Interactive Sound Propagation and Rendering for Large Multi-Source Scenes

Carl Schissler and Dinesh Manocha
University of North Carolina at Chapel Hill

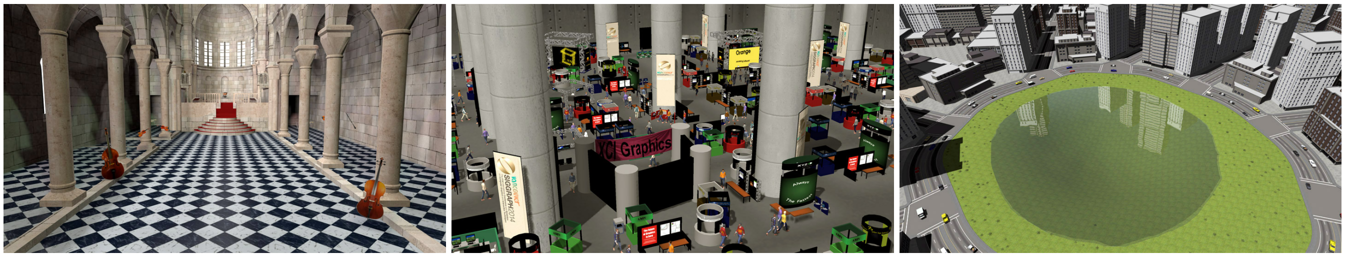

Fig. 1. Our interactive propagation and rendering algorithms can generate plausible acoustic effects, including early reflections and late reverberation, for large and complex indoor and outdoor scenes with many sources: (left) Sibenik cathedral, 18 sources; (center) Tradeshow, 200 sources; (right) City, 50 sources.

We present an approach to generate plausible acoustic effects at interactive rates in large dynamic environments containing many sound sources. Our formulation combines listener-based backward ray tracing with sound source clustering and hybrid audio rendering to handle complex scenes. We present a new algorithm for dynamic late reverberation that performs high-order ray tracing from the listener against spherical sound sources. We achieve sub-linear scaling with the number of sources by clustering distant sound sources and taking relative visibility into account. We also describe a hybrid convolution-based audio rendering technique that can process hundreds of thousands of sound paths at interactive rates. We demonstrate the performance on many indoor and outdoor scenes with up to 200 sound sources. In practice, our algorithm can compute over 50 reflection orders at interactive rates on a multi-core PC, and we observe a 5x speedup over prior geometric sound propagation algorithms.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Physically based modeling*; I.3.5 [**Computer Graphics**]: Applications—*Sound rendering*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: sound propagation, ray tracing, source clustering

1. INTRODUCTION

The geometric and visual complexity of scenes used in games and interactive virtual environments has increased considerably over the last few years. Recent advances in visual rendering and hardware technologies have made it possible to generate high-quality visuals at interactive rates on commodity GPUs. This has motivated increased focus on other modalities, such as sound rendering, to improve the realism and immersion in virtual environments. However, it still remains a major challenge to generate realistic sound effects in complex scenes at interactive rates. The high *aural complexity* of these scenes is characterized by various factors:

Large number of sound sources: There can be many, many sound sources in these scenes: from a few hundred to thousands. These sources may correspond to cars on the streets, crowds in a shopping mall or a stadium, or noise generated by machines on a factory floor.

Large number of objects: Many of these scenes consist of hundreds of static and dynamic objects. Furthermore, they may correspond to large architectural models or outdoor scenes spanning over tens or hundreds of meters.

Acoustic effects: It is important to simulate various acoustic effects including early reflections, late reverberations, echoes, diffraction, scattering, etc.

The high aural complexity results in computational challenges for sound propagation as well as for audio rendering. At a broad level, sound propagation methods can be classified into wave-based and geometric techniques. Wave-based methods, which numerically solve the acoustic wave equation, can accurately simulate all acoustic effects. However, they are limited to static scenes with few objects and are not yet practical for scenes with many sources. Geometric propagation techniques, based on ray theory, can be used to interactively compute early reflections (up to 5-10 orders) and diffraction in dynamic scenes with a few sources [Lentz et al.

Project Website: http://gamma.cs.unc.edu/MULTISOURCE

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

2007; Pelzer and Vorländer 2010; Taylor et al. 2012; Schissler et al. 2014].

A key challenge is to simulate late reverberation (LR) at interactive rates in *dynamic* scenes. The LR corresponds to the sound reaching the listener after a large number of reflections with decaying amplitude and corresponds to the tail of the impulse response [Kuttruff 2007]. Perceptually, LR gives a sense of the environment's size and of its general sound absorption. Many real-world scenarios, including a concert hall, a forest, a city street, or a mountain range, have a distinctive reverberation [Valimaki et al. 2012]. But this essential aural element, LR, is computationally expensive; using ray tracing in a typical room-size environment, calculating only 1-2 seconds of LR length requires the calculation of high-order reflections (e.g. >50 bounces) in moderately-sized rooms.

The complexity of sound propagation algorithms increases linearly with the number of sources. This limits current interactive sound-propagation systems to only a handful of sources. Many techniques have been proposed in the literature to handle multiple sources: sound source clustering [Tsingos et al. 2004], multi-resolution methods [Wang et al. 2004], and a combination of hierarchical clustering and perceptual metrics [Moeck et al. 2007], etc. to handle a large number of sources. However, a major challenge is to combine them with sound propagation methods to generate realistic reverberation effects.

A third major challenge in generating realistic acoustic effects is realtime audio rendering for geometric sound propagation. A dense impulse response for a single source generated with high order reflections can contain tens of thousands of propagation paths; hundreds of sources can result in millions of paths. Current audio rendering algorithms are unable to deal with such complexity at interactive rates.

Main Results: We present a novel approach to perform interactive sound propagation and rendering in large, dynamic scenes with many sources. Our formulation is based on geometric acoustics and can handle all three challenges described above. The underlying algorithm is based on backward ray tracing from the listener to various sources and is combined with sound source clustering and real-time audio rendering. Some of the novel components of our approach include:

(1) **Acoustic Reciprocity for Spherical Sources**: We use backward ray-tracing from the listener to compute higher-order reflections in dynamic scenes for spherical sound sources and observe a 5x speedup over forward ray tracing algorithms.

(2) **Interactive Source Clustering in Dynamic Scenes**: We present an algorithm for perceptually clustering distant sound sources based on their positions relative to the listener and relative source visibility. This is the first clustering approach that is applicable to both direct as well as propagated sound. We observe sub-linear scaling in the number of sources.

(3) **Hybrid Convolution Rendering:** We present a hybrid approach to render large numbers of sound sources in real time with Doppler shifting by performing either delay interpolation or partitioned convolution based on a perceptual metric. This results in more than 5x improvement over prior Doppler-shift audio rendering algorithms.

We have implemented this system on a 4-core PC, and its propagation and rendering performance scales linearly with the number of CPU cores. Our system is applicable to large complex scenes and it can compute over 50 orders of specular or diffuse reflection for tens or hundreds of moving sources at interactive rates. In addition, the hybrid audio rendering algorithm can process hundreds of thousands of paths in real time.

2. RELATED WORK

In this section, we give a brief overview of related work on interactive sound propagation, late reverberation, handling of complex scenes, multiple sound sources, and audio rendering.

Interactive Sound Propagation: Many wave-based and geometric propagation algorithms have been proposed for interactive sound propagation. The wave-based methods are more accurate, but their complexity increases significantly with the simulation frequency and the surface areas of the objects or the volume of the acoustic space. Many precomputation-based interactive algorithms have been proposed for wave-based sound propagation in static indoor and outdoor scenes [James et al. 2006; Tsingos et al. 2007; Raghuvanshi et al. 2010; Mehra et al. 2013; Yeh et al. 2013]. Most interactive sound propagation algorithms for large scenes with a high number of objects are based on geometric propagation. These include fast algorithms based on beam tracing [Funkhouser et al. 1998; Tsingos et al. 2001] and frustum tracing [Chandak et al. 2009] in static scenes. Recent advances in ray tracing have been used for interactive sound propagation in dynamic scenes [Lentz et al. 2007; Pelzer and Vorländer 2010; Taylor et al. 2012] and exploit the parallel capabilities of commodity CPUs and GPUs. Temporal coherence methods have also been proposed to improve the performance of interactive ray tracing [Schissler et al. 2014; Schissler et al. 2016]. In the area of sound source modeling, modal sound synthesis has recently been efficiently coupled with sound propagation in an interactive system [Rungta et al. 2016]. However, current algorithms are mostly limited to computing early reflections or can only handle a few sources.

