RESound: Interactive Sound Rendering for Dynamic Virtual Environments*

Micah T. Taylor
University of North Carolina
taylormt@cs.unc.edu

Anish Chandak
University of North Carolina
achandak@cs.unc.edu

Lakulish Antani
University of North Carolina
lakulish@cs.unc.edu

Dinesh Manocha
University of North Carolina
dm@cs.unc.edu

ABSTRACT

We present an interactive algorithm and system (RESound) for sound propagation and rendering in virtual environments and media applications. RESound uses geometric propagation techniques for fast computation of propagation paths from a source to a listener and takes into account specular reflections, diffuse reflections, and edge diffraction. In order to perform fast path computation, we use a unified ray-based representation to efficiently trace discrete rays as well as volumetric ray-frusta. RESound further improves sound quality by using statistical reverberation estimation techniques. We also present an interactive audio rendering algorithm to generate spatialized audio signals. The overall approach can handle dynamic scenes with no restrictions on source, listener, or obstacle motion. Moreover, our algorithm is relatively easy to parallelize on multi-core systems. We demonstrate its performance on complex game-like and architectural environments.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*modeling, systems*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*ray-tracing*

General Terms

Performance

Keywords

Acoustics, sound, ray tracing

*Project webpage:
`http://gamma.cs.unc.edu/Sound/RESound/`

MM'09, October 19–24, 2009, Beijing, China.

1. INTRODUCTION

Extending the frontier of visual computing, an *auditory display* uses sound to communicate information to a user and offers an alternative means of visualization or media. By harnessing the sense of hearing, sound rendering can further enhance a user's experience in multimodal virtual worlds [10, 27]. In addition to immersive environments, auditory display can provide a natural and intuitive human-computer interface for many desktop or handheld applications (see Figure 1). Realistic sound rendering can directly impact the perceived realism of users of interactive media applications. An accurate acoustic response for a virtual environment is attuned according to the geometric representation of the environment. This response can convey important details about the environment, such as the location and motion of objects. The most common approach to sound rendering is a two-stage process:

- **Sound propagation:** the computation of impulse responses (IRs) that represent an acoustic space.
- **Audio rendering:** the generation of spatialized audio signal from the impulse responses and dry (anechoically recorded or synthetically generated) source signals.

Sound propagation from a source to a listener conveys information about the size of the space surrounding the sound source and identifies the source to the listener even when the source is not directly visible. This considerably improves the immersion in virtual environments. For instance, in a first-person shooter game scenario (see Figure 1(b)), the distant cries of a monster coming around a corner or the soft steps of an opponent approaching from behind can alert the player and save them from fatal attack. Sound propagation is also used for acoustic prototyping (see Figure 1(d)) for computer games, complex architectural buildings, and urban scenes. Audio rendering also provides sound cues which give directional information about the position of the sound source relative to a listener. The cues are generated for headphones or a 3D surround sound speaker system. Thus, the listener can identify the sound source even when the sound source is out of the field of view of the listener. For example, in a VR combat simulation (see Figure 1(a)), it is critical to simulate the 3D sounds of machine guns, bombs, and missiles. Another application of 3D audio is user interface design, where sound cues are used to search for data on a multi-window screen (see Figure 1(c)).

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

Figure 1: Multimedia applications that need interactive sound rendering (a) Virtual reality training: Virtual Iraq simulation to treat soldiers suffering from post-traumatic stress disorder (top) and emergency training for medical personnel using Second Life (bottom). (b) Games: Half-life 2 (top) and Crackdown, winner of the best use of audio at the British Academy of Film and Television Arts awards (bottom). (c) Interfaces and Visualization: Multimodal interfaces (top) and data exploration & visualization system (bottom). (d) Computer aided Design: Game level design (top) and architectural acoustic modeling (bottom).

The main computational cost in sound rendering is the real-time computation of the IRs based on the propagation paths from each source to the listener. The IR computation relies on the physical modeling of the sound field based on an accurate description of the scene and material properties. The actual sensation of sound is due to small variations in the air pressure. These variations are governed by the three-dimensional *wave equation*, a second-order linear partial differential equation, which relates the temporal and spatial derivatives of the pressure field [40]. Current numerical methods used to solve the wave equation are limited to static scenes and can take minutes or hours to compute the IRs. Moreover, computing a numerically accurate solution, especially for high frequencies, is considered a very challenging problem.

Main Results: We present a system (RESound) for interactive sound rendering in complex and dynamic virtual environments. Our approach is based on geometric acoustics, which represents acoustic waves as rays. The geometric propagation algorithms model the sound propagation based on rectilinear propagation of waves and can accurately model the early reflections (up to $4-6$ orders). Many algorithms have been proposed for interactive geometric sound propagation using beam tracing, ray tracing or ray-frustum tracing [5, 14, 37, 40]. However, they are either limited to static virtual environments or can only handle propagation paths corresponding to specular reflections.

