of variance estimation, we tag every variance estimation σ^2 with a quality indicator Q, which could be regarded as equivalent to the number of samples used for estimation. In each frame, we achieve a new variance estimation σ_0^2 from Q_0 samples, then combine σ_0^2 with the estimation of the last frame σ_{i-1}^2 to evaluate the estimation of the current frame σ_i^2. To balance between the quality and the responsiveness to scene changes, we designed an update strategy of σ^2 and Q:

$$\sigma_i^2 = \begin{cases} \sigma_0^2, & Q_0 > Q^* \\ \dfrac{Q_{i-1}\sigma_i^2 + Q_0\sigma_0^2}{Q_{i-1}+Q_0}, & Q_{i-1}+Q_0 < Q^* \\ \gamma\sigma_i^2 + (1-\gamma)\sigma_0^2, & \text{otherwise.} \end{cases} \tag{18}$$

$$Q_i = \begin{cases} Q_0, & Q_0 > Q^* \\ Q_0 + Q_{i-1}, & Q_{i-1}+Q_0 < Q^* \\ Q^*, & \text{otherwise.} \end{cases} \tag{19}$$

where Q^* is a predefined quality standard, and γ is given as

$$\gamma = \frac{Q_{i-1} - \sqrt{Q_{i-1}Q_0(\frac{Q_{i-1}+Q_0}{Q^*} - 1)}}{Q_{i-1}+Q_0}. \tag{20}$$

A detailed discussion of the update strategy can be found in our supplementary material.

Algorithm Summary The outline of our bidirectional sound propagation algorithm is presented in Algorithm 1.

Sec. 3.1 covers most of the algorithm above, including the evaluation of $w(\cdot)$ and the sample allocation among estimators, as in line 6. The main difference between our algorithm and the original BDPT is the Optimize Step (line 18, 19) and the distribution of samples among integrals (line 4). In the Optimize Step, the resulting temporal distribution of samples in the current frame is used to evaluate the sample distribution among integrals of the next frame, while at the beginning of the processing of every frame, the samples are distributed among integrals according to the distribution evaluated in the last frame.

Algorithm 1 Bidirectional Sound Transport

Input:
Sound environment. subpath budget M, sample budget N, Maximum path bounce K.
Initialize:
Sample probability x_n of bounce n as $1/K$,
variance estimation $\sigma_{m,n}^2$ as 0.
Output: Energy response $E(\mathbf{x}, \omega, t)$ of each frame.

1: **for** each frame **do**
2:
3: *// Preparation*
4: Allocate samples among bounces according to x_n;
5: **for** $i = 0$ to K **do**
6: Allocate samples among different estimators of $T^i L_0$;
7: **end for**
8:
9: *// Trace step*
10: Trace new subpaths;
11:
12: *// Connect step*
13: Connect between the subpaths to generate samples;
14: Evaluate $f(\cdot)$, $p(\cdot)$ and MIS weight $w(\cdot)$ for each sample;
15: Evaluate $\mathrm{ER}(\mathbf{x}, \omega, t)$ according to Eq. (7) and Eq. (12);
16:
17: *// Optimize step*
18: Update $\sigma_{m,n}^2$ according to Eq. (18);
19: Update x_n according to Eq. (17).
20: **end for**

4 Interactive Sound Propagation

In this section, we show how to combine our BST algorithm with caching schemes that exploit the temporal coherence of sound propagation to reduce the overall propagation computation. Note that techniques mentioned in this section will introduce bias.

4.1 Diffuse Cache

Diffuse cache is a technique proposed by [Schissler et al. 2014] that exploits the temporal coherence of sound propagation to improve the render quality. Inspired by the (ir)radiance cache [Ward et al. 1988; Křivánek et al. 2005] used in visual rendering, diffuse cache maintains a moving average of certain intermediate variables used in sound propagation to quickly update the contributions. Diffuse cache gives the user a strong impression of temporal consistency, which might otherwise require a huge number of samples to achieve.

In the original version of diffuse cache, the entries are stored in a hash table, and the query index is a list of subdivided face lists of scene geometry. However, this cache structure is not practical for our BST renderer, which generates a large number of samples during every frame. We have found that the probability of having two valid paths that can hit the same face list is extremely small, and the most common operation in the diffuse cache algorithm is the traversal of a large hash table, which can be time consuming.