Late Reverberation: Previous methods for computing interactive LR can be placed in three general categories: artificial reverb, statistical methods, and geometric precomputation. *Artificial reverberators* are widely used in games and VR and make use of recursive filters to efficiently produce plausible LR [Schroeder 1962], or can use convolution with a room impulse response [Valimaki et al. 2012]. Games often use artist-specified reverb filters for each region within a virtual environment. However, this is not physically based and is time-consuming to specify. Moreover, these reverberators cannot accurately reproduce outdoor late reverberation because they are designed to model the decay of sound in rooms. *Statistical techniques* estimate the decay rate for an artificial reverberator from the early reflections [Taylor et al. 2009]. These methods are applicable to dynamic scenes but have some limitations. The use of room acoustic models may not work well for outdoor environments, and cannot produce complex acoustic phenomena like coupled rooms and directional reverberation. Methods based on *geometric precomputation* use high-order ray tracing or some other sound propagation technique to precompute impulse response filters at various locations in an environment. At runtime, the correct filter is chosen and convolved with the audio. These methods can produce plausible reverberation and are inexpensive to compute, but also require lots of memory to store many impulse responses. Frequency-domain compression techniques have been used to reduce the storage required [Tsingos 2009; Raghuvanshi et al. 2010]. Other methods are based on the acoustic rendering equation [Siltanen et al. 2007] and combine early-reflection ray tracing with acoustic transfer operators to compute reverberation [Antani et al. 2012]. However, precomputed techniques cannot handle the acoustic effect of dynamic objects (e.g. doors). This work aims to generate these reverberation effects in real time with similar accuracy.

Handling Complex Datasets: There is extensive literature to accelerate visual rendering of complex datasets based on model simplification, image-based simplification and visibility computations [Yoon et al. 2008]. Some of the ideas from visual rendering have been extended or modified and applied to sound rendering. These include methods based on interactive ray tracing and precomputed radiance transfer that are used for sound propagation. Level-of-detail techniques have also been used for acoustic simulation [Siltanen et al. 2008; Pelzer and Vorländer 2010; Tsingos et al. 2007; Schissler et al. 2014].

Large Number of Sources: Many hierarchical and clustering techniques have been proposed to render such scenes. Current methods perform source clustering using clustering cones [Herder 1999], perceptual techniques [Tsingos et al. 2004], or multi-resolution methods [Wand and Straßer 2004]. Other algorithms are based on recursive clustering [Moeck et al. 2007], which classify sources into different clusters based on a dynamic budget. The use of perceptual sound masking has also been proposed to reduce the number of sources that must be rendered [Tsingos et al. 2004; Moeck et al. 2007]. These techniques are aimed at optimizing digital signal processing for audio rendering once all sound paths and sources have been computed, and therefore can't be directly used to accelerate the computation of sound propagation paths from each source to the listener.

Interactive Audio Rendering: Most current interactive techniques for generating smooth audio in dynamic scenes are based on interpolation and windowing techniques [Savioja et al. 2002; Taylor et al. 2012; Tsingos 2001]. Other techniques use fractionally-interpolated delay lines to perform a direct convolution with the propagated paths [Savioja et al. 1999; Wenzel et al. 2000; Tsingos et al. 2004] or dynamic convolution [Kulp 1988]. Fouad et al. [1997] present a level-of-detail audio rendering algorithm by processing every k-th sample in the time domain. Time-varying impulse responses are rendered using interpolation in the time domain [Müller-Tomfelde 2001], or efficient interpolation in the frequency domain [Wefers and Vorländer 2014] that can reduce the number of inverse FFTs required. Low-latency processing of hundreds of channels can be performed in real time on current hardware using non-uniform partitioned convolution [Battenberg and Avizienis 2011].

3. INTERACTIVE PROPAGATION & RENDERING

In this section we present our algorithms for sound propagation and audio rendering in complex scenes. Our sound propagation approach is based on Geometric Acoustics (GA) and we assume a homogeneous propagation medium. GA algorithms assume that the scene primitives are larger than the wavelength; in other cases we use mesh simplification techniques to increase the primitive size [Siltanen et al. 2008; Schissler et al. 2014]. We use ray-tracing-based algorithms to compute specular and diffuse reflections [Krokstad et al. 1968; Vorländer 1989; Taylor et al. 2012] and approximate wave effects with higher-order edge diffraction based on the Uniform Theory of Diffraction (UTD) [Tsingos et al. 2001; Taylor et al. 2012; Schissler et al. 2014].

In order to handle scenes with high aural complexity, we present new algorithms for interactive late reverberation in dynamic scenes using backward ray tracing (Section 3.1); source clustering (Section 3.2); sound propagation for clustered sources (Section 3.3); and a hybrid convolution audio rendering algorithm for Doppler shifting (Section 3.4). The overall pipeline is shown in Figure 2.

3.1 Acoustic Reciprocity for Spherical Sources

The computation of high-order reflections is an important aspect of geometric sound propagation. Most of the reflected acoustic energy received at the listener's position after the early reflections in indoor scenes is due to late reverberation, the buildup and decay of many high-order reflections [Kuttruff 2007]. It has been shown that after the first 2 or 3 reflections, scattering becomes the dominant effect in most indoor scenes, even in rooms with relatively smooth surfaces [Lentz et al. 2007]. In addition, the sonic characteristics of the reverberation such as decay rate, directional effects, and frequency response vary with the relative locations of sound sources and listeners within a virtual environment. As a result, it is important to compute late reverberation in dynamic scenes based on high-order reflections and to incorporate scattering effects.

Previous work on geometric diffuse reflections has focused on Monte Carlo path tracing [Embrechts 2000]. These methods uniformly emit many rays or particles from each sound source, then diffusely reflect each ray through the scene up to a maximum number of bounces. Each ray represents a fraction of the sound source's total energy, and that energy is attenuated by both reflections off of objects in the scene and by air absorption as the ray propagates. If a ray intersects a listener, usually represented by a detection sphere the size of a human head, that ray's current energy is accumulated in the output impulse response (IR) for the sound source. These approaches are generally limited to low orders of reflections for interactive applications due to the large number of rays required for convergence.

More recently, the concepts of "diffuse-rain", proposed by Schröder [2011], and "diffuse-cache", proposed by [Schissler et al. 2014], have been used to accelerate interactive sound propagation. With diffuse-rain, each ray estimates the probability of the reflected ray intersecting the listener at every hit point along its path, rather than relying on rays to hit the listener by random chance. On the other hand, the diffuse-cache takes advantage of temporal coherence in the sound field to accelerate the computation of ray-traced diffuse reflections. A cache of paths that were detected during previous frames is stored as a separate hash table for each sound source. The cache is used to maintain a moving average of the sound energy for propagation paths grouped together based on a scene surface subdivision of user-defined resolution.

However, the performance of these techniques is not interactive in scenes with many sound sources and high-order reflections. Each source emits many rays (e.g. thousands or tens of thousands), and the total cost scales linearly with the number of sound sources and reflection bounces that are simulated. Moreover, many of the rays that are emitted from the sources may never reach the listener, especially if the source and listener are in different parts of an interconnected environment. This results in a large amount of unnecessary computation.