In order to perform interactive sound rendering, we use fast techniques for sound propagation and audio rendering. Our propagation algorithms use a hybrid ray-based representation that traces discrete rays [18] and ray-frusta [24]. Discrete ray tracing is used for diffuse reflections and frustum tracing is used to compute the propagation paths for specular reflections and edge diffraction. We fill in the late reverberations using statistical methods. We also describe an audio rendering pipeline combining specular reflections, diffuse reflections, diffraction, 3D sound, and late reverberation.

Our interactive sound rendering system can handle models consisting of tens of thousands of scene primitives (e.g. triangles) as well as dynamic scenes with moving sound sources, listener, and scene objects. We can perform interactive sound propagation including specular reflections, diffuse reflections, and diffraction of up to 3 orders on a multi-core PC. To the best of our knowledge, RESound is the first interactive sound rendering system that can perform plausible sound propagation and rendering in dynamic virtual environments.

Organization: The paper is organized as follows. We review the related methods on acoustic simulation in Section 2. Section 3 provides an overview of RESound and highlights the various components. We present the underlying representations and fast propagation algorithms in Section 4. The reverberation estimation is described in Section 5 and the audio rendering algorithm is presented in Section 6. The performance of our system is described in Section 7. In Section 8, we discuss the quality and limitations of our system.

2. PREVIOUS WORK

In this section, we give a brief overview of prior work in acoustic simulation. Acoustic simulation for virtual environment can be divided into three main components: sound synthesis, sound propagation, and audio rendering. In this paper, we only focus on interactive sound propagation and audio rendering.

2.1 Sound Synthesis

Sound synthesis generates audio signals based on interactions between the objects in a virtual environment. Synthesis techniques often rely on physical simulators to generate the forces and object interactions [7, 30]. Many approaches have been proposed to synthesize sound from object interaction using offline [30] and online [32, 47, 48] computations. Anechoic signals in a sound propagation engine can be replaced by synthetically generated audio signal as an input. Thus, these approaches are complementary to the presented work and could be combined with RESound for an improved immersive experience.

2.2 Sound Propagation

Sound propagation deals with modeling how sound waves propagate through a medium. Effects such as reflections, transmission, and diffraction are the important components. Sound propagation algorithms can be classified into two approaches: numerical methods and geometric methods.

Numerical Methods: These methods [6, 19, 26, 29] solve the wave equation numerically to perform sound propagation. These methods can provide very accurate results but are computationally expensive. Despite recent advances [31], these methods are too slow for interactive applications, and only limited to static scenes.

Geometric Methods: The most widely used methods for interactive sound propagation in virtual environments are based on geometric acoustics. They compute propagation paths from a sound source to the listener and the corresponding impulse response from these paths. Specular reflections of sound are modeled with the image-source method [2, 34]. Image-source methods recursively reflect the source point about all of the geometry in the scene to find specular reflection paths. BSP acceleration [34] and beam tracing [13, 21] have been used to accelerate this computation in static virtual environments. Other methods to compute specular paths include ray tracing based methods [18, 49] and approximate volume tracing methods [5, 23].

There has also been work on complementing specular reflections with diffraction effects. Diffraction effects are very noticeable at corners, as the diffraction causes the sound wave to propagate in regions that are not directly visible to the sound source. Two diffraction models are commonly used: the Uniform Theory of Diffraction (UTD) [17] and a recent formulation of the Biot-Tolstoy-Medwin method [41]. The BTM method is more costly to compute than UTD, and has only recently been used in interactive simulation [35]. The UTD, however, has been adapted for use in several interactive simulations [3, 42, 44].

Another important effect that can be modeled with GA is diffuse reflections. Diffuse reflections have been shown to be important for modeling sound propagation [9]. Two common existing methods for handling diffuse reflections are radiosity based methods [37, 38] and ray tracing based methods [8, 16].

The GA methods described thus far are used to render the early reflections. The later acoustic response must also be calculated [15]. This is often done through statistical methods [12] or ray tracing [11].

2.3 Audio Rendering

Audio rendering generates the final audio signal which can be heard by a listener over the headphones or speakers [20]. In context of geometric sound propagation, it involves convolving the impulse response computed by the propagation algorithm with an anechoic input audio signal and introduce 3D cues in the final audio signal to simulate the direction of incoming sound waves. In a dynamic virtual environment, sound sources, listener, and scene objects may be moving. As a result, the impulse responses change frequently and it is critical to generate an artifact-free smooth audio signal. Tsingos [43] and Wenzel et al. [52] describe techniques for artifact-free audio rendering in dynamic scenes. Introducing 3D cues in the final audio signals requires convolution of an incoming sound wave with a Head Related Impulse Response (HRIR) [1, 22]. This can only be performed for a few sound sources in real-time. Recent approaches based on audio perception [28, 46] and sampling of sound sources [51] can handle 3D sound for thousands of sound sources.

