Adaptive Impulse Response Modeling for Interactive Sound Propagation

Carl Schissler* Dinesh Manocha†
University of North Carolina at Chapel Hill

Figure 1: *Our sound propagation technique is 5x faster and uses 100x less memory than the previous state of the art. It can produce plausible acoustic effects for complex interactive scenes: (left) Space Station, 21 sources; (center) Office, 24 sources; (right) Hangar, 18 sources.*

Abstract

We present novel techniques to accelerate the computation of impulse responses for interactive sound rendering. Our formulation is based on geometric acoustic algorithms that use ray tracing to compute the propagation paths from each source to the listener in large, dynamic scenes. In order to accelerate generation of realistic acoustic effects in multi-source scenes, we introduce two novel concepts: the impulse response cache and an adaptive frequency-driven ray tracing algorithm that exploits psychoacoustic characteristics of the impulse response length. As compared to prior approaches, we trace relatively fewer rays while maintaining high simulation fidelity for real-time applications. Furthermore, our approach can handle highly reverberant scenes and high-dynamic-range sources. We demonstrate its application in many scenarios and have observed a 5x speedup in computation time and about two orders of magnitude reduction in memory overhead compared to previous approaches. We also present the results of a preliminary user evaluation of our approach.

Keywords: sound propagation; impulse response; temporal coherence, psychoacoustic

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

1 Introduction

The problem of generating high-fidelity sound effects is important for many interactive applications, including games, virtual reality, and human-computer interaction systems. It is well-known that realistic sounds can improve a user's sense of presence and immersion [Begault et al. 2001]. Moreover, recent availability of commodity head-mounted displays and augmented reality devices has renewed interest in sound simulation and rendering.

In this paper, we address the problem of interactive sound propagation in complex indoor and outdoor environments with a large number of sources. The primary goal is to simulate how sound waves in a virtual scene interact with the environment and are heard by a listener. This includes modeling of various acoustic effects including reflections, diffraction, Doppler shift, echoes, etc. Another major issue is simulating reverberation, the persistence of sound that occurs due to the build up of a large number of reflections that then decay over time due to absorption. Current algorithms for interactive sound propagation in dynamic scenes are based on geometric sound propagation. These methods compute the propagation paths from each source to the listener using ray tracing, beam tracing, or frustum tracing [Vorländer 1989; Funkhouser et al. 1998; Taylor et al. 2009; Schissler et al. 2014]. The result of sound propagation is a one-dimensional filter called an *impulse response* (IR) that specifies the linear transfer function between a source and listener pair [Kuttruff 2007]. The final audio at the listener position is generated by convolving the IR with the unprocessed source sound.

The runtime complexity of geometric propagation algorithms is generally dominated by the computation of IRs using ray tracing. The runtime complexity is a linear function of the number of rays being traced for each acoustic effect, and it is important to minimize the number of rays in order to achieve interactive performance. A recent method exploits temporal coherence of the sound field and reduces the number of rays required to achieve high-quality sound [Schissler et al. 2014]. However, it can require over 100MB per source to store the propagation paths for high-order reflections.

Another significant issue for sound propagation algorithms is to determine the length of the impulse response *a priori*. The length of the IR specifies how long that sound should travel at the speed of sound. If this parameter is longer than the actual response, ray tracing computation is wasted on sound that is inaudible to a human listener. In highly reverberant environments (e.g. parking garage, cathedral), the IR will be truncated if it does not capture the entire sound decay. The loudness of the sound source is also an important factor, since louder sources tend to have longer audible reverberation. In practice, the length of the IR needs to be tuned for each acoustic space and sound source that the listener is likely to encounter. As a result, choosing the appropriate IR length is a major issue in generation of realistic sound effects at interactive rates.

*schissle@cs.unc.edu
†dm@cs.unc.edu
Project website

I3D '16,, March 01 2016, Redmond, WA, USA
ISBN: 978-1-4503-4043-4/16/03
DOI: http://dx.doi.org/10.1145/2856400.2856414

Main results: In this paper, we present algorithms for adaptive impulse response modeling that can significantly improve the performance and reduce the memory requirements for ray-tracing based interactive sound propagation. The two novel contributions include:

1. **Impulse Response Cache**: Our algorithm caches the IRs from prior frames and performs recursive filtering in order to take advantage of temporal coherence and trace fewer rays during each frame while maintaining high-quality sound.

2. **Adaptive Impulse Response Length**: We augment our ray-tracing algorithm to dynamically detect the current impulse response length for each sound source based on a psychoacoustic metric and the source's power.

These techniques have been implemented on a 4-core consumer desktop machine and evaluated using a variety of complex dynamic indoor and outdoor scenes with dozens of sources. We demonstrate a 5x savings in computation time as well as almost 2 orders-of-magnitude decrease in memory requirements as compared to previous interactive methods. We have also performed a preliminary user evaluation that suggests our results are comparable to a ground-truth offline simulation with 100x as many rays. In practice, our approach can generate realistic acoustic effects at interactive rates for a wide variety of complex indoor and outdoor scenes with dynamically-varying reverberation.

2 Related Work

The most accurate algorithms for sound propagation are based on directly solving the wave-equation using numeric methods. The recent trend is to use precomputation techniques to compute the sound pressure field for a static scene and perform runtime computations to evaluate the IRs at the listener position [Tsingos et al. 2007; Raghuvanshi et al. 2010; Mehra et al. 2013]. However, the computational complexity of these methods increases as a fourth power of frequency and as a linear function of the surface area or the volume of the scene. As a result, they are limited to lower frequencies (e.g. less than 1KHz) and static scenes. Many compression techniques have been proposed to reduce the memory overhead [Raghuvanshi et al. 2010; Raghuvanshi and Snyder 2014].