We propose a new technique for simulating a high number of reflections in scenes with a large number of sources using backward ray tracing. We take advantage of the principle of acoustic reciprocity which states that the sound received at a listener from a source is the same as that produced if the source and listener exchanged positions [Case 1993]. Rather than emitting many rays from each sound source and intersecting them with the listener, we trace rays backwards from only the listener's position and in-

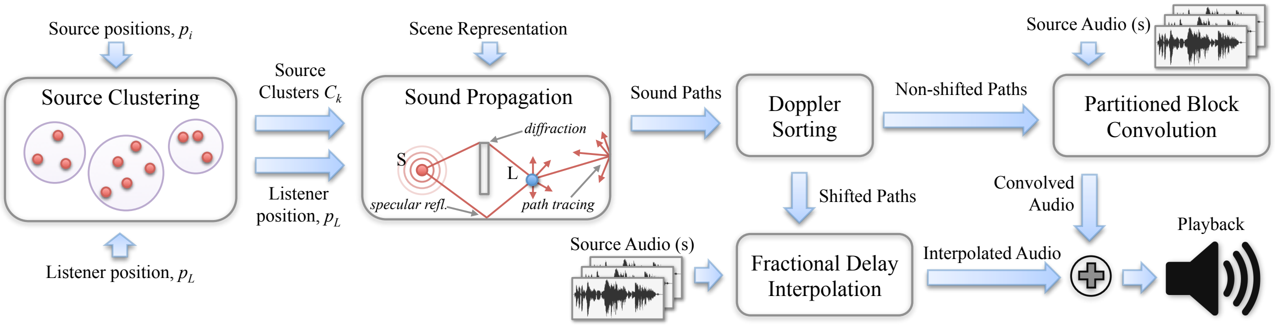

Fig. 2. An overview of our sound propagation and auralization pipeline. On each propagation frame, the sound sources are merged into clusters, then sound propagation is performed from the listener's position, computing early and late reflections using backward ray tracing. The output paths are then sorted based on the amount of Doppler shifting. Paths with significant shifting are rendering using fractional delay interpolation, while other paths are accumulated into an impulse response, then rendered using partitioned convolution. The final audio for both renderings is then mixed together for playback.

tersect them with sound sources. This provides significant savings in the number of rays required since the number of primary rays traced is no longer linearly dependent on the number of sources. We achieve better scaling with the number of sources than with forward ray tracing, which allows our system to compute high-order reflections for complex scenes with many sources at interactive rates. We represent sound sources as detection spheres with non-zero radii, though our formulation can be applied to sources with arbitrary geometric representation. Moreover, we combine the diffuse-cache with diffuse-rain technique to increase the impulse-response density for late reverberation. Our approach computes specular reflections, diffuse reflections, and diffraction effects separately and combines the results.

Our approach begins by emitting uniform random rays from the listener, then reflecting those rays through the scene, up to a maximum number of bounces. Vector-based scattering [Christensen and Koutsouris 2013] is used to incorporate scattering effects with a scattering coefficient $s \in [0, 1]$ that indicates the fraction of incident sound that is diffusely reflected [Christensen and Rindel 2005]. With this formulation, the reflected ray is a linear combination of the specularly reflected ray and a ray scattered according to the Lambert distribution, where the amount of scattering in the reflection is controlled by s. At each bounce, a ray is traced from the reflection point to each source in the scene to check if the source is visible. If so, the contribution from the source is accumulated in the diffuse cache. The hash code that uniquely identifies a path in the cache is incrementally computed as each ray propagates through the scene. The hash for a path of length d is given by the expression $\sum_{i=0}^{d}(i+1)T_i$, where T_i is the hash code for the ith triangle patch that was hit along the ray path.

We also extend the image source method to computing specular reflection paths for spherical sources by sampling the visibility of each path using random rays. To find specular paths, rays are traced from the listener and *specularly* reflected through the scene to find potential sound paths. Each combination of reflecting triangles is then checked to see if there is a valid specular path as shown in Figure 3. First, the listener's position is recursively reflected over the sequence of planes containing the triangles, as in the original image source algorithm. Then, a small number of random rays (e.g. 20) are traced backwards from the source in the cone containing the source sphere with vertex at the final listener image position. These rays are specularly reflected over the sequence of triangles back to the listener. The intensity of that specular path is multiplied by the fraction of rays that reach the listener to get the final intensity. The benefit of this approach to computing specular sound for area sources is that source images can become partially occluded, resulting in a smoother sound field for a moving listener. On the other hand, point sources produce abrupt changes in the sound field as specular reflections change for a moving listener. Modeling sources as spheres allows large sound sources (e.g. cars, helicopters) to be represented more accurately than with point sound sources.

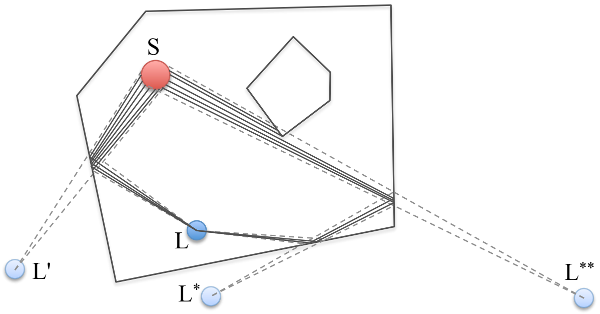

Fig. 3. This example shows how specular sound is computed for spherical sources. A first-order and second-order reflection are visible. The listener is reflected recursively over the sequence of reflecting planes, then the cone containing the source and the last listener image (L' or L**) is sampled using a small number (e.g. 20) random rays. These rays are specularly reflected back to the listener L. The fraction of rays that are not occluded by obstacles is multiplied by the energy for the specular path to determine the final energy.

After all rays are traced on each frame, the current contents of the diffuse cache for each source are used to produce output impulse responses for the sources. The final sound for each reflection path is a linear combination of the specular and diffuse sound energy based on the scattering coefficient s.

3.2 Source Clustering

The performance of most sound propagation algorithms scales linearly with increasing numbers of sources and this is a significant bottleneck for the interactive simulation of large complex scenes. One way to achieve sub-linear scaling and reduce the computation

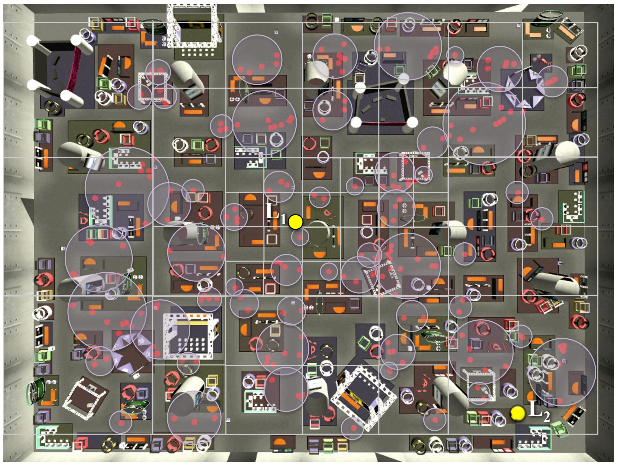

Fig. 4. In this top-down view of the Tradeshow scene with 200 sources, clustering has been done for listener L_1. Clustering at listener L_1 produces 95 clusters (2.1x reduction in number of sources), while clustering at listener L_2 produces 52 clusters (3.8x reduction), since the clustering can be more aggressive. The size of nodes in the octree used to accelerate clustering increases with the distance from the listener.

required for propagation and rendering is to cluster sound sources. Prior techniques for handling large number of sources [Tsingos et al. 2004; Wand and Straßer 2004; Moeck et al. 2007] have been mainly used for auralization to generate direct sound or are combined with pre-computed filters to generate different acoustic effects. These techniques do not take into account the position of obstacles in the scene or the relative visibility of sound sources or listeners when performing clustering. As a result, they may not work well when the sources are in different acoustic spaces (e.g. adjacent rooms separated by a wall) or in scenes with many occluding obstacles.

We present an efficient approach for clustering sound sources that uses visibility computations to handle complex scenes and high-order reflections and diffraction. Our formulation is based on the observation that distant or occluded sound sources can be difficult to distinguish individually. This happens when the sound field at a point is mainly due to late reverberation that masks the location of sources [Griesinger 2009]. We use a distance and angle-based clustering metric similar to that of [Tsingos et al. 2004], where sources are clustered more aggressively at greater distances from the listener. Our approach uses the relative visibility among the sources in order to avoid incorrectly clustering sources that are in different rooms.

Given a scene with a list of spherical sources S_j and listener position L, a set of source clusters C_k is computed based on our clustering metric. We use a dynamic octree to efficiently partition sources into groups that can then be independently clustered. We ensure that the size of the leaf nodes of the octree is governed by our clustering metric. In particular, the size of the nodes increases with the distance of the node from the listener, as shown in Figure 4.