(a) (b) (c)

Figure 3: Example scene showing (a) specular, (b) diffraction, and (c) diffuse propagation paths.

3. SYSTEM OVERVIEW

In this section, we give an overview of our approach and highlight the main components. RESound simulates the sound field in a scene using geometric acoustics (GA) methods.

3.1 Acoustic modeling

All GA techniques deal with finding propagation paths between each source and the listener. The sound waves travel from a source (e.g. a speaker) and arrive at a listener (e.g. a user) by traveling along multiple propagation paths representing different sequences of reflections, diffraction, and refractions at the surfaces of the environment. Figure 3 shows an example of such paths. In this paper, we limit ourselves to reflections and diffraction paths. The overall effect of these propagation paths is to add reverberation (e.g. echoes) to the dry sound signal. Geometric propagation algorithms need to account for different wave effects that directly influence the response generated at the listener.

When a small, point like, sound source generates non-directional sound, the pressure wave expands out in a spherical shape. If the listener is set a short distance from the source, the wave field eventually encounters the listener. Due to the spreading of the field, the amplitude at the listener is attenuated. The corresponding GA component is a direct path from the source to the listener. This path represents the sound field that is diminished by distance attenuation.

As the sound field propagates, it is likely that the sound field will also encounter objects in the scene. These objects may reflect or otherwise scatter the waves. If the object is large relative to the field's wavelength, the field is reflected specularly, as a mirror does for light waves. In GA, these paths are computed by enumerating all possible reflection paths from the source to the listener, which can be a very costly operation. There has been much research focused on reducing the cost of this calculation [14], as most earlier methods were limited to static scenes with fixed sources. The delay and attenuation of these contributions helps the listener estimate the size of the propagation space and provides important directional cues about the environment.

Objects that are similar in size to the wavelength may also be encountered. When a sound wave encounters such an object, the wave is influenced by the object. We focus on two such scattering effects: edge diffraction and diffuse reflection.

Diffraction effects occur at the edges of objects and cause the sound field to be scattered around the edge. This scattering results in a smooth transition as a listener moves around edges. Most notably, diffraction produces a smooth transi-

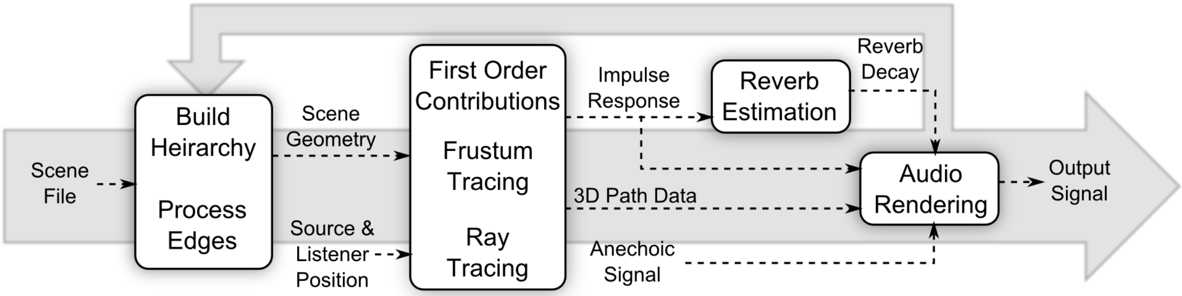


Figure 2: The main components of RESound: scene preprocessing; geometric propagation for specular, diffuse, and diffraction components; estimation of reverberation from impulse response; and final audio rendering.

tion when the line-of-sight between the source and listener is obstructed. The region behind an edge in which the diffraction field propagates is called the *shadow region*.

Surfaces that have fine details or roughness of the same order as the wavelength can diffusely reflect the sound wave. This means that the wave is not specularly reflected, but reflected in a *Lambertian* manner, such that the reflected direction is isotropic. These diffuse reflections complement the specular components [9].

As the sound field continues to propagate, the number of reflections and scattering components increase and the amplitude of these components decrease. The initial orders (e.g. up to four or six) of reflection are termed *early reflections*. These components have the greatest effect on a listener's ability to spatialize the sound. However, the early components are not sufficient to provide an accurate acoustic response for any given scene. The later reverberation effects are a function of the scene size [12] and convey an important sense of space.

3.2 Ray-based Path Tracing

RESound uses a unified ray representation for specular reflections, diffuse reflections, and diffraction path computations. The underlying framework exploits recent advances in interactive ray tracing in computer graphics literature. We compute diffuse reflections using a discrete ray representation [25, 50] and specular reflections and diffraction using a ray-frustum representation [5, 24]. A frustum is a convex combination of four corner rays [24]. We use fast ray tracing algorithms to perform intersection tests for the discrete rays as well as volumetric frusta.