In our formulation, the diffuse cache of a single frame consists of a table of path node information and a queue of cache entries. The information of a path node includes its position, normal and material index. Every valid path of more than one bounce will add an entry to the queue. For a valid path $\mathbf{x}_0 \cdots \mathbf{x}_i$, the cache entry stores the value $f(\mathbf{x}_1 \cdots \mathbf{x}_{i-1})/p(\mathbf{x}_0 \cdots \mathbf{x}_i)$, the index of node $\mathbf{x}_1$ and $\mathbf{x}_{i-1}$, and the direction of $\mathbf{x}_2 \to \mathbf{x}_1$ and $\mathbf{x}_{i-2} \to \mathbf{x}_{i-1}$ (if valid). In a static scene, this information is sufficient to evaluate

$f(\mathbf{x}_0 \cdots \mathbf{x}_i)/p(\mathbf{x}_0 \cdots \mathbf{x}_i)$ for any $\mathbf{x}_0$ and $\mathbf{x}_i$. The only information that requires recomputation is the geometry term $G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_i)$ and $G(\mathbf{x}_{i-2} \leftrightarrow \mathbf{x}_{i-1})$. Since the geometry term contains visibility information, an accurate reevaluation would be computationally expensive. To reduce the computation cost, we simplify the visiblity test to a test of normal orientation, which still eliminates a portion of invalid segments.

4.2 Path Cache

Subpath generation is one of the most computation-intensive tasks within a BDPT-based tracer. In fact, generating a new node on a path is always more costlier than connecting two nodes. In scenes with static objects, one could reuse the subpaths emitted from fixed sound sources and listeners. For example, when the listener is moving and the source is static, we could cache the subpaths emitted by the source, or vice versa. This only works with BDPT-based algorithms as they generate subpaths from both sides. Since the cached paths cannot represent all the information of the sound environment, the resulting energy estimation will be biased.

In our implementation, the information of the generated subpaths is stored and categorized by the emitter of the subpaths. Before the trace step of the next frame, we check the status of each path emitter to determine the validity of these cached subpaths. For a certain emitter, if we have not detected any change that threatens the validity of its cached subpaths, then the emitter is regarded as "static." It will not generate new paths in this frame, but reuse the cached subpath information instead.

5 Results and Analysis

We have conducted a series of experiments in different sound environments to test the performance of BST and to compare it with other existent sound propagation algorithms, especially backward PT with diffuse rain. All of the results are produced by a computer with an Intel i7 3.50Ghz CPU and 32GB memory.

Both the implementations of BST and the reference algorithm (backward PT with DR, denoted as PT+DR in the following part of the paper) execute on a single thread. We use SIMD instruction sets to accelerate various parts of our implementations, including ray intersection (with the help of an Embree raytracing engine [Wald et al. 2014]).

To make the resulting data comprehensible, we present a brief description of the benchmarks used in our experiments.

(a) (b) (c) (d) (e)

Figure 6: *Sound environments used in our experiments: (a) Roomset, 19266 triangles; (b) Elmia Round Robin, 1047 triangles; (c) Crytek Sponza, 279163 triangles; (d) Sibenik Cathedral, 75155 triangles; (e) Tradeshow, 177405 triangles.*

Roomset An indoor scene with two floors, stairs, several small rooms, and complex occlusion. The mean free path of this scene is much shorter than the other scenes. Generating valid paths in this type of scene is a very challenging task for sound simulators due to severe occlusion.

Elmia Round Robin Model of a real concert hall with detailed description of sound materials. A typical scene designed for offline simulation. This model was originally used for the evaluation of professional room acoustic software [Bork 2000].

Crytek Sponza A half-indoor scene created by Frank Meinl of Crytek with detailed geometric objects and moderately complex occlusion. This benchmark has been used by prior sound propagation algorithms.

Sibenik Cathedral Indoor scene created by Marko Dabrovic with little occlusion. This scene has a long reverberation time and the quality of the late response is very important.

Tradeshow Huge indoor scene, the interior space of which is approximately $2.6\times10^5\mathrm{m}^3$. The sound sources, the listener, and most of the occluders are located close to the floor of the scene geometry. We know from the previous discussion that this scene is strongly unfavorable to PT with diffuse rain.

5.1 Single Source Benchmarks

A key issue is to evaluate the rendering quality of the auralized sound. In Sec. 3.2, we introduced our new quality criteria. However, an energy response can only represent the sound propagation between a single source-listener pair, and does not extend directly to multi-source environments. As a result, the quality assessment is carried out in single-source scenes only.