Geometric propagation techniques are more suited to interactive applications due to their ability to handle large scenes and dynamic environments. These methods make the simplifying assumption that the wavelength of sound is much smaller than the size of the objects in the scene and tend to be accurate at higher frequencies. These include image source methods [Borish 1984], beam tracing [Funkhouser et al. 1998] and frustum tracing [Taylor et al. 2009] that can be used for specular reflections. Diffuse reflections can be computed using Monte Carlo path tracing [Vorländer 1989; Embrechts 2000; Taylor et al. 2009; Schissler et al. 2014] or the acoustic rendering equation [Siltanen et al. 2007]. Low frequency wave effects such as diffraction can be approximated using the uniform theory of diffraction (UTD) [Tsingos et al. 2001; Taylor et al. 2009], or the more accurate but slower Biot-Tolstoy-Medwin (BTM) formulation [Svensson et al. 1999]. Late reverberation can be computed using artificial reverberation, statistical methods and geometric acoustics [Valimaki et al. 2012]. Frequency-domain compression techniques have been used to reduce the storage required for precomputed filters [Tsingos 2009].

3 Overview

In this section, we describe the novel components of our interactive sound propagation algorithm. An overview of the system pipeline is shown in Figure 3. As input, the system takes a set of spherical sound sources, one or more listeners, and a representation of the scene, usually in the form of triangle meshes for each object with material properties specified per-triangle. The sound propagation module computes an impulse response for each source and listener pair during each frame. We use distinct algorithms to compute specular, diffraction, and diffuse reflections, and incorporate new techniques to improve the results of sound propagation through our adaptive impulse response modeling. The resulting responses are convolved with source audio and played for the user.

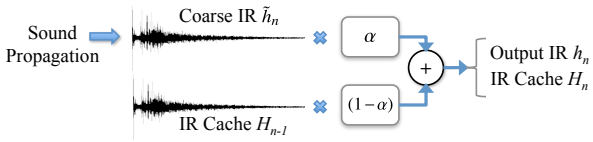


Figure 2: *This figure summarizes our IR cache algorithm. A persistent copy of the IR, H_{n-1}, is stored for each sound source. A small number of rays are used to compute a coarse impulse response for frame n, $\tilde{h}_n$, and this coarse IR is combined with the IR cache H_{n-1} at frame $n-1$ using the interpolation factor α. The resulting filtered IR is then stored in the IR cache for the next frame before it is used for sound rendering.*

3.1 Impulse Response Cache

One promising area of research in the field of interactive sound propagation is the use of persistent data structures that allow incremental computation of the sound field by exploiting temporal coherence in interactive applications. For many interactive applications, there is not much change in the sound field over time, and so these approaches take advantage of that property to reduce the computations per frame. This idea has been applied to diffuse sound paths [Schissler et al. 2014]. However, this approach does not work well for late reverberation (i.e. 50-100 orders of reflections). In this case, the number of paths that need to be stored can number in the millions per sound source. This results in both a significant performance and memory overhead that limits the interactivity of these systems on complex scenes.

To address these issues, we introduce the notion of the *impulse response cache*, H_{n-1}, a copy of the last impulse response that is used to filter the output of a path-tracing based sound propagation algorithm. The IR cache H_{n-1} stores the accumulated weighted sum of past impulse responses, and so uses multiple frames of path tracing to compute the resulting final IR for frame n, h_n. Our approach is general and can be applied to any sound propagation algorithm that uses a stochastic method such as Monte Carlo path tracing to compute an impulse response. During each frame, a different set of uniform random rays is traced, producing a slightly different impulse response. The quality of the impulse response and the computation time is dependent on the number of rays. The weighted sum of many of these impulse responses is a better estimate of the actual sound field than the IR computed only for that frame alone, since it contains the contributions of many more sound paths.

Our IR caching module takes as input an impulse response from path tracing, $\tilde{h}_n$, that contains the contributions from a small number of rays traced on the current frame. The module produces a filtered impulse response utilizing the history information stored in the IR cache H_{n-1}. This process is summarized in Figure 2. We also introduce a parameter $\alpha \in [0, 1]$ that controls how responsive the IR cache is to changes in the impulse responses. The ith

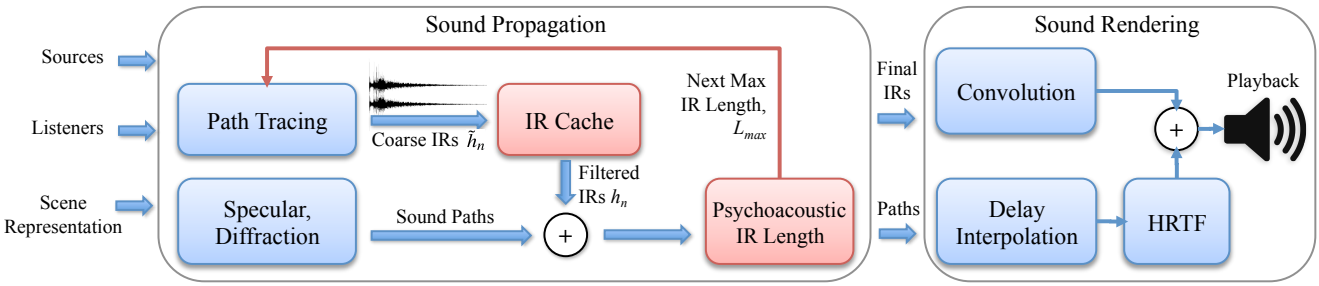


Figure 3: *An overview of our sound propagation and rendering pipeline; our novel components are highlighted in red. Given a set of sources, listeners, and the scene representation, our system computes diffraction, specular reflections, and diffuse reflections using path tracing. The IR cache filters the coarse output of path tracing, and the audible IR length is determined using a psychoacoustic metric. The final IRs and early reflection paths are rendered using streaming convolution and delay interpolation, respectively. Head-related transfer functions (HRTF) are applied to direct and early reflection paths for spatial sound. The resulting outputs are mixed and played through the audio device.*