We assume that the scene is composed of triangles and is represented using a bounding volume hierarchy (BVH) of axis-aligned bounding boxes (AABBs). A BVH can be used to handle dynamic scenes efficiently [25]. The same underlying hierarchy is used for both discrete rays and ray-frusta as part of our unified representation. Rays are shot as ray packets [25] and efficient frustum culling is used for fast intersection of ray packets and frusta with the BVH. In order to perform fast intersection tests with scene triangles the frustum representation uses Plücker coordinates [36].

3.3 RESound Components

Our system consists of three main processing steps. These are outlined in Figure 2.

Preprocessing: As part of preprocessing, a scene bounding volume hierarchy is created. This is a hierarchy of axis-aligned bounding boxes and is updated when the objects in the scene move. This hierarchy is used to perform fast intersection tests for discrete ray and frustum tracing. The edges of objects in the scene are also analyzed to determine appropriate edges for diffraction.

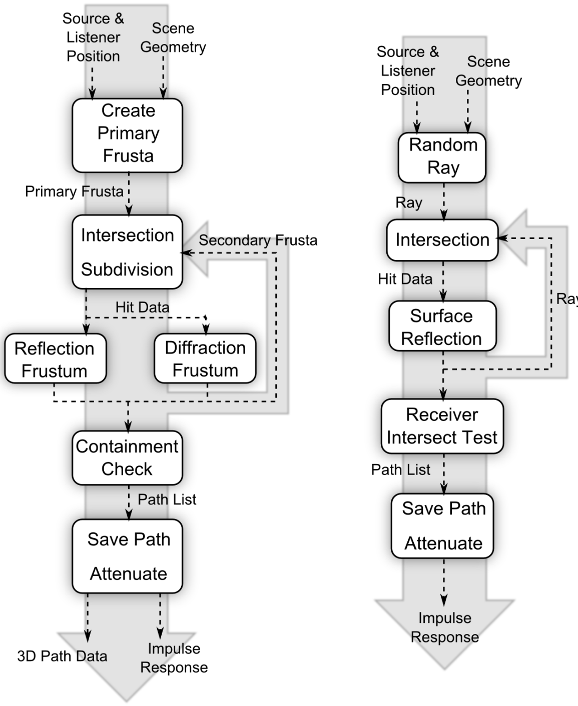
Interactive Sound Propagation: This stage computes the paths between the source and the listener. The direct path is quickly found by checking for obstruction between the source and listener. A volumetric frustum tracer is used to find the specular and edge diffraction paths. A stochastic ray tracer is used to compute the diffuse paths. These paths are adjusted for frequency band attenuation and converted to appropriate pressure components.

Audio Rendering: After the paths are computed, they need to be auralized. A statistical reverberation filter is estimated using the path data. Using the paths and the estimated filter as input, the waveform is attenuated by the auralization system. The resulting signal represents the acoustic response and is output to the system speakers.

4. INTERACTIVE SOUND PROPAGATION

In this section, we give an overview of our sound propagation algorithm. Propagation is the most expensive step in the overall sound rendering pipeline. The largest computational cost is the calculation of the acoustic paths that the sound takes as it is reflected or scattered by the objects in the scene. Under the assumption of geometric acoustics, this is primarily a visibility calculation. Thus, we have chosen rays as our propagation primitive. For example, the direct sound contribution is easily modeled by casting a ray between the source and listener. If the path is not obstructed, there is a direct contribution from the source to the listener. The other propagation components are more expensive to compute, but rely on similar visibility computations.

When computing the propagation components, many intersection computations between the scene triangles and the ray primitives are performed. In order to reduce the computation time, we would like to minimize the cost of the intersection tests. Since our propagation method is ray based, an acceleration structure to minimize ray intersections against scene geometry can be used. Specifically, our system constructs a bounding volume hierarchy (BVH) of axis aligned bounding boxes [25]. This structure can be updated for dynamic scene objects with refitting algorithms. Also, we mark all possible diffraction edges. This allows the diffraction propagation to abort early if the scene edge is not marked as a diffracting edge.



(a) (b)

Figure 4: Unified ray engine: Both (a) frustum tracing and (b) ray tracing share a similar rendering pipeline.

4.1 Specular paths

We use volumetric frustum tracing [24] to calculate the specular paths between the source and listener. From our basic ray primitive, we form a convex volume bounded by 4 rays. In order to model a uniform point sound source, we cast many of these frustum primitives such that all the space around the source is covered. For each frustum, the bounding rays of the volume are intersected with the scene primitives. After the rays have hit the geometric primitives, they are specularly reflected. This gives rise to another frustum that is recursively propagated. This continues until a specified order of reflection is achieved.