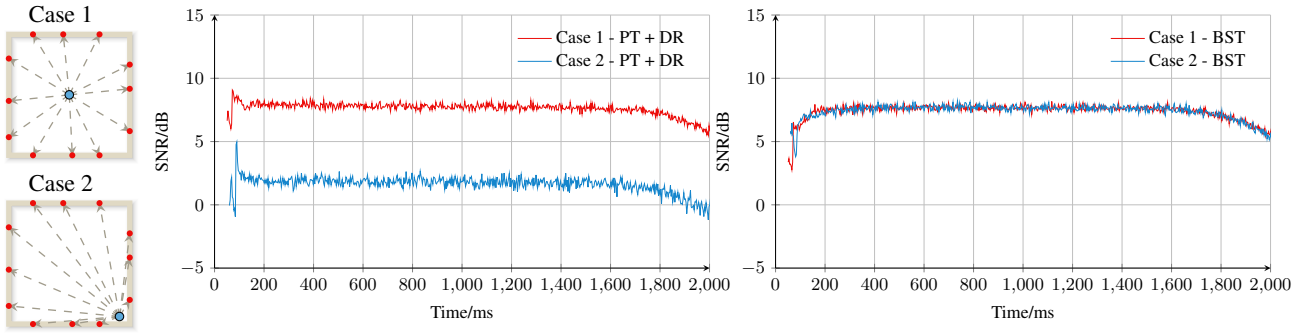
In Fig. 7 we compare the rendering quality of BST and backward PT+DR under comparable tracing performance. As shown, BST achieves higher rendering quality (in terms of the SNR metric) than PT+DR. Moreover, in scenes containing moving sources, BST produces more stable results than PT+DR. For PT+DR, we observe an obvious drop in the SNR when the source moves close to scene objects. This is because the final connections of the source and the hit points on the path vary significantly in length when the source is located near scene objects. As can be seen from Eq. (11), the connection segment can have a major influence on the estimator, and therefore large variance in the length of the final connection often leads to large variance in the geometry term in the nominator of Eq. (11) and hence the estimator. In contrast, BST produces similar SNR curves regardless of the source location, which indicates that we will have stable rendering quality when the source is moving around the scene. This benefit of our algorithm makes it much easier to determine the ray budget in interactive applications.

In Fig. 8 we compare the performance of BST with PT+DR under comparable quality. In this experiment, the parameters of our method are fixed to 100 subpaths per frame and 4 connections per path node. The parameter for the reference PT+DR implementation is tweaked for each scene to match the output quality of our method. The output results of 200 consequent frames are collected for evaluating the data shown in Fig. 8. Note that the comparison is for the sound propagation algorithms only and neither the path cache nor the diffuse cache is used for both methods.

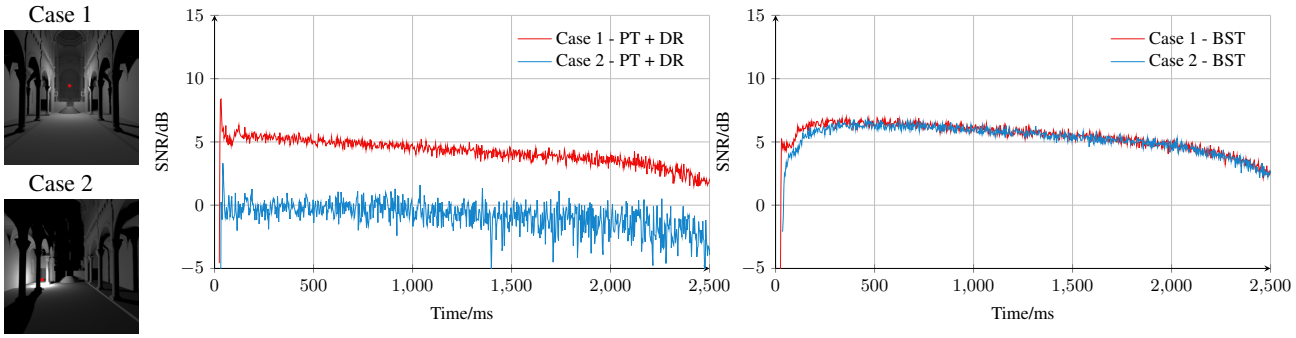
The result shows that our new algorithm outperforms PT+DR in all 5 scenes with a 2.5-58$\times$ speedup. The advantage of BST is related to the scene geometry and most obviously shown in the Tradeshow scene, in which the distance between the source and the floor is small compared with the scale of the scene.