Relative Source Visibility: To determine if a pair of sources are in the same acoustic space and therefore candidates for clustering, we use ray tracing to compute the mutual visibility of the sources. In the simplest form, two sources cannot be clustered if there is no direct line-of-sight between them. This can be efficiently evaluated for point sources by tracing a single ray between each pair of sources in a potential cluster. However, this binary visibility may increase the number of clusters that are produced and it neglects the ability of sound to readily reflect and diffract around obstacles.

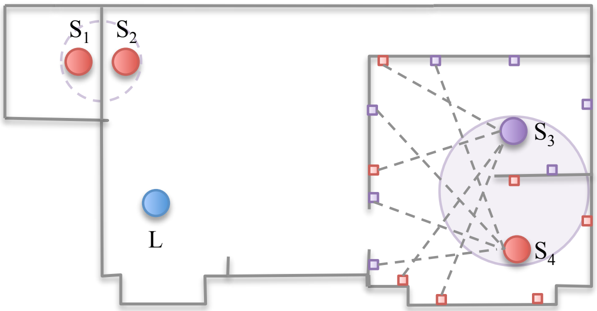

Fig. 5. Previous clustering approaches do not consider obstacles in the scene while generating the clusters. As a result, they may incorrectly cluster the sources S_1 and S_2 that are close to each other but in different rooms. The listener hears clustered source S_1 even though the source should be inaudible. However, in our approach, we computed a soft relative visibility term ν by tracing a small number of random rays from each source (colored red or purple), then determining the fraction of the ray-scene intersections that are visible to the other source. In this scenario, S_3 and S_4 do not have direct line-of-sight visibility but will still be clustered using our approach since they have significant first-order visibility and can be assumed to reside in the same acoustic space.

We propose a *soft* visibility metric that instead traces a small number of rays (e.g. 50) from each sound source in uniform random directions and finds the intersection points with obstacles in the scene. An additional ray is traced from each of the ray-scene intersections of S_1 to source S_2, as well as from each of S_2's ray-scene intersections to source S_1. For two sources S_1 and S_2, the soft relative visibility $\nu \in [0, 1]$ is determined by the fraction of these scene intersection points that are visible to the other source and not occluded by any obstacles. If the relative visibility ν of two sources is greater than some threshold amount ν_{min}, the sources are clustered. We use $\nu_{min} = 0.5$, indicating that sources must be visible to at least 50% of the intersection points to be clustered. With this approach, sources are clustered if they can see the same parts of the scene, rather than only considering line-of-sight visibility. Figure 5 illustrates this approach.

After sources have been partitioned into valid clusters, each cluster is then used for sound propagation as a proxy for the sound source(s) it contains. The proxy source for each cluster is represented during sound propagation by a larger bounding sphere around its individual sources that is centered at their centroid.

3.3 Clustered Sound Propagation

In order to deal with clusters rather than individual sound sources, it is necessary to modify our sound propagation algorithm. Clustered sources use a larger detection sphere, which may result in too much sound energy for sources with small radii that are part of a cluster. In addition, the caches that are used to exploit temporal coherence need to be handled differently for the clusters.

After the sources in the scene have been clustered based on our clustering criteria, the sound propagation algorithm proceeds as described in Section 3.1. Each cluster is propagated as if it were a spherical sound source with that cluster's centroid and bounding sphere radius. This approach works well for computing specular and diffraction sound paths, since those algorithms assume point sound sources. However, the sound energy computed for clusters is incorrect, since the probability of hitting a large spherical clustered source detector is much greater than the probability of hitting a small unclustered source. We overcome this shortcoming by applying a normalization factor to the sound energy computation to compensate for the increased detector size. In a diffuse sound field, the probability of a ray hitting a spherical detector is proportional to the projected silhouette area of the sphere on a plane, πr^2. Therefore, in order to compensate for the increased hit probability, the normalization factor w is the ratio of the source's silhouette area to the cluster's silhouette area:

$$w = \frac{\pi r_i^2}{\pi r_{BS}^2}. \tag{1}$$

By applying this normalization factor to every diffuse path, the total diffuse energy from the source is approximately the same, independent of the source detector size or whether or not it is clustered. This allows the use of large cluster detectors for distant clustered sources without significantly affecting the sound quality.

Another consideration for clustered sound propagation is maintaining the cache data structures used to take advantage of temporal coherence. While computing the sound propagation paths, each source stores a cache of both specular and diffuse reflectance data (paths) from previous frames. When a previously unclustered source becomes a member of a cluster during the current frame, the cache for the source is merged with the cluster's cache, so that the cluster's cache now contains the accumulation of both caches. To merge the cache, all sound paths for the source are inserted in the cluster's cache. If there are any duplicate paths with the same hash code, their contributions are summed into a single path. Likewise, when a previously clustered source becomes unclustered, the cache for the cluster is copied and a copy is associated with the cache associated with the source. By handling the caches in this way, we generate smooth transitions in the acoustic responses for sources that become clustered.

3.4 Hybrid Convolution Rendering

An important aspect of interactive sound propagation is the auralization of the resulting propagation paths. For scenes with large numbers of sources and high-order reflections, there may be more than 10^6 individual paths to render. For moving sources and listeners, there may also be different amounts of Doppler shifting for each propagation path, depending on the extent of source or listener motion relative to the propagation medium. Therefore, in order to accurately render the audio for a dynamic real-time simulation, it may be necessary to render Doppler shifting for millions of propagation paths. Previous techniques for rendering Doppler shifting, such as *fractionally-interpolated delay lines* [Wenzel et al. 2000], become prohibitively expensive when the number of sound paths grows over a few thousand. On the other hand, partitioned frequency-domain convolution is ideal for efficiently rendering arbitrarily-complex impulse responses, but cannot perform accurate Doppler shifting.

We present a hybrid rendering system which uses interpolating delay lines for propagation paths with perceptually-significant Doppler shifting and partitioned impulse-response convolution for the other paths. By dynamically switching between these methods using a psychoacoustic metric, our approach can reduce the amount of computation for path delay interpolation because only a few sound paths must be rendered using the expensive interpolation. The input of our audio rendering algorithm is a list of acoustic responses and one or more sound sources (e.g. clustered sources) that should be rendered using each IR. During sound propagation, our approach determines which rendering method should be used for new sound paths based on the amount of Doppler shifting. We use the relative speed of the source and listener along each path, δv, to compute the shift amount, then compare this to a psychoacoustic threshold to sort the paths into one category or the other.

The amount of Doppler shifting s that shifts the frequency f to $\tilde{f}$ is given by the following well-known relation, where the motion of the source and listener are small relative to the speed of sound c:

$$\tilde{f} = sf = \left(1 + \frac{\delta v}{c}\right) f. \tag{2}$$

We convert the shift s to a signed shift in *cents* ($1/100$th of a halftone interval) on a log frequency scale using the relation $s_{cents} = 1200 \log_2(s)$. The Doppler shift amount s_{cents} is compared to a parameter s_{min} that signifies the threshold above which paths have significant shifting. If $|s_{cents}| > s_{min}$ for a given path, that path is rendered using a fractional delay line. Otherwise, the path is added to the convolution impulse response. Previous work in psychoacoustics has shown that the human ear has difficulty distinguishing pitch shifts of up to 20 cents or more [Geringer et al. 2012], so we use $s_{min} = 20$ cents. To maintain real-time performance for audio rendering of thousands of paths, our approach allows the value of s_{min} to be scaled based on the current system load. If there are too many Doppler-shifted paths to render interactively, the paths are sorted by decreasing amount of shift and weighted by the per-path sound intensity. As many paths as the CPU budget allows are then rendered from this sorted list using delay interpolation while the remaining paths are rendered using convolution.

4. IMPLEMENTATION

Sound Propagation: On each frame, we trace 1000 primary rays from the listener and recursively reflect them through the scene to determine possible specular, diffuse, and diffraction paths. This number of rays provided a nice tradeoff in terms of IR quality and computation time. We use a combination of the spherical image source method for specular paths and diffuse path tracing for diffuse reflections. Edge diffraction (up to order 3) is computed using the high-order UTD algorithm from [Schissler et al. 2014]. We use per-triangle material parameters to model frequency-dependent reflection attenuation and scattering [Christensen and Rindel 2005]. The spherical sources in our simulations correspond to the bounding spheres of the objects producing the sound. The number of reflections traced for each benchmark is summarized in Table I.