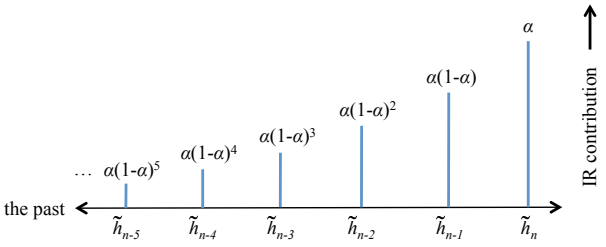


Figure 4: *The contribution of the impulse responses from previous frames decreases for frames that are further in the past. The coarse impulse response from the current frame, $\tilde{h}_n$, contributes with weight α, the last frame $\tilde{h}_{n-1}$ contributes with weight $\alpha(1-\alpha)$, and the jth previous frame with weight $\alpha(1-\alpha)^j$.*

impulse response sample on frame n is computed by the recursive relationship in equation 1.

$$h_n^i = H_n^i = \alpha \tilde{h}_n^i + (1-\alpha) H_{n-1}^i. \quad (1)$$

The final IR sample h_n^i is a linear combination of the current frame's path tracing output, $\tilde{h}_n^i$, and the contents of the IR cache for the sample, H_{n-1}^i. This is an application of the well-known temporal coherence technique called *exponential smoothing* [Brown 1956]. In graphics, exponential smoothing has previously been used to reduce the overhead of shading [Nehab et al. 2007], as well as to reduce spatial and temporal aliasing in shadow maps [Scherzer et al. 2007]. In essence, the IR cache applies a 1st-order recursive low-pass filter to each sample in the impulse response, thereby using the history stored in H_{n-1} to produce a higher-quality impulse response with fewer undersampling artifacts. The parameter α determines the weight of the current IR $\tilde{h}_n$ in the final output IR. A value of α close to 1 means that the system is more responsive to dynamic changes in the scene, but with less benefit from the IR cache. A value of α closer to 0 indicates that more weight is given to the IR cache history stored in H, and thus the simulation will benefit more from the cache but be less responsive. The contribution of past IRs to the current frame is illustrated by Figure 4.

Since it is unintuitive to determine the value of α directly, we propose the following method to compute α for a filtering window τ. This filtering window, given in seconds, is chosen to correspond to how long a coarse IR $\tilde{h}$ from a previous frame is considered to contribute to the final IR h_n. From the recurrence relation in equation 1, the contribution of coarse IR $\tilde{h}_{n-j}$ during the jth previous frame is $\alpha(1-\alpha)^j$. To compute a value of α, given τ and time step Δt, our goal is to compute α such that $\alpha(1-\alpha)^j \leq \epsilon$, where $j = \frac{\tau}{\Delta t}$. $\epsilon \in [0,1]$ corresponds to the weight where an IR is not considered contributing. In our simulations, we use $\epsilon = 0.01$. There is no closed-form expression for α, but standard root-finding techniques can be applied to find real $\alpha \in [0,1]$. Alternatively, the contribution of an IR can be approximated as $(1-\alpha)^j$ by dropping the initial α factor. This produces the following analytical solution in equation 2.

$$\alpha = 1 - \epsilon^{\Delta t/\tau}. \quad (2)$$

This expression enables an approximate value of α to be efficiently computed for a given time step and τ.

Variable Response Time: While a constant value of τ for every impulse response sample may give satisfactory results, we propose an extension that allows τ to vary for different parts of the IR. Previous work has shown that early reflections are perceptually important to a listener, especially in dynamic scenes. Therefore, they should be updated more often than later parts of the IR that are less useful for localization and that change more slowly [Begault et al. 2001; Müller-Tomfelde 2001]. Late reverberation can also take advantage of more temporal coherence by using a longer response time. As a result, we choose a smaller value of τ towards the beginning of the IR and a larger value towards the end. One possible approach is to define τ to be proportional to the delay time d. In our implementation, we compute τ using the relationship in equation 3.

$$\tau(d) = max(\beta d, \tau_{min}). \quad (3)$$

The scale factor β determines how quickly τ increases relative to the delay time. We use $\beta = 2$. The value of τ is clamped to always be greater than some minimum τ_{min}. From the value of $\tau(d)$ at each IR sample, the value of α is computed with equation 2, then used to update the IR cache with equation 1. This formulation enables the impulse response filtering to be applied in a perceptually relevant manner with a faster response time for the earlier reflections. It is also possible to design more complex schemes that can allow for τ to vary in arbitrary ways.

Frequency Bands: Another important aspect of sound propagation includes the generation of frequency-dependent sound effects. Like many prior geometric sound propagation systems, our approach computes the impulse response in discrete frequency bands. The output of ray tracing is a time-domain histogram of the sound energy for each band. Our technique operates directly on this representation and applies equation 1 to each band independently. When the convolution subsystem is updated, a pressure impulse response

suitable for audio rendering is computed by combining the information in all frequency bands [Schröder 2011].