5.2 Multiple Sources

The speed advantage of BST above does not naturally extend to multi-source scenes. In PT+DR, the computation cost of the trace step scales sub-linearly with the source number, while BST needs to compute the forward subpaths for every sound source. Since forward subpath generation counts for a large proportion of the overall



(a) Cube (12m × 12m × 12m)



(b) Sibenik Cathedral

Figure 7: *The SNR curves of energy responses rendered in two scenes. For the same algorithm, different cases in the same scene use the same number of samples. Different algorithms use different numbers of samples to achieve comparable tracing performance. In the cube scene, the sound source is located at the center of the cube in case 1, and near a corner-edge in case 2. In the Sibenik cathedral scene, the source is represented as red dots in pictures above, and the geometry term for connection is visualized in grayscale. As shown, in scenes containing moving sources, our BST produces more stable results than backward PT+DR. Moreover, our BST achieves higher rendering quality than PT+DR, under comparable tracing performance.*

propagation cost, as can be seen from Fig. 8, the time cost of BST scales almost linearly with the source number. On the other hand, source clustering is frequently used in practice to control the overall source number [Tsingos et al. 2004]. Moreover, in scenes with a limited number of moving sources, the scalability of our algorithm can be improved using path caching.

In our benchmarks, a fraction of the sound sources are fixed sound sources and their positions are randomly chosen in the Tradeshow scene and the Roomset scene, which represent the best and worst case for BST. The computation budget for each sound source is the same as in the single-source experiment (the number of subpaths is fixed for backward PT with DR), which makes the render quality similar to the results in Fig. 8 for every source. The performance of different algorithms is shown in Fig. 9.

5.3 Validation

To verify the reliability of our algorithm, we have compared our results with the commercial room acoustic simulation software ODEON, which uses a GA-based algorithm that combines the image-source method and path tracing. The simulation accuracy of ODEON is validated by measured results. In fact, the IR generated by ODEON in the Elmia Round Robin scene has a very good correspondence with real-world measurements [Bork 2000]. Fig. 11 shows the comparison of energy curves generated by our algorithm and ODEON in the Elmia Round Robin scene. We have compared the energy curves of 6 octave frequency bands with the central frequency between 250Hz and 8000Hz (since GA methods are inaccurate in low frequency bands, we avoided comparison in extremely low frequencies). These results match well for all bands, and the average difference of energy in each band is between 1.32-2.33dB.

The unbiasedness of our algorithm is verified by our experiment in a similar way to [Qin et al. 2015]. The validation is done by comparing the average energy response of independent runs with a ground truth calculated in a single run with a large number of samples, and computing the RMS error. The result is shown in Fig. 10. One can see that the error agrees with the square root convergence rule of an unbiased Monte-Carlo estimator, which is a typical behavior of an unbiased algorithm.

6 Conclusions, Limitations, and Future Work

We present a novel energy-based quality metric (SNR) and a sound propagation algorithm based on BDPT. Our metric can be used for formal evaluation of ray-tracing sound propagation algorithms and we use this metric to characterize the performance of different algorithms. Our algorithm can offer considerable speedup over prior sound propagation algorithms based on forward or backward ray tracing. We highlight its benefit on different benchmarks.

Our approach has some limitations. Our algorithm inherits part of the limitations of GA methods: inaccuracy at low frequency and inability to simulate all wave-based sound effects. Based on BDPT, our method also has difficulty in processing materials with ideal specular reflection, which are currently approximated with narrow BSDFs. Our method for adjusting the temporal sample distribution suffers from its low precision. Further, many features of the resulting energy response, like the smoothness, cannot be controlled by our approach. Finally, the scalability with the number of sound