However, it is possible that the 4 bounding rays of the frustum did not all hit the same object in the scene. In this case, it cannot be guaranteed that the resulting specular frustum correctly contains the reflection volume. As such, we employ an adaptive subdivision strategy [5] to reduce the error in the volume. If it is found that the 4 rays do not intersect the same geometric primitive, that is, the frustum face is not fully contained within the bounds of the geometric primitive, the frustum is subdivided using a quad-tree like structure into 4 sub-frusta. The sub-frusta are then intersected with the scene and the subdivision process continues until a user-defined subdivision level is reached. When the subdivision is complete, any ambiguous intersections are resolved by choosing the closest intersected object and reflecting the subdivided frustum's rays against it. This process results in a reasonably [5] accurate volumetric covering of the scene space.

Given any propagation frusta, if the listener is contained within the volume, there must exist some sound path from the source to the listener. This path is verified by casting a ray from the listener towards the frustum origin. If the ray intersection point is contained in the frustum origin face on the triangle, the path segment is valid. This validation process is repeated using the computed intersection point to the origin of the previous frustum. If the entire path is valid, the path distance and attenuation are recorded. Figure 4(a) shows an overview of the frustum engine.

4.2 Edge Diffraction paths

Frustum tracing can be modified to account for diffraction contributions [42] using the Uniform Theory of Diffraction (UTD). The UTD can be used to calculate the diffraction attenuation for ray paths used in GA. When a sound ray encounters an edge, the ray is scattered about the edge. In the UTD formulation, the region covered by the diffraction contribution is defined by the angle of the entrance ray. If a ray hits the edge with an angle of θ, the ray is scattered about the edge in a cone shape where the cone makes an angle θ with the edge.

As the frusta intersect the scene triangles, the triangle edges are checked whether they are marked as diffracting edges. If the triangle has diffracting edges, and the edges are contained within the frustum face, a new diffraction frustum is created. Similar to other approaches [42, 44], we compute the diffraction component only in the shadow region. As such, the frustum is bounded by the line-of-sight from the frustum origin and the far side of the triangle. This frustum then propagates through the scene as normal.

The final sound path is verified using the same process described for specular paths. However, for diffraction sequences, the path is attenuated using the UTD equation [17]. The UTD equation is in the frequency domain, and is thus computed for a number of frequency bands. The resulting UTD coefficients are combined with the attenuation for the other path segments to create the final path attenuation.

4.3 Diffuse component

In order to compute sound reflected off diffuse materials, we use a stochastic ray tracer (Figure 4(b)). Rays are propagated from the sound source in all the directions. When a ray encounters a triangle it is reflected and tracing continues. The reflection direction is determined by the surface material. The listener is modeled by a sphere that approximates the listener's head. As the rays propagate, we check for intersections with this sphere. If there is an intersection, the path distance and the surfaces encountered are recorded for the audio rendering step.

The scattering coefficient for surface materials varies for different sound frequencies. Thus, for one frequency incoming rays may be heavily scattered, while eor another frequency the reflection is mostly specular. Since intersecting rays with the objects in the scene is a costly operation, we wish to trace rays only once for all the frequencies. As such, for each ray intersection, we randomly select between diffuse and specular reflection [11].

If the ray hit the listener, we scale the energy for each frequency band appropriately based on the material properties and type of reflections selected. If a path is found to be composed entirely of specular reflections, it is discarded as such paths are found in the frustum tracing step. Once all paths have been computed and attenuated, the resulting values are converted to a histogram which combines nearby

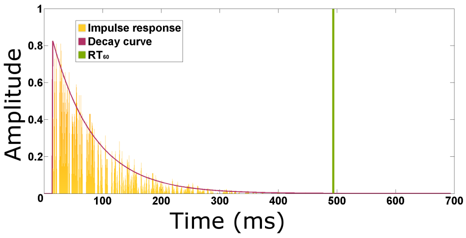


Figure 5: Extrapolating the IR to estimate late reverberation: The red curve is obtained from a least-squares fit (in log-space) of the energy IR computed by GA, and is used to add the reverberant tail to the IR.

contributions into single, larger contributions. The energy for each contribution is reduced based on the number of rays that have been propagated. The square root of the each contribution is used to compute a final pressure value.

5. REVERBERATION ESTIMATION

The propagation paths computed by the frustum tracer and stochastic ray tracer described in Section 4 are used only for the early reflections that reach the listener. While they provide important perceptual cues for spatial localization of the source, capturing late reflections (reverberation) contributes significantly to the perceived realism of the sound simulation.