Directional Sound: Directional and spatial sound effects are also important for sound localization and immersion [Barron 1974]. The IR cache can be extended to incorporate directional information for each sample using a *direction IR*. Our implementation uses one 3D Cartesian vector per IR sample. Other formats such as low-order spherical harmonics are also possible. The length of a vector in the direction IR indicates the strength of the directivity for that sample, and the direction of the vector is used for sound spatialization. Each ray that is accumulated in the IR during sound propagation adds its unit-length direction vector to the directional IR and is weighted by the ray's energy. To update the directional part of the IR cache, we apply equation 1 to the x, y, and z components of each vector independently. Because the directional information in the IR will vary at a slow rate controlled by τ, we store all directional information in the world coordinate frame. For sound rendering, we transform the direction for each IR sample into the listener's local coordinate space for sound spatialization. This allows the listener's spatial sound to be updated at a rate faster than τ allows.

3.2 Adaptive Impulse Response Length

A prominent feature of many geometric sound propagation algorithms is the need to choose a maximum order to which sound reflections are computed. The number of bounces required to capture a complete impulse response often varies widely for scenes of different sizes, shapes, and material properties. In addition, interactive simulations must consider the effects of dynamic objects in the scene that may alter the reverberation time, such as the opening or closing of a door. Outdoor scenes and coupled rooms can also be challenging for reverb time estimation [Carvalho 1995]. Many existing systems either specify a maximum distance or time that sound rays are propagated, or choose an arbitrarily high maximum reflection depth at which to truncate the response (e.g. 200). Other systems avoid computing certain inaudible paths based on time and spatial incidence [Begault 1996], but will not work for diffuse late reverberation, which corresponds to the sum of many individually inaudible paths. Due to the dependence of the reverberation time on a variety of factors, it can be hard to predict the IR length that should be computed for general scenes. To make matters worse, such a system can result in poor performance by tracing rays to higher reflection order than may be audible to a human listener.

We propose a novel approach that uses a psychoacoustic metric to dynamically optimize the ray propagation depth for interactive applications. We use information about the impulse response length from the previous frame $(n-1)$ to determine how far to propagate rays on the current frame (n). This feedback mechanism automatically adapts to changes in the response length, and thereby avoids parameter tuning. In order to determine the audible length of a given impulse response, we make use of the *absolute threshold of hearing*. The human threshold of hearing corresponds to the smallest sound pressure level that can be perceived by a human listener and is a well-studied topic in psychoacoustics literature [Fletcher 1940; Robinson and Dadson 1957]. The threshold, T_q, can be well-approximated over the audible frequency range for the average adult listener by equation 4 [Terhardt 1979].

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

This equation is used to compute the threshold in sound pressure level (dB SPL) for a frequency f in Hertz. A visualization of the threshold in Figure 5 shows that the threshold varies greatly over the audible frequency range. We use this formulation to compute

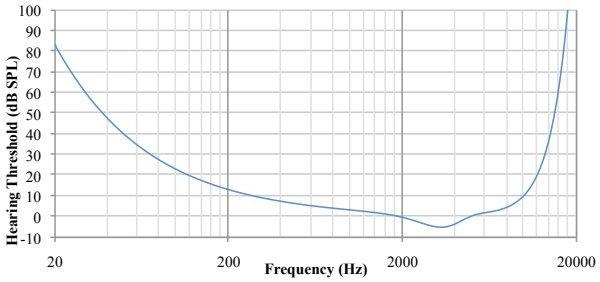


Figure 5: Psychoacoustic Metric: *The human threshold of hearing varies greatly with frequency and is well-approximated by equation 4. The highest sensitivity is in the* 2kHz − 3kHz *band, and the sensitivity decreases for very low and very high frequencies. Note that the horizontal axis is on a logarithmic scale.*

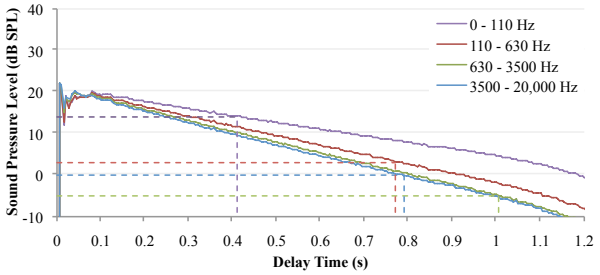


Figure 6: *An example impulse response that shows the audible IR length for 4 frequency bands. Horizontal dashed lines correspond to the threshold of hearing for each band, and vertical dashed lines correspond to the IR length per-band. In this case, the maximum IR length over all bands is* 1.01s*. On the next time step, rays will be propagated for up to* 1.01s + ΔL*, where* ΔL *is a parameter that determines how much the IR length can change per-frame.*

the threshold at runtime for a given simulation frequency band, and use the minimum value across each band as the threshold when determining the audible length of an IR.

As an input, our approach takes the full-length impulse response for each source consisting of an IR for each frequency band. Starting at the end of the IR for a given frequency band, we iterate over the IR samples in reverse and determine the last sample in the IR that is over the threshold of hearing. The delay time of this sample is reported as the perceptual IR length in seconds, L_p, for the frequency band. The results of this perceptual thresholding for an example IR are shown in Figure 6. In effect, there is no need to trace rays that are further than L_p time length, since those rays paths will be inaudible. To accomplish this, our algorithm stores this IR length information for each source so that it can be used to determine how far (in terms of propagation delay time) rays should be traced on the next frame. However, it is not sufficient to simply trace rays up to L_p, since this can cause the IR to artificially shrink in length over time, and it does not allow the IR to grow in length if the listener enters a more reverberant acoustic space. In order to address this problem, we always trace rays slightly past L_p. As rays are propagated through the scene, they are allowed to travel up to $L_{max} = L_p + \Delta L$ seconds, where ΔL is a parameter that limits how much the IR length can grow or shrink during a given frame.