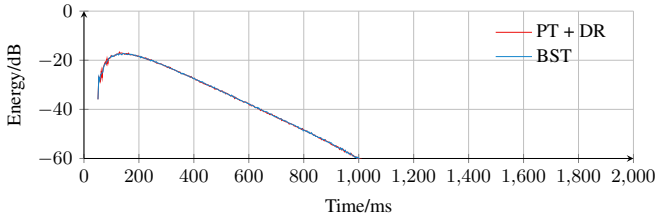
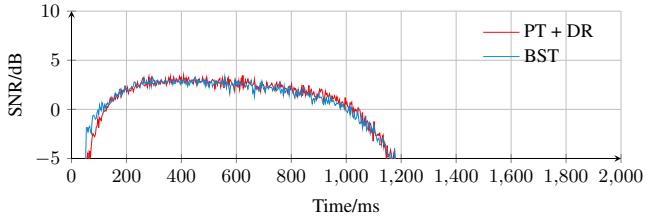
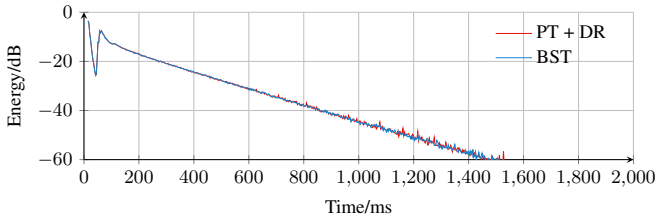
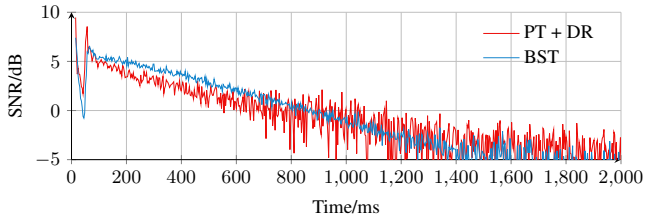
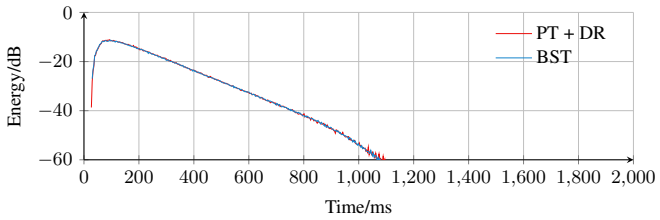
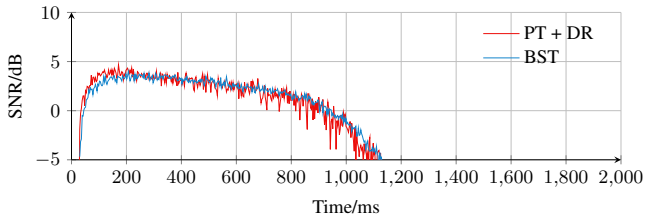
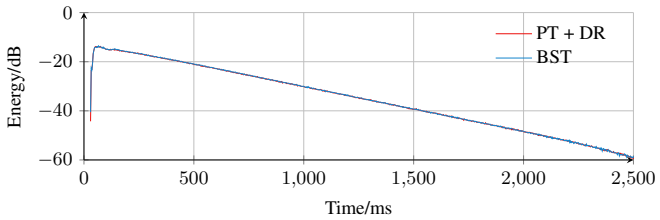
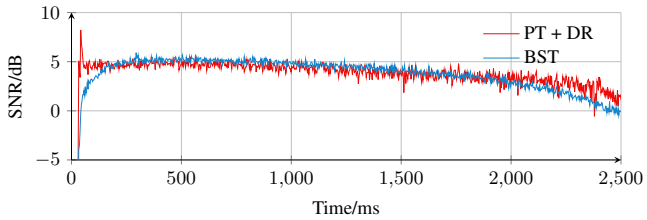
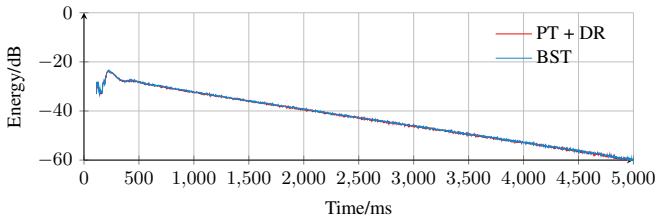
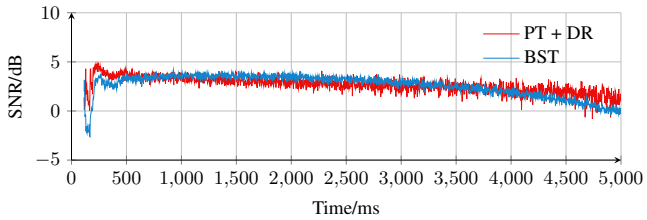
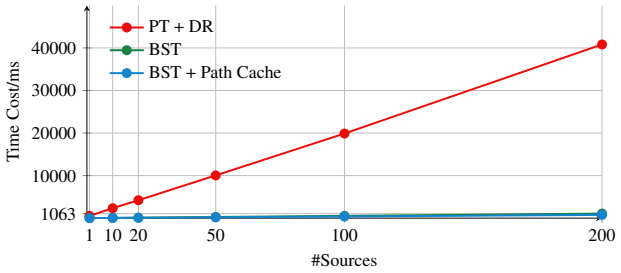
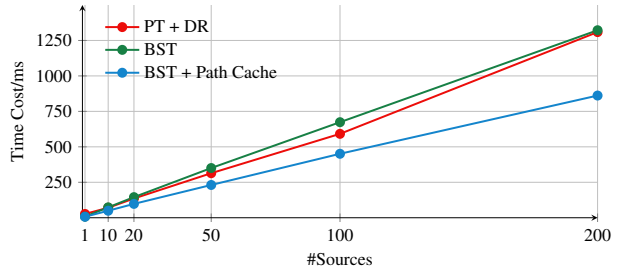
Scene Complexity			Computation Budget		Time Cost (ms)					
#Tris	#Bounces	Method	#Subpaths	#Connections	Trace	Connect	Optimize	Total	ASNR (dB)	Range (ms)
Roomset, 22m×11m×6m										
										