Temporal Coherence: Our system makes use of the *diffuse path cache* technique proposed by [Schissler et al. 2014] that uses a cache of sound energy from previous frames to incrementally compute the diffuse sound contribution for the current frame. This allows many fewer rays to be traced on each frame with similar sound quality, thereby improving the interactivity of the resulting sound propagation algorithm. However, this approach can also introduce some small errors in the resulting sound, especially for fast-moving sources, listeners, or objects in the scene. In such cases, the diffuse cache will take a few frames to update to the changes in the sound. The response time is controllable via a parameter τ. We use $\tau = 2$s. In practice, this provides a good balance between responsiveness and sound quality. We compare the results with and without the diffuse cache in the supplementary video and show that the slower response or update time is not very noticeable (or audible) in practice, even for fast-moving sources and listeners.

Table I. The main results for our sound propagation and rendering pipeline.

	Scene Complexity		Clustering		Propagation					Rendering
Scene	#Tris	#Sources	#Clusters	Time(ms)	#Specular	#Diffuse	Time(ms)	#Paths	#Doppler Paths	Time(ms)
Sibenik	75,273	18	14	0.02	10	50	49.2	208,832	7	15.6
Tradeshow	28,070	200	95	0.21	10	40	182.7	871,982	17	75.1
City	206,976	50	24	0.08	10	50	47.2	27,711	125	4.5
Space Station	35,653	9	9	0.01	10	100	34.8	108,492	5	11.2
Elmia R.R.	1,047	2	2	0.0	2	100	46.1	96,278	0	2.5

Audio Rendering: We render audio for our simulations at 44.1kHz using 4 logarithmically-distributed frequency bands to divide the human hearing range: $0 - 110$Hz, $110 - 630$Hz, $630 - 3500$Hz, and $3500 - 22050$Hz. This allows efficient use of 4-wide SIMD vector instructions. The fractional delay interpolation module renders frequency-dependent audio by pre-filtering source audio into bands that are written to a circular delay buffer. Doppler-shifted propagation paths are rendered by reading interpolated delay taps from the delay buffer using linear resampling, then mixing at the appropriate place in the output buffer. The convolution module uses a non-uniform partitioning scheme for low-latency streaming real-time convolution. Each sound path is spatialized using vector-based amplitude panning for arbitrary speaker arrays [Pulkki 1997]. Frequency-dependent IRs for convolution are computed by band-pass filtering each IR band by its respective filters and then summing the filtered IRs to produce the final time-domain IR. The partitions in each IR are updated at varying rates that correspond to the FFT size for the partition, with later parts of the IR updated at slower rates than the early parts. The outputs from the convolution and delay-interpolation modules are mixed for each source cluster, then each cluster's output is mixed to the final output buffer.

Parallelization: We make use of SIMD instructions and the multithreading capabilities of current CPUs to accelerate sound propagation and rendering. Our ray tracer uses a 4-wide BVH (bounding volume hierarchy) with efficient SIMD traversal for incoherent rays. The sound propagation module is highly parallelized and performance scales linearly with the number of available CPU cores. Each sound propagation thread computes the sound for a subset of the total number of rays traced per frame. The propagation paths produced by each thread are merged once that thread finishes its computation. The audio rendering module uses two concurrent processing threads that each manage a small pool of threads. One of the processing threads takes a buffer of propagation paths from the propagation module on each frame and asynchronously builds a frequency-dependent impulse response for each source in parallel using its thread pool. Once computed, each IR is atomically swapped with the previous IR for the source cluster. The other concurrent rendering thread runs from the output device driver and schedules the asynchronous processing of non-uniform partitioned convolution. The convolution processing runs on as many threads as there are partition sizes, where the thread for each partition size has its own scheduling requirements. In practice, our highly parallel architecture has resulted in a system that scales well to any number of CPU cores and can maintain load-balanced real-time processing without audio dropouts for scenes with hundreds of sources on commodity CPUs.

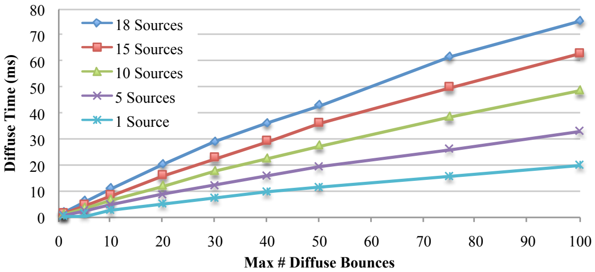

Fig. 6. We highlight the performance of our propagation algorithm as a function of the maximum diffuse reflection order in the Sibenik scene (no clustering) for 1, 5, 10, and 18 sound sources. Each additional sound source after the first only incurs a small further cost with backward ray tracing.

5. RESULTS AND ANALYSIS

5.1 Benchmarks

We evaluated our sound propagation system on indoor and outdoor scenes with large volumes and high model complexity. Our approach can generate plausible acoustic effects, including specular and diffuse reflections and edge-diffraction, at interactive rates on a 3.5 GHz 4-core CPU (see Table I).

Sibenik: This cathedral scene demonstrates the necessity of high-order diffuse reflections to generate plausible late reverberation. A virtual orchestra with 18 sound sources plays classical music.

Tradeshow: The listener walks around an indoor trade show floor with 200 people, where each person is a sound source. We show the benefits of clustering distant sources in order to reduce the computational load. Due to the small primitives in the visual model for this scene, we apply the mesh simplification technique of [Schissler et al. 2014]. The simplified model is shown in the video and appendix and the number of triangles is reported in Table I.

City: In a large outdoor city environment, there are 50 moving sound sources, including cars, trucks, planes, and helicopters. This scene shows how our approach can scale well to challenging large environments and can interactively compute reverberation for multiple moving sources.

Space Station: A door opens and closes, changing the reverberation for a loud occluded source. This scene shows how our backward ray tracing approach is robust to occlusion and can handle dynamic geometry.

5.2 Main Results

Backward Sound Propagation: By tracing rays backward from the listener, rather than from sources, our algorithm for high-order reflections significantly reduces the computation required to

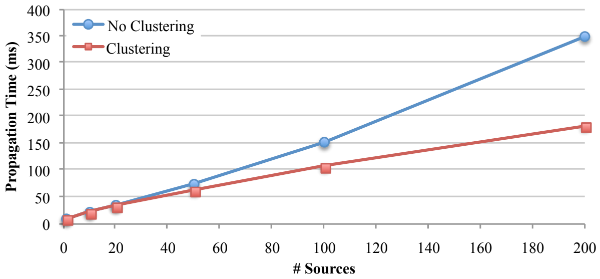

Fig. 7. We show how the time taken by our sound propagation system varies with the numbers of sources in the Tradeshow scene. We compare the results with and without source clustering. We obtain sub-linear scaling with the number of sources due to source clustering.

compute late reverberation for many sources. Figure 8 highlights the performance benefit of tracing rays from the listener when there are many sound sources. For 200 sources in the Tradeshow benchmark, backward ray propagation is 4.8 times faster than a forward ray tracing approach. We also demonstrate how the performance of our algorithm scales with the maximum reflection order. Figure 6 shows that the time for sound propagation is a linear function of the number of ray bounces. This clearly shows the advantage of backward ray tracing for scenes with multiple sources. There is a significant cost for the first sound source, but each additional source after the first incurs only a small further performance penalty. For example, backward ray tracing for 10 sources takes only about double the time that it takes for 1 source. This behavior can be explained because the same initial ray paths are used for all sound sources. The Space Station benchmark demonstrates a difficult condition for backward ray tracing. In this scene, a loud alarm sound source is inside of a box that is slightly ajar. The probability of detecting a path to the sound source is small. However, the diffuse rain sampling approach combined with the diffuse cache improves the chances significantly. In the video we show that our technique can generate plausible sound for this case. In order to compare the accuracy of our approach with traditional forward ray tracing, we show the error in the computed sound in Figure 9. We observe an average error of 0.5 dB in the total broad-band sound energy using our backward ray-tracing algorithm for a moving listener in the Sibenik benchmark when compared to forward ray tracing with 50k rays. These results show that backward sound propagation is a viable method for computing dynamic late reverberation effects in scenes with many sources at interactive rates.