The larger the value of ΔL, the quicker the algorithm can react to a change in the IR length. However, this is at the expense of tracing rays further past L_p. In our simulations, we use $\Delta L = 2\Delta t$, for time step Δt. Therefore, if $\Delta t = 100$ms, we always trace rays for

$L_{max} = L_p + 200$ms. This enables the IR length to grow or shrink by two seconds for every one second of real time, but still keeps extra ray propagation budget to a reasonable amount.

Once the value of L_{max} is computed, this information is stored and used to control the rays during the next frame for each source. When rays are propagated, no paths are computed that have a length greater than L_{max}. This results in significant savings for sound sources that are distant or quiet, and also allocates additional ray tracing for loud sources that require longer impulse responses. By computing an adaptive IR length, our approach also optimizes the ray tracing budget for scenes with different reverberation times.

4 Implementation

4.1 Sound Propagation

We compute sound propagation for a multiple-of-four discrete frequency bands (4, 8, 12, etc.). This enables efficient use of SIMD vector instructions on current CPUs. In our current implementation, we use these logarithmically-distributed frequency bands: $0-110$Hz, $110-630$Hz, $630-3500$Hz, and $3500-22050$Hz. The IRs are stored as arrays of floating-point samples. The frequency bands are interleaved so that the four bands for each sample are contiguous and can be efficiently loaded into vector registers. Our system uses different ray tracing approaches for computing specular, diffuse, and diffracted sound. Backwards path tracing from the listener is used to compute diffuse reflections [Schissler et al. 2014]. Vector-based scattering [Christensen and Koutsouris 2013] incorporates Lambertian scattering effects into our simulation. A scattering coefficient $s \in [0, 1]$, specified per-material, indicates the fraction of reflected sound that is scattered [Christensen and Rindel 2005]. We make use of *diffuse rain* sampling [Schröder 2011] to increase the number of sound paths found by generating an additional reflection to the sound source from each ray intersection point.

Specular early reflection paths are computed up to 5th order for our benchmarks. We use a ray-based variant of the image-source algorithm that handles area sound sources [Schissler and Manocha 2014]. For reflections up to order 5, we weight the strength of path tracing and specular reflection paths based on the scattering coefficient s. Above order 5, path tracing is used exclusively and rays are traced until the L_{max} threshold is reached. Diffraction paths are found up to 3rd order using the high-order UTD diffraction approach of [Schissler et al. 2014] that uses a precomputed edge visibility graph to accelerate path finding. To maintain interactive performance (e.g. 10-15 Hz), we automatically adjust the number of rays traced per frame so that the performance goal is met. Our system can also handle directional sound sources using a spherical-harmonic representation [Mehra et al. 2014] that multiplies each ray's energy by a scalar function of frequency and direction.

4.2 Sound Rendering

The input of the sound rendering module is an impulse response containing frequency band and directional information, a list of early reflection paths, and a stream of audio samples for each sound source. The early reflection and direct sound paths are rendered using fractional delay interpolation [Wenzel et al. 2000]. The source audio stream is filtered into discrete frequency bands using a 4th-order all-pass crossover and written to the delay buffer as interleaved samples similar to how IRs are stored in frequency bands. Each path linearly interpolates from the nearest delayed time samples and multiplies each frequency band by its corresponding gain coefficient to generate frequency-dependent effects. The final output sample value is the sum of all frequency bands. Accurate

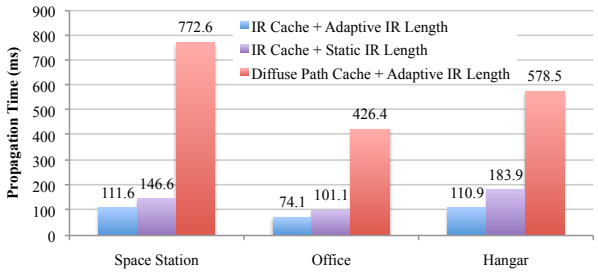


Figure 7: *When compared to a static IR length chosen by hand for each scene, our adaptive IR length is* $30-60\%$ *faster for these benchmarks. The diffuse path cache of [Schissler et al. 2014] incurs significant overhead when compared to our IR cache approach. The ray tracing time is the same for all techniques.*

Doppler shifting of these paths is modeled using the relative speed of the source and listener along each path to determine the delay interpolation rate. Our rendering system supports playback over both headphones or over a multichannel surround sound system. For headphones, a head-related transfer function (HRTF) is applied to direct sound and early reflections. We use a frequency-domain spherical harmonic representation of the HRTF for efficient filter construction and interpolation [Pollow et al. 2012].

To compute a pressure impulse response that can be used for frequency-domain convolution, we first filter each input IR band using a crossover filter to isolate its respective frequency band, then sum the band IRs to produce a monaural time-domain IR that incorporates all frequency-dependent effects. The directional information in the IR is then used to perform vector-based amplitude panning [Pulkki et al. 2001] for each IR sample to generate the final IR for convolution. In order to handle many concurrent convolutions, we use a non-uniform partitioned convolution algorithm similar to the one proposed in [Battenberg and Avizienis 2011]. To update to a new IR, each FFT partition in the impulse response is converted to frequency domain and updated using time-domain output interpolation [Müller-Tomfelde 2001]. The final sound is the sum of the output of the delay interpolation and convolution subsystems.

5 Results and Analysis

We evaluated our approach on complex dynamic indoor and outdoor scenes typical of those found in games and interactive applications (Figure 1). We achieve real-time performance for dozens of sound sources on a 4-core Intel i7 4770k desktop CPU. These results are summarized in Table 1.

Space Station: This scene is set on a space station in Earth orbit. There are 21 sound sources that include ventilation, computers, radios, and other ambient sounds. We show that our system can handle the effects of dynamic geometry (doors) on reverberation time. In this scene the IR length varies from 0.8s to 1.25s.