19266	120	PT+DR	620	74400	30.8	4.46		35.3	1.92	54-990
		BST	100	48000	4.76	7.71	1.31	13.9	1.93	
Elmia Round Robin, 42m×39m×16m										
										
1047	100	PT+DR	1900	190000	62.1	41.1		103	-0.69	18-1917
		BST	100	40000	3.50	12.9	1.81	18.3	-0.65	
Crytek Sponza, 23m×37m×15m										
										
279163	80	PT+DR	2250	180000	90.3	41.7		132	2.36	33-972
		BST	100	32000	4.24	8.84	1.17	14.3	2.37	
Sibenik Cathedral, 19m×46m×35m										
										
75155	90	PT+DR	6500	585000	241	109		349	4.24	33-2193
		BST	100	36000	4.63	11.6	1.18	17.4	4.29	
Tradeshow, 105m×84m×29m										
										
177405	70	PT+DR	28000	1960000	476	240		716	2.88	117-4392
		BST	100	28000	2.25	8.97	1.11	12.4	2.94	

Figure 8: *Comparison of different propagation algorithms in single-source scenes. The reported data is the average of all 200 output frames. Values in column ASNR is the averaged SNR in the interval given in the Range column. The total time cost in this table includes the cost of the "trace" step, the "connect" step, sample distribution optimization, and various trivial operations. One can see from the time cost that BST outperforms backward PT+DR in all 5 benchmarks.*



(a) Tradeshow



(b) Roomset

Figure 9: *Comparison of different propagation algorithms in multi-source scenes of comparable quality. BST performance scales worse than backward PT with DR in the Roomset scene due to its inefficiency in handling a large number of sources and the absence of scene features that favor BST.*

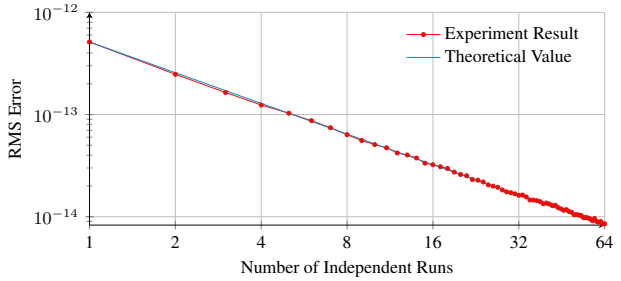


Figure 10: *Bias experiment. The RMS error of our algorithm in the Tradeshow scene is compared with the theoretical error estimation calculated with the convergence rule of an unbiased Monte-Carlo estimator. The experiment result matches with the theoretical estimation very well.*

sources of our method is worse than that of backward PT.