Source Clustering: We have analyzed the runtime performance of our source clustering algorithm, as well as the error it introduces. In the Tradeshow benchmark with 200 sources, the clustering algorithm takes 0.21 ms and generates 95 clusters for sound propagation, on average. This corresponds to a 1.9x speedup, as shown in Figure 7. In general, the clustering reduces the number of sources by around a factor of 2 for the tested scenes. However, there may be some scenarios where the clustering can provide more significant reduction in the number of sources, especially when many of the sources are far away from the listener. This is illustrated for the Tradeshow scene in Figure 4, where a reduction in the number of sources of 3.8x is achieved for a listener that is far away from most sources, versus a reduction of 2.1x for a listener in the center of the sources. The time taken to compute the clustering for our benchmarks is shown in Table I. For 1000 sources, the clustering takes 0.81 ms. Overall, our clustering algorithm scales well with a large numbers of sources. In Figure 10, we analyze the error introduced by clustering for a single group of sources in the Sibenik scene. We find that our clustering approach only slightly affects the accuracy of the sound energy computed at the listener, introducing an average error of 1.98 dB. These results show our clustering algorithm can be used to effectively reduce the computation required for sound propagation in complex scenes, and that it introduces only a relatively small error in the final sound output.

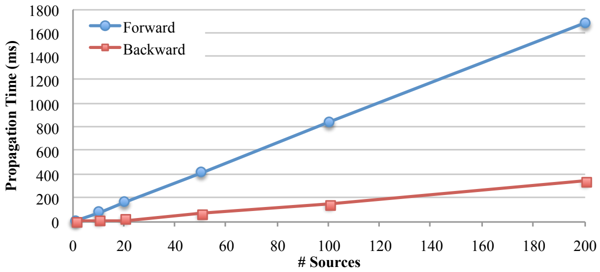

Fig. 8. This graph shows the performance of backward versus forward diffuse path tracing on the Tradeshow scene with varying numbers of sources (no clustering). By tracing rays from the listener, we obtain improved performance for many sound sources since fewer rays are traced overall. Performance remains linear in the number of sources due to the linear number of ray vs. sphere intersections that must be computed for backward ray tracing.

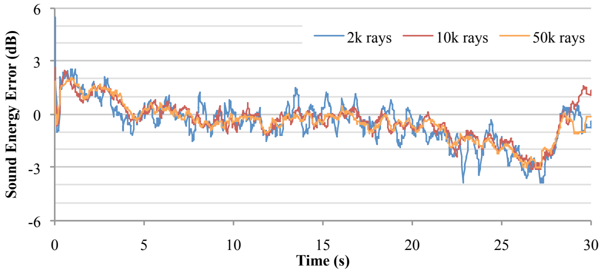

Fig. 9. This graph demonstrates the relative error in decibels that arises due to backward sound propagation from a moving listener in the Sibenik benchmark with 1 source. We compare the results for backwards ray tracing with 2k, 10k, and 50k rays to the ground-truth results for forward ray tracing with 50k rays. The average error for the each simulation is -0.53 dB, -0.37 dB, and -0.47 dB, respectively. We observe that backward sound propagation is a good approximation of forward sound transport, even with fewer numbers of rays.

Hybrid Sound Rendering: We summarize the performance of our sound-rendering approach for various scenes in Table I. The audio rendering computation is concurrent with propagation, and so the total time per frame is the maximum of the time for propagation or for rendering. In order to efficiently compute Doppler-shifted sound effects in aurally complex scenes, our algorithm uses a perceptual metric for the amount of Doppler shifting to sort and render the propagation paths. Our rendering system is able to render the audio for 200 sources and 871K propagation paths in the Tradeshow scene in real time by only rendering Doppler effects on a subset the paths, and using partitioned convolution for

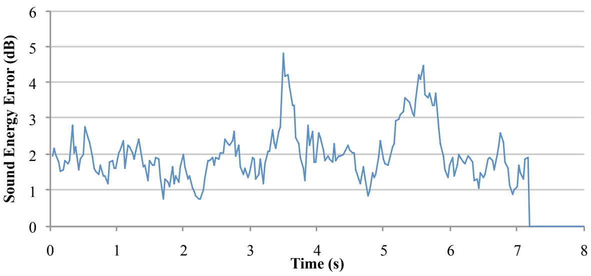

Fig. 10. We illustrate the error in the total sound energy in the Sibenik scene due to our clustering approach for a single cluster with 5 sources. In this case, the listener moves towards the cluster until it is divided into individual sources after 7 seconds. We present the maximum error in dB among all sources in the cluster for each frame. The average error for the entire simulation is 1.98 dB.

the other paths. In the supplementary video, we compare with a system that uses only delay-interpolation rendering. For the City scene, such an approach would require roughly 8x as much time to render Doppler effects as our hybrid technique. On the other hand, traditional convolution is fast but can't handle Doppler shifting. Our approach enables Doppler effect rendering for complex scenes with a small additional cost of only 10-20% over traditional convolution by the selective computation of Doppler shifting for significant paths.

5.3 Comparison with Previous Works

In this section we provide comparisons of our algorithms with prior techniques.

Interactive Late Reverberation: The generation of plausible dynamic late reverberation is regarded as a challenging problem in interactive sound propagation [Valimaki et al. 2012]. Previous interactive systems compute early reflections using ray tracing and combine them with statistical techniques for late reverberation [Antani and Manocha 2013; Taylor et al. 2012; Schissler et al. 2014]. However, statistical reverberation techniques are not able to model certain scenes and effects. Outdoor scenes and coupled spaces are a challenge because the reverb does not decay according to simple room acoustic models [Tsingos 2009]. In addition, dynamic scenes can have varying reverberation that is difficult to predict, such as when a door opens or closes or when a sound source or listener moves in an environment. Furthermore, statistical methods cannot handle directional reverberation effects, such as with a sound source at the other end of a long reverberant hallway that produces reverberant sound from the direction of the source. Techniques based on the acoustic-rendering equation require considerable precomputation, are primarily limited to static scenes, and can take about 193ms to handle a single source [Antani et al. 2012]. Moreover, their accuracy is governed by the sampling scheme and they cannot handle dynamic changes in the scene geometry that affect the quality of reverb, such as with a large door opening or closing. Other methods for LR computation are based on using a coarse spatial subdivision for wave-based simulation [Raghuvanshi et al. 2010] and are limited to small static indoor scenes.

In contrast, our system is able to compute dynamic late reverberation with no preprocessing for the Tradeshow scene with 200 sources in 182.7 ms, more than an order of magnitude faster than prior path tracing and other LR algorithms for the same scene. The advantages of our approach over offline reverb for a scene with a moving door can be seen in the supplementary video for the Space Station scene. In this scene, the door alters the reverberation and loudness of the sound source in a way that cannot be modeled with traditional reverberation approaches. On the other hand, benchmarks like the Tradeshow and Sibenik do not contain any dynamic occluders and so may not gain a significant benefit over precomputation or statistical approaches. Offline approaches may be more accurate in those cases because more rays can be traced and more orders of reflection and diffraction can be computed. In addition, important acoustic parameters like the reverberation time are frequently homogeneous in single-room environments (e.g. Figure 12) and so can be more efficiently modeled with global reverberation in those cases.