We use well-known statistical acoustics models to estimate the reverberant tail of the energy IR. The Eyring model [12] is one such model that describes the energy decay within a single room as a function of time:

$$E(t) = E_0 e^{\frac{cS}{4V} t \log(1-\alpha)} \tag{1}$$

where c is the speed of sound, S is the total absorbing surface area of the room, V is the volume of the room and α is the average absorption coefficient of the surfaces in the room.

Given the energy IR computed using GA, we perform a simple linear least-squares fit to the IR in log-space. This gives us an exponential curve which fits the IR and can easily be extrapolated to generate the reverberation tail. From the curve, we are most interested in estimating the RT_{60}, which is defined as the time required for the energy to decay by 60 dB. Given the slope computed by the least-squares fit of the IR data, it is a simple matter to estimate the value of RT_{60}. This value is used in the audio rendering step to generate late reverberation effects.

Note that Equation (1) is for a single-room model, and is not as accurate for scenes with multiple rooms (by "rooms" we mean regions of the scene which are separated by distinct apertures, such as doors or windows). The single-room model is a good approximation for large interior spaces and many outdoor scenes. Other models exist for coupled rooms [39], but they would require fitting multiple curves to the IR, and the number of curves to fit would depend on the number of rooms in the scene. In the interests of speed and simplicity, we have chosen to use a single-room model.

6. AUDIO RENDERING

Audio rendering is the process of generating an audio signal which can be heard by a listener using headphones or speakers. In this section, we provide details on the real-time audio rendering pipeline implemented in our interactive sound propagation system. Our audio rendering pipeline is implemented using XAudio2[1], a cross-platform audio library for Windows and Xbox 360.

Our sound propagation algorithm generates a list of specular, diffuse, and diffracted paths from each source to the listener. These paths are accessed asynchronously by the audio rendering pipeline as shown in Figure 6 at different rates. Furthermore, each path can be represented as a virtual source with some attenuation, distance from the listener, and the incoming direction relative to the listener. The direction of a virtual source relative to the listener is simulated by introducing 3D sound cues in the final audio. Additionally, the source, listener, and scene objects can move dynamically. In such cases, the impulse response (IR) computed during the sound propagation step could vary significantly from one frame to another. Thus, our approach mitigates the occurence of artifacts by various means. Our system also uses the previously described reverberation data to construct the appropriate sound filters.

6.1 Integration with Sound Propagation

The paths computed by the sound propagation algorithm in Section 4 are updated at different rates for different orders of reflection. These paths are then queried by the audio rendering system in a thread safe manner. To achieve a high quality final audio signal, the audio rendering system needs to query at the sampling rate of the input audio signal (44.1 KHz). However, our audio rendering system queries per audio frame. We have found frames containing 10ms worth of audio samples suitable. Various user studies support that a lower update rate [33] can be used without any perceptual difference. It should be noted that the direct sound component and the early reflection components are very fast to compute. Thus, we update the direct contribution and first order reflections at a higher rate than the other components. For the direct and first order reflection paths, we also introduce 3D sound cues in the final audio signal. To produce the final audio we band pass the input signal into eight octave bands. For each octave band we compute an impulse response, which is convolved with the band pass input audio to compute final audio as shown in Figure 7. The details on computing an impulse response using the paths from the sound propagation engine are below.

Specular and Diffraction IR: The specular reflections and diffraction are formulated as a function of the sound pressure, as described in the previous sections. Thus, any path reaching from a source to the listener has a delay computed as d/C where d is the distance traveled, and C is the speed of sound. Each impulse is attenuated based on frequency dependent wall absorption coefficients and the distance traveled. For all the paths reaching from a source to the listener, a value with attenuation A_{path} is inserted at time index d/C in the impulse response. One such impulse response is computed for all different octave bands for a source-listener pair.

Diffuse IR: The diffuse reflections are formulated as a function of the energy of the sound waves. Using the paths collected at the listener, an energy IR is constructed for all the reflection paths reaching the listener This energy IR is

[1] http://msdn.microsoft.com/en-us/library/bb694503(VS.85).aspx

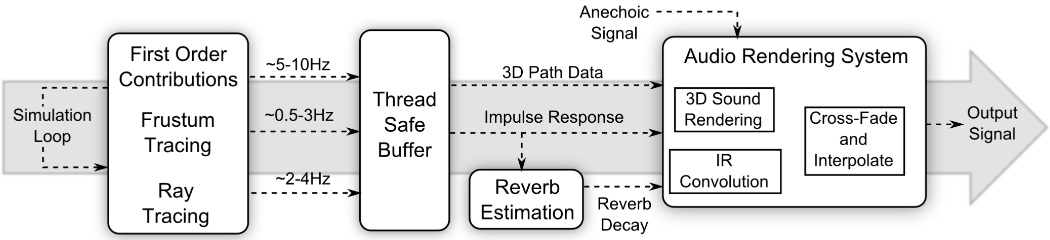


Figure 6: An overview of the integration of audio rendering system with the sound propagation engine. Sound propagation engine updates the computed paths in a thread safe buffer. The direct path and first order reflection paths are updated at higher frequency. The audio rendering system queries the buffer and performs 3D audio for direct and first order paths and convolution for higher order paths. The cross-fading and interpolation components smooth the final audio output signal.