Office: The listener walks through a large office environment with 24 sound sources that include human voices, moving doors, running water, and an elevator. The IR length varies from 0.5s to 1.0s.

Hangar: This aircraft hangar with 18 sound sources demonstrates the ability of our approach to dynamically determine the IR length for different acoustic spaces and source power levels. As the listener moves into the open outdoor hangar, the reverb time increases for loud aircraft sound sources. Due to these loud sources and different environments, the IR length changes from 1.5s to 3.0s.

Scene	Scene Complexity		Propagation				Time (ms)		
	#Tris	#Sources	Avg. #Rays	Avg. #Bounces	IR Length (s)	Memory (MB)	Ray Tracing	IR Cache	Total
Space Station	35,581	21	588	136	0.8 - 1.25	40.8	105.0	3.42	**111.57**
Office	82,125	24	680	129	0.5 - 1.0	39.3	69.0	2.70	**74.08**
Hangar	71,461	18	1094	95	1.5 - 3.0	75.3	100.5	6.10	**110.91**

Table 1: *We highlight the primary results of our interactive sound propagation system. Our system is able to achieve interactive performance on complex scenes with dozens of sources. The IR cache uses only a few MB of memory per sound source, while the adaptive impulse response length ensures that only audible rays are traced when the IR length changes.*

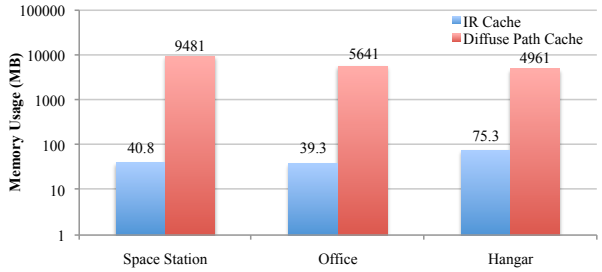


Figure 8: *We compare the memory usage of our IR cache approach to the diffuse path cache of [Schissler et al. 2014] that stores individual ray paths. Our technique uses roughly 2 orders of magnitude less memory to compute sound of similar fidelity. Note that the vertical axis has a logarithmic scale.*

5.1 Performance

On these complex scenes, our interactive sound propagation system is able to compute high-quality sound in $74\text{ms} - 111\text{ms}$. The use of the IR cache enables just $500 - 1000$ rays to be traced per frame. As a result, the time spent in ray tracing for our method is interactive for scenes with around 20 sources. The additional time required to update the cache using equation 1 is only a few milliseconds and scales linearly with the length of the IR.

We also compared the performance of our adaptive IR length approach relative to that for a static IR length. The static IR length was chosen by hand for each scene to be the longest audible IR length for the given scene. The lengths were 1.25s, 1.0s, and 3.0s for the Space Station, Office, and Hangar scenes respectively. Figure 7 shows that our adaptive approach provides a $30 - 60\%$ speedup over the static IR length. Since the audible IR length for a sound source (L_p) is often less than the maximum IR length for a scene, our method can reduce the computation required for quieter sources by tracing fewer ray reflections. The adaptive IR length can handle dynamically changing reverberation times and there is no longer the requirement to set the IR length parameter by hand. For any given situation, our approach computes only the impulse response samples that are audible to a human listener.

5.2 Comparison with Previous Works

In the supplementary video, we compare the sound generated by our method with 1500 rays to that of naive path tracing [Embrechts 2000] with 90,000 rays. Using the IR cache, we achieve better sound quality using our method than path tracing with around 50x as many rays. Since path tracing computes the IRs for each frame independently without any history information, it is susceptible to under-sampling artifacts (e.g. unnatural variations in volume, spatial sound) unless a very large number of rays is traced. On the other hand, our approach produces IRs that change smoothly by taking into account many frames of history. At a cost of some responsiveness (controlled via τ) and extra storage for the IR cache, our system can generate high-quality sound using substantially fewer rays, thereby improving interactivity. In the video, we show how the sound quality changes as the value of τ varies from long (3.0s), to short (0.5s), and then finally 0.0s (equivalent to path tracing). The overall performance benefit of our method over standard path tracing is at least 50x.

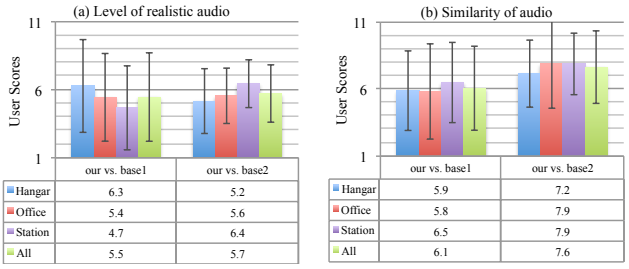
Recent work in sound has used temporal coherence to improve the results of path tracing for interactive sound propagation. One technique suggests a *diffuse path cache* that caches individual ray reflection paths and stores a moving average of the sound energy for each path in the cache [Schissler et al. 2014]. This approach achieves roughly a 10x speedup over standard path tracing by using the diffuse path cache. However, the memory overhead required to store all of the paths (possibly millions) is significant. On our benchmarks, this method requires $200 - 300$MB of storage *per sound source*. Also, the diffuse path cache must eventually remove paths from the cache to keep from growing too large. This process can alter the characteristics of the resulting sound, causing there to be slightly less reverberation than for a ground-truth simulation.

In comparison, the IR cache is roughly 5x faster than the diffuse path cache. We show the performance for each method on our benchmarks in Figure 7. The IR cache does not need to discard any past results, so effectively an infinite amount of history can be used with no additional computation or memory overhead by increasing the value of the response time τ. The memory required for the IR cache approach is only a few additional MB per sound source, and it scales linearly with the length of the IR. In Figure 8, we show that the IR cache uses around 2 orders of magnitude less memory than the diffuse path cache. In the video comparison, the IR cache achieves similar or better sound quality.