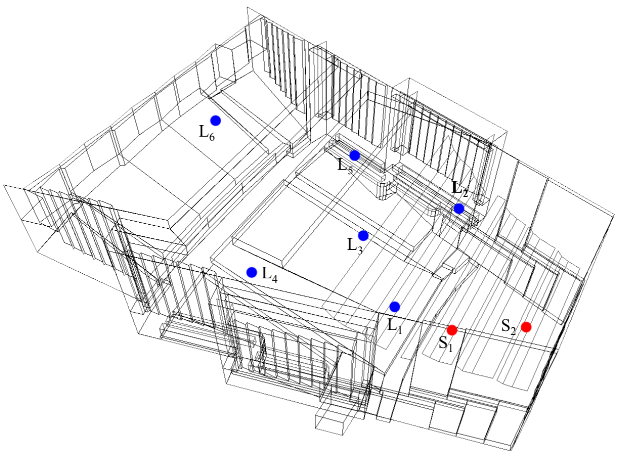

Fig. 11. The Elmia Round Robin benchmark scene that was used to compare the accuracy of our system to the commercial ODEON^{TM}acoustics software. The scene consists of a concert hall with 2 sources on the stage (red) and 6 listeners in the audience (blue). The benchmark is static. IRs were generated at each listener position and compared to both the ground-truth measurements and the results from ODEON.

Large Numbers of Sources: The prior techniques for handling large numbers of sources either focus on clustering sources for spatial sound rendering (e.g. HRTFs) or techniques based on reverberation filters, rather than geometric sound propagation. Our approach is complimentary to these methods. The algorithm in [Tsingos et al. 2004] can cluster 355 sources into 20 clusters in 1.14 ms, while the work of [Moeck et al. 2007] can cluster 1815 dynamic sound sources into 12 clusters for 3D sound spatialization in a game engine. However, when performing the clustering, these systems do not take into account the positions of the obstacles or the relative visibility of the sources and may therefore introduce more error. Figure 5 shows a situation where visibility is important for correct clustering. The results generated by our clustering algorithm use relative visibility information and so are better suited for sound propagation.

Sound Rendering: Prior approaches to rendering many sound sources have used stochastic importance sampling to reduce the number of propagation paths rendered for a sound source from 20K to 150 [Wand and Straßer 2004]. These methods can render 150 Doppler-shifted paths in real time. Another approach based on fractional delay lines is able to render 700 sound sources (propagation paths) on the CPU and up to 1050 on a GPU using tex-

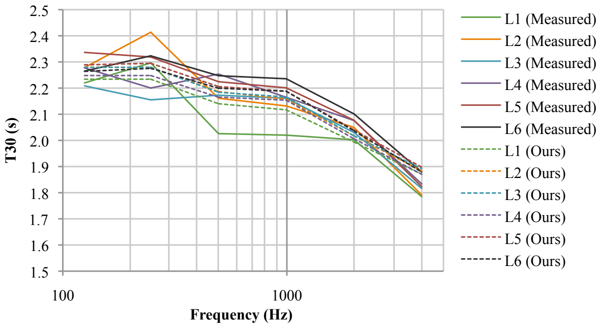

Fig. 12. This compares the ground-truth measurements in the Elmia Round Robin benchmark to our approach for the T_{30} room acoustic parameter at 6 listener positions.

ture resampling [Gallo et al. 2004]. However, these algorithms cannot handle the hundreds of thousands of paths that arise in our high aural complexity benchmarks at interactive rate. Frequency-domain convolution algorithms fare better because they do not operate on discrete propagation paths, but these algorithms cannot perform accurate Doppler shifting. The algorithm described in [Battenberg and Avizienis 2011] can render $100 - 200$ independent channels of time-invariant convolution on a 6-core CPU, while [Müller-Tomfelde 2001] describes a technique to efficiently implement time-varying convolution. In contrast, our algorithm can handle both Doppler shifting and large numbers of propagation paths; We use Doppler shifting information for each propagation path to sort and render the paths using either fractional delay lines or partitioned convolution. Our algorithm can thereby render 871K propagation paths for 200 sources in real time. We obtain significant benefit by performing expensive delay interpolation only when there is significant Doppler shifting. In the Tradeshow scene, our system renders 17 Doppler-shifted paths and 190 channels of partitioned convolution in real time.

5.4 Validation

In order to verify the accuracy of our results, we compare to the ground-truth measurements for the Elmia Round Robin benchmark [Bork 2000]. The scene is shown in Figure 11 and has 2 sound sources and 6 listener locations. The comparison was performed with respect to the measured room acoustic parameters T_{30}, *EDT*, C_{80}, D_{50}, G, and *TS* for 6 octave bands from 125Hz to 4kHz. A formal description of these parameters and how to compute them from an impulse response is given by ISO 3382 [ISO 2012]. In Figure 12, we present the results for the T_{30} parameter, corresponding to the time in seconds for the late reverberation to decay by 60dB. There is generally good agreement between our method and the measured data across most of the listener positions, with an average error of 2.4% across all listeners. This is less than the just noticeable difference (JND) of 5%, indicating that our system is plausible when compared to measured data. The results for the other parameters are presented in the appendix. In general, there is moderately good agreement (close to one JND) with the measurements across most of the acoustic parameters. However, the largest errors are at low frequencies, particularly in the 125Hz frequency band. These errors are similar to those of other geometric sound propagation systems on the same benchmark [Bork 2000]. This suggests that geometric sound propagation may not be sufficient to accurately model all low-frequency wave effects. Another possible source of error is the lack of accurate acoustic material properties for the virtual scene. Accurately modeling materials is still an open problem when performing simulations of real rooms.

We also compared our results on this benchmark to version 12 of the offline commercial architectural acoustics software ODEON$^{\text{TM}}$that has been shown to accurately predict the acoustic properties of real-world rooms [Rindel and Christensen 2003; Christensen et al. 2008]. ODEON uses a combination of ray tracing from sound sources for late reverberation and the image source method for early reflections (e.g. up to order 2) [Christensen and Koutsouris 2013]. Energy decay histograms were computed for each frequency band at each listener position in both ODEON and our sound propagation system. Figure 13 shows the results for listener position L_1. We found good correspondence between our system and ODEON for this position with an average discrepancy of -2.7dB, 0.75dB, -1.2dB, and 1.6dB for the 125Hz, 500Hz, 2kHz, and 8kHz bands respectively. At listener position L_5 that is more distant and toward one side of the room, we found similar results with differences of -0.4dB, 1.2dB, -1.8dB, and 0.1dB. Results for additional listener positions are provided in the appendix. These results demonstrate that our sound propagation system has accuracy comparable to existing commercial geometric acoustics systems.

6. CONCLUSIONS

We have presented an interactive algorithm for sound propagation and rendering in complex, dynamic scenes with a large number of sources. Our formulation combines fast backward ray tracing from the listener with sound source clustering to compute propagation paths. Furthermore, we use a novel, hybrid convolution audio rendering algorithm that can render hundreds of thousands of paths at interactive rates. We demonstrate our algorithm's performance on complex indoor and outdoor scenes with high aural complexity, and observe significant speedups over prior algorithms.

Our approach has a few *limitations*. Since it is based on geometric acoustics, all limitations of ray tracing-based algorithms, such as inaccuracies at lower frequencies, are inherent in the formulation. We assume homogeneous environments and our current implementation performs only a few orders of edge diffraction. We assume that the scene primitives are larger than the wavelength, though in some scenes (e.g. Tradeshow) that assumption is not valid due to the presence of small surfaces. This assumption can be partly satisfied using model simplification algorithms, though there are still many open issues in terms of computing good approximations for acoustic simulation. For example, we can't provide any tight bounds on the errors introduced in sound rendering due to model simplification. It is possible that our source clustering and hybrid convolution audio-rendering algorithms may not work well in certain cases, resulting in artifacts. The backward ray tracing approach can result in a different sampling of the sound field than would be generated by forward methods. In adversarial conditions like the Space Station scene, this may reduce the quality of the sound for occluded sources. However, neither backward nor forward ray tracing alone will handle all cases where there are significant occlusions. If the propagation algorithm only traces low-order specular reflections, our approach may not correctly simulate certain phenomena like flutter echoes. In addition, the diffuse cache can result in slower changes to the sound field with fast-moving sources or listeners. We approximate sound sources with spheres and this may not work well in some situations. Also, it may be possible for there to be rendering artifacts caused by updating the impulse response during convolution rendering, but these artifacts can be avoided by