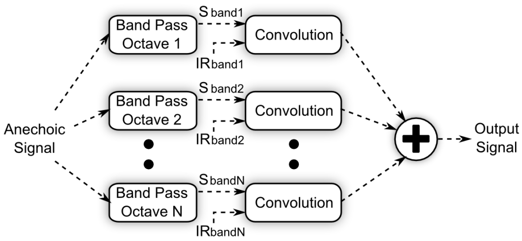


Figure 7: IR Convolution: The input audio signal S is band passed into N octave bands which are convolved with the IR of the corresponding band.

converted into pressure IR for audio rendering. We take the square root of energy response to create a pressure IR for each frequency band. This IR is combined with specular and diffraction IRs to produce the final IR used in the audio rendering.

6.2 Issues with Dynamic Scenes

Our sound propagation system is general and can handle moving sources, moving listener, and dynamic geometric primitives. This introduces a unique set of challenges for our real-time audio rendering system. Due to the motion of the sources, listener, and scene objects, the propagation paths could change dramatically and producing artifact-free audio rendering can be challenging. Therefore, we impose physical restrictions on the motion of sources, listener, and the geometric primitives to produce artifact-free audio rendering. To further mitigate the effects of the changing IRs, we convolve each audio frame with the current and the previous IRs and crossfade them to produce the final audio signal. The window of cross-fading can be adjusted to minimize the artifacts due to motion. Other more sophisticated approaches like predicting the positions and velocities of source or the listener can also be used [43, 52].

6.3 3D Sound Rendering

In a typical sound simulation, many sound waves reach the listener from different directions. These waves diffract around the listener's head and provide cues regarding the direction of the incoming wave. This diffraction effect can be encoded in a Head-Related Impulse Response (HRIR) [1]. Thus, to produce a realistic 3D sound rendering effect, each incoming path to the listener can be convolved with an HRIR. However, for large numbers of contributions this computation can quickly become expensive and it may not be possible to perform audio rendering in real-time. Thus, only direct and first order reflections are convolved with a normalized HRIR [1]. Some recent approaches have been proposed to handle audio rendering of large numbers of sound sources [45, 51]. These approaches can also be integrated with our system.

6.4 Adding Late Reverberation

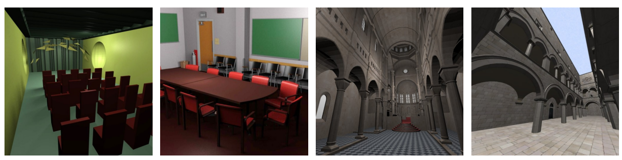
XAudio2 supports the use of user-defined filters and other audio processing components through the XAPO interface. One of the built-in filters is an artificial reverberation filter, which can add late decay effects to a sound signal. This filter can be attached to the XAudio2 pipeline (one filter per band) to add late reverberation in a simple manner.

The reverberation filter has several configurable parameters, one of which is the RT_{60} for the room. In Section 5, we described a method for estimating this value. The reverberation filter is then updated with the estimate. This approach provides a simple, efficient way of complementing the computed IRs with late reverberation effects.

7. PERFORMANCE

Our system makes use of several levels of parallel algorithms to accelerate the computation. Ray tracing is known to be a highly parallelizable algorithm and our system threads to take advantage of multi-core computers. Also, frustum tracing uses vector instructions to perform operations on a frustum's corner rays in parallel. Using these optimizations, our system achieves interactive performance on common multi-core PCs.

In this section, we detail the performance of RESound. We highlight each subsystem's performance on a varying set of scenes. The details of the scenes and system performance are presented in Table 1, and the scenes are visually shown in Figure 8. In all benchmarks, we run RESound using a multi-core PC at 2.66Ghz; the number of threads per component is described in each section.



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

Figure 8: Test scenes used: (a) Room, (b) Conference, (c) Sibenik, and (d) Sponza.

		Specular + diffraction (3 orders)			Specular + diffraction (1 order)			Diffuse (3 orders)	
Scene	Triangles	Time	Frusta	Paths	Time	Frusta	Paths	Time	Paths
Room	6k	359ms	278k	4	77ms	7k	3	274ms	228
Conference	282k	1137ms	320k	7	157ms	5k	2	323ms	318
Sibenik	76k	2810ms	900k	14	460ms	10k	5	437ms	26
Sponza	66k	1304ms	598k	8	260ms	10k	3	516ms	120

Table 1: Performance: Test scene details and the performance of the RESound components.