To better understand the relationship between the energy response SNR and auditory perception, we plan to perform further evaluation of the energy-based SNR metric and evaluate our algorithm in more complex scenarios. In our benchmarks, most of the output energy responses can be described with a few representative characteristics, especially for the late response, which is well-described by its decay rate. One could use these characteristics to "guess" the missing part of the energy response and reduce the overall computation cost of our algorithm. We would also like to study the path reuse techniques in sound propagation under the BST framework.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. The authors from Zhejiang University are partially supported by the National Key Research & Development Plan of China (No. 2016YFB1001403), the NSF of China (No. 61272305, No. 61379070), and the National Program for Special Support of Eminent Professionals of China. This research is also supported by US National Science Foundation and the National Thousand Talents Program of China.

References

ANTANI, L., AND MANOCHA, D. 2013. Aural proxies and directionally-varying reverberation for interactive sound propagation in virtual environments. *Visualization and Computer Graphics, IEEE Transactions on 19*, 4, 567–575.

ANTANI, L., CHANDAK, A., SAVIOJA, L., AND MANOCHA, D. 2012. Interactive sound propagation using compact acoustic transfer operators. *ACM Transactions on Graphics (TOG) 31*, 1, 7.

BERTSEKAS, D., AND NEDIC, A. 2003. Convex analysis and optimization (conservative).

BLAUERT, J. 1997. *Spatial hearing: the psychophysics of human sound localization*. MIT press.

BORK, I. 2000. A comparison of room simulation software-the 2nd round robin on room acoustical computer simulation. *Acta Acustica united with Acustica 86*, 6, 943–956.

BREBBIA, C. A., AND CISKOWSKI, R. 1991. *Boundary element methods in acoustics*. Computational mechanics.

DACHSBACHER, C., KŘIVÁNEK, J., HAŠAN, M., ARBREE, A., WALTER, B., AND NOVÁK, J. 2014. Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1, 88–104.

DEVALOIS, R. L. 1988. *Spatial vision*. No. 14. Oxford University Press, USA.

KELLER, A. 1997. Instant radiosity. In *Proceedings of SIGGRAPH'97*, ACM Press/Addison-Wesley Publishing Co., 49–56.

KREYSZIG, E. 1978. *Introductory Functional Analysis with Applications*. Society for Industrial and Applied Mathematics.

KŘIVÁNEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination computation. *Visualization and Computer Graphics, IEEE Transactions on 11*, 5, 550–561.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In *Proceedings of Compugraphics*, vol. 93, 145–153.

LAFORTUNE, E. 1996. Mathematical models and monte carlo algorithms for physically based rendering. *Department of Computer Science, Faculty of Engineering, Katholieke Universiteit Leuven*, 20–23.

LENTZ, T., SCHRÖDER, D., VORLÄNDER, M., AND ASSENMACHER, I. 2007. Virtual reality system with integrated sound field simulation and reproduction. *EURASIP journal on applied signal processing 2007*, 1, 187–187.

LI, D., FEI, Y., AND ZHENG, C. 2015. Interactive acoustic transfer approximation for modal sound. *ACM Transactions on Graphics (TOG) 35*, 1, 2.

MAHNEKE, A. 1957. Flicker-fusion thresholds. *Acta ophthalmologica 35*, 1, 53–61.

MEHRA, R., RAGHUVANSHI, N., ANTANI, L., CHANDAK, A., CURTIS, S., AND MANOCHA, D. 2013. Wave-based sound

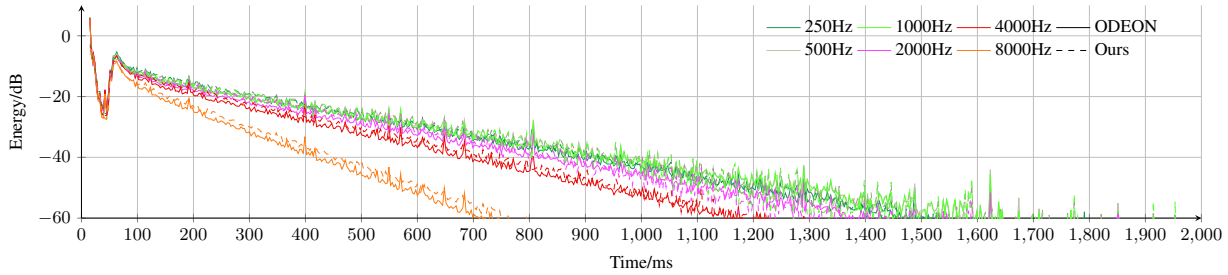


Figure 11: *Energy curves generated by ODEON and our method in 6 octave bands with central frequencies between 250-8000Hz. The computation cost is 1000 subpaths, 100 bounces, and 4 connections per path nodes for our algorithm, and 40000 rays for ODEON. Our results closely match that of ODEON in all bands.*

propagation in large open scenes using an equivalent source formulation. *ACM Transactions on Graphics (TOG) 32*, 2, 19.