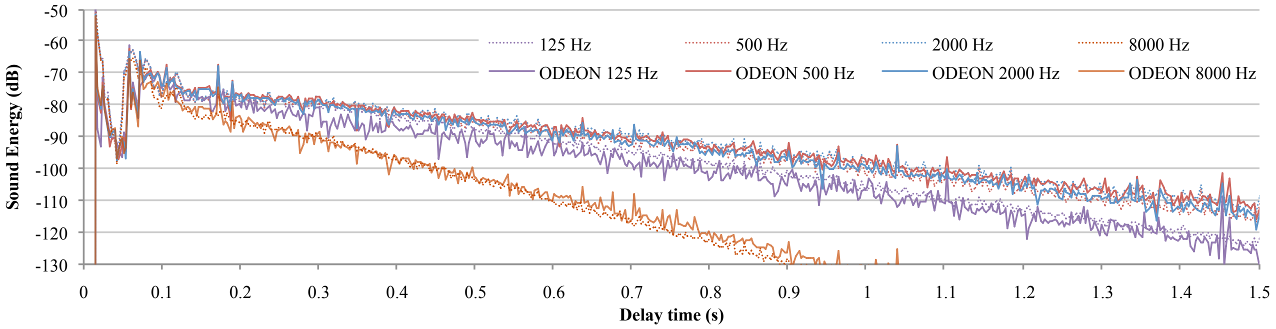

Fig. 13. This shows the energy decay histogram of the impulse response for the 125Hz, 500Hz, 2kHz and 8kHz frequency bands for both our system and the commercial ODEON$^{\mathrm{TM}}$acoustics software for listener position L_1 in the Elmia Round Robin scene. There is generally good agreement between the two systems. The average differences between the IRs were -2.7dB, 0.75dB, -1.2dB, and 1.6dB for the 125Hz, 500Hz, 2kHz, and 8kHz bands respectively.

increasing the maximum latency for changes to the IR and performing longer interpolation [Müller-Tomfelde 2001].

There are many avenues for future work. We would like to evaluate the approach in more scenarios, including challenging outdoor environments such as forests or mountains with complex reverberation effects. More validation with real-world measurements is also needed, especially for scenes with lots of occlusions or moving obstacles. However, we are not aware of any suitable existing datasets. We would like to perform a user study to evaluate the benefits of these acoustic effects in gaming and virtual environments. It would be useful to extend the approach to handle more complex sound sources (e.g. musical instruments), where each original source would need to be approximated with a high number of point sources. We would like to develop bidirectional path tracing algorithms and improved sampling strategies for sound propagation. Finally, we would like to take into account non-linear environmental effects, such as changes in the temperature and pressure in large outdoor scenes.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful feedback. This work is supported in part by NSF award 1456299.

REFERENCES

ANTANI, L., CHANDAK, A., SAVIOJA, L., AND MANOCHA, D. 2012. Interactive sound propagation using compact acoustic transfer operators. *ACM Trans. Graph. 31,* 1 (Feb.), 7:1–7:12.

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

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*.

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.

CASE, K. 1993. Structural acoustics: A general form of reciprocity principles in acoustics. Tech. rep. JSR-92-193. The MITRE Corporation.

CHANDAK, A., ANTANI, L., TAYLOR, M., AND MANOCHA, D. 2009. Fastv: From-point visibility culling on complex models. *Computer Graphics Forum (Proc. of EGSR) 28,* 3, 1237–1247.

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

CHRISTENSEN, C., NIELSEN, G., AND RINDEL, J. 2008. Danish acoustical society round robin on room acoustic computer modeling. *Odeon A/S: Lyngby, Denmark*.

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.

FOUAD, H., HAHN, J., AND BALLAS, J. 1997. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. In *Proceedings of International Conference on Auditory Display*.

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.

GALLO, E., TSINGOS, N., ET AL. 2004. Efficient 3D audio processing on the GPU. In *ACM Workshop on General Purpose Computing on Graphics Processors*.

GERINGER, J. M., MACLEOD, R. B., AND SASANFAR, J. 2012. High school string players perception of violin, trumpet, and voice intonation. *String Research Journal 3*, 81–96.

GRIESINGER, D. 2009. The importance of the direct to reverberant ratio in the perception of distance, localization, clarity, and envelopment. In *Audio Engineering Society Convention 126*. Audio Engineering Society.

HERDER, J. 1999. Optimization of sound spatialization resource management through clustering. In *The Journal of Three Dimensional Images, 3D-Forum Society*. Vol. 13. 59–65.

ISO. 2012. ISO 3382, Acoustics—Measurement of room acoustic parameters. *International Standards Organisation* 3382.

JAMES, D. L., BARBIC, J., AND PAI, D. K. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *Proc. of ACM SIGGRAPH*. 987–995.

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

KULP, B. D. 1988. Digital equalization using fourier transform techniques. In *Audio Engineering Society Convention 85*. Audio Engineering Society.

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

LENTZ, T., SCHRÖDER, D., VORLÄNDER, M., AND ASSENMACHER, I. 2007. Virtual reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Advances in Singal Processing 2007*, 187–187.

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.

MOECK, T., BONNEEL, N., TSINGOS, N., DRETTAKIS, G., VIAUD-DELMON, I., AND ALLOZA, D. 2007. Progressive perceptual audio rendering of complex scenes. In *Proceedings of Symposium on Interactive 3D graphics and games*. ACM, 189–196.

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

PELZER, S. AND VORLÄNDER, M. 2010. Frequency-and time-dependent geometry for real-time auralizations. In *Proceedings of 20th International Congress on Acoustics, ICA*.

PULKKI, V. 1997. Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society 45,* 6, 456–466.

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.

RINDEL, J. H. AND CHRISTENSEN, C. L. 2003. Room acoustic simulation and auralization–how close can we get to the real room? In *Proc. 8th Western Pacific Acoustics Conference, Melbourne*.

RUNGTA, A., SCHISSLER, C., MEHRA, R., MALLOY, C., LIN, M., AND MANOCHA, D. 2016. Syncopation: Interactive synthesis-coupled sound propagation. *IEEE transactions on visualization and computer graphics*.

SAVIOJA, L., HUOPANIEMI, J., LOKKI, T., AND VÄÄNÄNEN, R. 1999. Creating interactive virtual acoustic environments. *Journal of the Audio Engineering Society (JAES) 47,* 9 (September), 675–705.

SAVIOJA, L., LOKKI, T., AND HUOPANIEMI, J. 2002. Auralization applying the parametric room acoustic modeling technique - the diva auralization system. *8th Int. Conf. on Auditory Display*, 219–224.

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 (SIGGRAPH 2014) 33,* 4, 39.

SCHISSLER, C., NICHOLLS, A., AND MEHRA, R. 2016. Efficient hrtf-based spatial audio for area and volumetric sources. *IEEE transactions on visualization and computer graphics*.

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

SCHROEDER, M. R. 1962. Natural sounding artificial reverberation. *Journal of the Audio Engineering Society 10,* 3, 219–223.

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.

SILTANEN, S., LOKKI, T., SAVIOJA, L., AND LYNGE CHRISTENSEN, C. 2008. Geometry reduction in room acoustics modeling. *Acta Acustica united with Acustica 94,* 3, 410–418.

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.

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

TSINGOS, N. 2001. A versatile software architecture for virtual audio simulations. In *International Conference on Auditory Display (ICAD)*. Espoo, Finland.

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

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., 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., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. *ACM Trans. Graph. 23,* 3, 249–258.

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

WAND, M. AND STRASSER, W. 2004. Multi-resolution sound rendering. In *SPBG'04 Symposium on Point - Based Graphics 2004*. 3–11.

WANG, L. M., RATHSAM, J., AND RYHERD, S. R. 2004. Interactions of model detail level and scattering coefficients in room acoustic computer simulation. In *International Symposium on Room Acoustics: Design and Science*.

WEFERS, F. AND VORLÄNDER, M. 2014. Efficient time-varying FIR filtering using crossfading implemented in the DFT domain. In *Forum Acousticum, Krakow, Poland*. European Acoustics Association.

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.

YEH, H., MEHRA, R., REN, Z., ANTANI, L., MANOCHA, D., AND LIN, M. 2013. Wave-ray coupling for interactive sound propagation in large complex scenes. *ACM Trans. Graph. 32,* 6, 165:1–165:11.

YOON, S., GOBBETTI, E., KASIK, D., AND MANOCHA, D. 2008. *Real-Time Massive Model Rendering*. Morgan and Claypool Publishers.