6 User Evaluation

We performed a preliminary user evaluation to study the subjective impact of our sound propagation approach compared to previous methods. In the study, we compared the sound generated by our interactive approach, called *our* method, to the sound generated when our impulse response modeling is disabled, called the *base* method. For the *base* method, we test two conditions. In the first, denoted by *base1*, the number of rays and performance is the same as for *our* method (see Table 1). In the second, denoted by *base2*, a ground-truth offline simulation is performed with 200,000 rays.

Study Design: There are 2 comparison conditions used for each scene: *our* vs. *base1* and *our* vs. *base2*, for a total of 6 conditions. The study was conducted as an online survey where each subject was presented with a 6 pairs of identical videos with different audio in a random order and asked two questions about each pair. The questions were (a) "which video has the more realistic audio?" and (b) "how similar is the audio in the videos?". The responses were recorded on a scale from 1 to 11. For the first question, an answer of 1 indicates the left video was much more realistic, an answer of



	our vs. base1	our vs. base2
Hangar	6.3	5.2
Office	5.4	5.6
Station	4.7	6.4
All	5.5	5.7

	our vs. base1	our vs. base2
Hangar	5.9	7.2
Office	5.8	7.9
Station	6.5	7.9
All	6.1	7.6

Figure 9: *We compare the user evaluation results for* our *approach versus the* base1 *and* base2 *methods for 3 scenes and 2 comparison cases. For question (a), a score below 6 indicates a preference for the first method in the comparison, whereas a score above 6 indicates a preference for the second method. For question (b), the higher the score, the more similar the audio for the two methods under comparison.*

11 indicates the right video was much more realistic, and an answer of 6 indicates the videos were equal. On the second question, an answer of 1 indicates the videos sounded extremely different, while an answer of 11 indicates the videos sounded very similar. Our research hypotheses were: (1) *Our* method produces sound that is as realistic as *base2*, and more realistic than *base1*; (2) The sound generated by *our* method will be more similar to that of *base2* than it is to *base1*.

Study Procedure: The study was completed by a total of 18 subjects between the ages of 20 and 31, made up of 15 males and 3 females. The average age of the subjects was 25.2, and all subjects had normal hearing. At the start of the study, subjects were given detailed instructions and filled out a demographic information questionnaire. Subjects were required to use either headphones or earbuds when taking the study, and they were asked to calibrate their listening volume using a test audio clip. Subjects were then presented the 6 pairs of videos and responded to the questions for each pair. The subjects were allowed to replay the videos as many times as needed, and they were able to move forward and backward in the study to change their answers if necessary. After rating the 6 video pairs, the study was completed.

6.1 User Evaluation Results

The results of our preliminary user evaluation are summarized in Figure 9. For question (a), the subjects indicated the technique that had the more realistic audio. For the comparison between *our* method and the *base1* method, our method is slightly preferred across all scenes except for the Hangar with average scores between 4.7 and 6.3. When compared to the offline ground-truth simulation of *base2*, the preference for *our* method is less, with scores between 5.2 and 6.4. A two-tailed one-sample t-test across all scenes was used to test hypothesis 1 that *our* method will be slightly more realistic than *base2* ($p = 0.23$), and much more realistic than *base1* ($p = 0.34$). Due to the statistical insignificance of these results we cannot rule out the possibility that the difference between the methods is due to other factors.

For question (b), the audio similarity between the techniques was considered. The subjects judged *our* method to be most similar to the *base2* offline simulation with scores from 7.2 to 7.9, while *our* method was scored less similar to the *base1* interactive simulation with scores from 5.8 to 6.5. Hypothesis 2 was evaluated by comparing the *our* vs. *base1* and *our* vs. *base2* conditions with a two-tailed Welch's t-test. In this case, the results are more significant ($p = 0.11$), but do not conclusively support the hypothesis that our method is more similar to *base2* than it is to *base1*. Overall, the large standard deviations of the scores indicate that subjects have difficulty discerning the subtle differences between the methods for the study questions.

7 Conclusions

In this work, we have presented two novel approaches of impulse response modeling for interactive sound propagation. The IR cache enables multiple frames of sound propagation history to be taken into account, thereby reducing the ray tracing required on each frame. Through the use of our adaptive IR length approach, the ray tracing is optimized for sources of varying power levels and scenes with changing reverberation times. We have evaluated the performance and memory requirements of our method on several complex dynamic scenes and observed a significant improvement compared to the prior state of the art in interactive sound propagation.

Our approach has some *limitations*. As it is based on geometric acoustics, all standard limitations, such as less accuracy for lower frequencies, are applicable. By limiting the maximum diffraction order to 3, some diffraction paths may be missed by our approach. In addition, our IR caching technique introduces some error in the sound field by smoothing impulse responses over time, though the effects of this error can be managed by tuning how the parameter τ varies over the IR length. Due to differences in the hearing threshold of different individuals, the use of equation 4 may not necessarily apply to all users of our system, and so our adaptive IR length algorithm may not align well with every user. However, an accurate threshold can be determined for each individual through personalized measurements or evaluation. In addition, we allow the maximum IR length L_{max} to change by only ΔL on each frame. This may result in slow responses to abrupt changes in the scene, such as a door that is suddenly closed.