NICOL, R., GROS, L., COLOMES, C., NOISTERNIG, M., WARUSFEL, O., BAHU, H., KATZ, B. F., AND SIMON, L. S. 2014. A roadmap for assessing the quality of experience of 3d audio binaural rendering. In *Proc. of the EAA Joint Symposium on Auralization and Ambisonics*.

PAJOT, A., BARTHE, L., PAULIN, M., AND POULIN, P. 2011. Combinatorial bidirectional path-tracing for efficient hybrid cpu/gpu rendering. *Computer Graphics Forum 30*, 2, 315–324.

PELZER, S., AND VORLÄNDER, M. 2010. Frequency-and time-dependent geometry for real-time auralizations. In *Proceedings 20th International Congress on Acoustics (ICA 2010), Sydney, Australia*, 23–27.

PELZER, S., ARETZ, M., AND VORLÄNDER, M. 2011. Quality assessment of room acoustic simulation tools by comparing binaural measurements and simulations in an optimized test scenario. In *Proc. Forum Acusticum Aalborg*.

POPOV, S., RAMAMOORTHI, R., DURAND, F., AND DRETTAKIS, G. 2015. Probabilistic connections for bidirectional path tracing. *Computer Graphics Forum 34*, 4, 12.

QIN, H., SUN, X., HOU, Q., GUO, B., AND ZHOU, K. 2015. Unbiased photon gathering for light transport simulation. *Acm Transactions on Graphics 34*, 6, 1–14.

RAGHUVANSHI, N., AND SNYDER, J. 2014. Parametric wave field coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG) 33*, 4, 38.

RAGHUVANSHI, N., SNYDER, J., MEHRA, R., LIN, M., AND GOVINDARAJU, N. 2010. Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. In *ACM Transactions on Graphics (TOG)*, vol. 29, ACM, 68.

ROYDEN, H. L., AND FITZPATRICK, P. 1988. *Real analysis*, vol. 198. Macmillan New York.

RYCHTÁRIKOVÁ, M., VAN DEN BOGAERT, T., VERMEIR, G., AND WOUTERS, J. 2011. Perceptual validation of virtual room acoustics: Sound localisation and speech understanding. *Applied Acoustics 72*, 4, 196–204.

SAVIOJA, L., AND SVENSSON, U. P. 2015. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America 138*, 2, 708–730.

SAVIOJA, L. 2010. Real-time 3d finite-difference time-domain simulation of low-and mid-frequency room acoustics. In *13th Int. Conf on Digital Audio Effects*, vol. 1, 75.

SCHISSLER, C., AND MANOCHA, D. 2014. Interactive sound propagation and rendering for large multi-source scenes. Tech. rep., Department of Computer Science, University of North Carolina. Submitted for publication.

SCHISSLER, C., MEHRA, R., AND MANOCHA, D. 2014. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (TOG) 33*, 4, 39.

SCHRÖDER, D. 2011. *Physically based real-time auralization of interactive virtual environments*, vol. 11. Logos Verlag Berlin GmbH.

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

TAYLOR, M., CHANDAK, A., MO, Q., LAUTERBACH, C., SCHISSLER, C., AND MANOCHA, D. 2012. Guided multi-view ray tracing for fast auralization. *Visualization and Computer Graphics, IEEE Transactions on 18*, 11, 1797–1810.

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

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

TSINGOS, N., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 249–258.

VEACH, E., AND GUIBAS, L. 1995. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 145–167.

VORLÄNDER, V. M. 1988. Die genauigkeit von berechnungen mit dem raumakustischen schallteilchenmodell und ihre abhängigkeit von der rechenzeit. *Acta Acustica united with Acustica 66*, 2, 90–96.

WALD, I., WOOP, S., BENTHIN, C., JOHNSON, G. S., AND ERNST, M. 2014. Embree: A kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG) 33*, 4, 143.

WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *ACM SIGGRAPH Computer Graphics*, vol. 22, ACM, 85–92.