There are many avenues for future work. To improve the statistical significance of the user evaluation, more subjects and better screening processes for subject listening abilities are needed. We would also like to investigate if sound source masking or other psychoacoustic effects can be used to enhance our sound propagation algorithm. For instance, quiet sound sources may be inaudible if there is a loud noise, and in this case some additional computation can be saved. Another challenge for sound propagation is the determination of accurate acoustic material properties. In the future, we would like to use the visual appearance of surfaces in a scene to estimate the materials automatically.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful feedback, as well as the anonymous participants of the user study. This work is supported in part by NSF award 1456299.

References

BARRON, M. F. 1974. *The effect of early reflections on subjective acoustical quality in concert halls*. PhD thesis, University of Southampton.

BATTENBERG, E., AND AVIZIENIS, R. 2011. Implementing real-time partitioned convolution algorithms on conventional operating systems. In *Proceedings of the 14th International Conference on Digital Audio Effects. Paris, France*.

BEGAULT, D. R., WENZEL, E. M., AND ANDERSON, M. R. 2001. Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on

the spatial perception of a virtual speech source. *Journal of the Audio Engineering Society 49*, 10, 904–916.

BEGAULT, D. R. 1996. Audible and inaudible early reflections: thresholds for auralization system design. In *Audio Engineering Society Convention 100*, Audio Engineering Society.

BORISH, J. 1984. Extension to the image model to arbitrary polyhedra. *The Journal of the Acoustical Society of America 75*, 6 (June), 1827–1836.

BROWN, R. G. 1956. Exponential smoothing for predicting demand. In *Proceedings of the 10th national meeting of the Operations Research Society of America, San Francisco*.

CARVALHO, A. P. O. 1995. The use of the sabine and eyring reverberation time equations to churches. *The Journal of the Acoustical Society of America 97*, 5, 3319–3319.

CHRISTENSEN, C., AND KOUTSOURIS, G. 2013. Odeon manual, chapter 6.

CHRISTENSEN, C. L., AND RINDEL, J. H. 2005. A new scattering method that combines roughness and diffraction effects. In *Forum Acousticum, Budapest, Hungary*.

EMBRECHTS, J. J. 2000. Broad spectrum diffusion model for room acoustics ray-tracing algorithms. *The Journal of the Acoustical Society of America 107*, 4, 2068–2081.

FLETCHER, H. 1940. Auditory patterns. *Reviews of modern physics 12*, 1, 47.

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

KUTTRUFF, H. 2007. *Acoustics: An Introduction*. Taylor and Francis, New York.

MEHRA, R., RAGHUVANSHI, N., ANTANI, L., CHANDAK, A., CURTIS, S., AND MANOCHA, D. 2013. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Trans. on Graphics 32*, 2, 19:1–19:13.

MEHRA, R., ANTANI, L., KIM, S., AND MANOCHA, D. 2014. Source and listener directivity for interactive wave-based sound propagation. *Visualization and Computer Graphics, IEEE Transactions on 20*, 4, 495–503.

MÜLLER-TOMFELDE, C. 2001. Time-varying filter in non-uniform block convolution. In *Proc. of the COST G-6 Conference on Digital Audio Effects*.

NEHAB, D., SANDER, P. V., LAWRENCE, J., TATARCHUK, N., AND ISIDORO, J. R. 2007. Accelerating real-time shading with reverse reprojection caching. In *Graphics hardware*, vol. 41, 61–62.

POLLOW, M., NGUYEN, K.-V., WARUSFEL, O., CARPENTIER, T., MÜLLER-TRAPET, M., VORLÄNDER, M., AND NOISTERNIG, M. 2012. Calculation of head-related transfer functions for arbitrary field points using spherical harmonics decomposition. *Acta acustica united with Acustica 98*, 1, 72–82.

PULKKI, V., ET AL. 2001. *Spatial sound generation and perception by amplitude panning techniques*. Helsinki University of Technology.

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 simualtion for real-time sound propagation of dynamic sources in complex scenes. *ACM Trans. on Graphics 29*, 4, 68:1 – 68:11.

ROBINSON, D., AND DADSON, R. 1957. Threshold of hearing and equal-loudness relations for pure tones, and the loudness function. *The Journal of the Acoustical Society of America 29*, 12, 1284–1288.

SCHERZER, D., JESCHKE, S., AND WIMMER, M. 2007. Pixel-correct shadow maps with temporal reprojection and shadow test confidence. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, Eurographics Association, 45–50.

SCHISSLER, C., AND MANOCHA, D. 2014. Interactive sound propagation and rendering for large multi-source scenes. Tech. rep., University of North Carolina at Chapel Hill.

SCHISSLER, C., MEHRA, R., AND MANOCHA, D. 2014. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. In *Proc. of ACM SIGGRAPH*, vol. 33, 1–12.

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 (September), 1624–1635.

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

TAYLOR, M., CHANDAK, A., ANTANI, L., AND MANOCHA, D. 2009. Resound: interactive sound rendering for dynamic virtual environments. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, ACM, 271–280.

TERHARDT, E. 1979. Calculating virtual pitch. *Hearing research 1*, 2, 155–182.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proc. of ACM SIGGRAPH*, 545–552.

TSINGOS, N., DACHSBACHER, C., LEFEBVRE, S., AND DELLEPIANE, M. 2007. Instant sound scattering. In *Proceedings of the Eurographics Symposium on Rendering*, 111–120.

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

VALIMAKI, V., PARKER, J. D., SAVIOJA, L., SMITH, J. O., AND ABEL, J. S. 2012. Fifty years of artificial reverberation. *Audio, Speech, and Language Processing, IEEE Transactions on 20*, 5, 1421–1448.

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

WENZEL, E. M., MILLER, J. D., AND ABEL, J. S. 2000. A software-based system for interactive spatial sound synthesis. In *ICAD, 6th Intl. Conf. on Aud. Disp*, 151–156.