# Impulses	Compute time (ms)
10	0.026
50	0.111
100	0.425
1000	37.805
5000	1161.449

Table 2: Timings for late reverberation estimation.

Specular and Diffraction: We generate two separate IRs using frustum tracing. One IR includes only the first order specular and diffraction contributions. Since these paths are fast to compute, we devote one thread to this task. The other IR we generate includes the contributions for 3 orders of reflection and 2 orders of diffraction. This is done using 7 threads. The performance details for both simulations cycles are described in Table 1.

Diffuse tracing: Our diffuse tracer stochastically samples the scene space during propagation. As such, the rays are largely incoherent and it is difficult to use ray packets. Nonetheless, even when tracing individual rays, RESound can render at interactive rates as shown in the performance table. The timings are for 200k rays with 3 reflections using 7 threads.

Late reverberation: We measured the time taken by our implementation to perform the least-squares fitting while estimating late reverberation. The execution time was measured by averaging over 10 frames. During testing, we vary the density of the impulse response. The reverberation calculation is not threaded due to its minimal time cost. The results are summarized in Table 2.

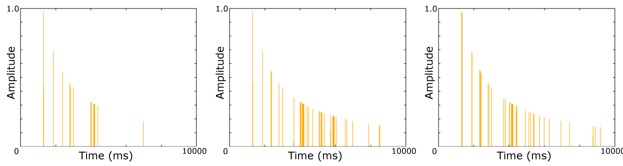
8. QUALITY AND LIMITATIONS

The algorithms used in RESound are based on the physical properties of high frequency acoustic waves. We discuss the output quality of each component in the RESound system and compare against the accurate known simulations. We also note the benefits that RESound offers over simpler audio rendering systems. The underlying limitations of the methods used are also discussed.

8.1 Quality

Since adaptive frustum tracing approximates the image source reflection model, its accuracy has been compared to image-source methods [5]. It was found that as the subdivision level increases, the number of contributions found by the frustum simulation approach the number found by the image-method. Moreover, the attenuation of the resulting impulse response from frustum tracing is similar to that found by image-source (Figure 9).

Similarly, the validity of diffraction using frustum tracing has also been compared to an accurate beam tracing system



(a) (b) (c)

Figure 9: Specular paths: With a subdivision (a) level of 2, frustum tracing finds 13 paths. A subdivision (b) level of 5 finds 40 paths. The (c) image-source solution has 44 paths.

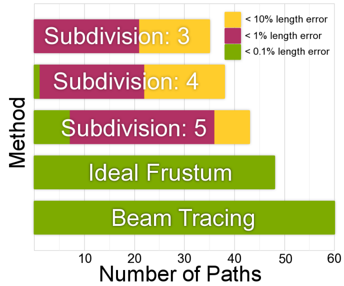


Figure 10: Diffraction paths: Increasing the frustum subdivision improves the diffraction accuracy.

with diffraction [42]. Due to limitations of frustum engines, it was found that certain types of diffraction paths could not be enumerated. However, as the frustum subdivision level was increased, the number of diffraction paths found approached an ideal solution (Figure 10) and the paths accurately matched the reference solution.

The diffuse IR in RESound is generated by stochastic ray tracing. The sampling and attenuation model RESound uses has previously been shown to be statistically valid with sufficient sampling. Detailed analysis and validation has been presented by Embrechts [11].

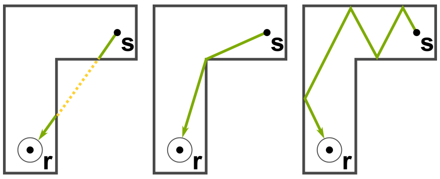
We compare our reverberation decay times to statistically estimated times in two simple scenes. Similar scenes are described in other work [15, 16]. The results are presented in Table 3.

8.2 Benefits

Interactive audio simulations used in current applications are often very simple and use precomputed reverberation effects and arbitrary attenuations. In RESound, the delays and attenuations for both reflection and diffraction are

Room size (m)	Absorption	Predicted	RESound
4x4x4	0.1	1030 ms	1170 ms
27.5x27.5x27.5	0.0	8890 ms	7930 ms

Table 3: Reverberation decay times for two models.



(a) (b) (c)

Figure 11: Path direction: (a) Binaural paths are physically impossible, but (b) diffraction and (c) reflection paths direct the listener as physically expected.

based on physical approximations. This allows RESound to generate acoustic responses that are expected given scene materials and layout.

In addition to calculating physically based attenuations and delays, RESound also provides accurate acoustic spatialization. When compared to simple binaural rendering, RESound provides more convincing directional cues. Consider a situation when the sound source is hidden from the listener's view (Figure 11). In this case, without reflection and diffraction, the directional component of the sound field appears to pass through the occluder. However, propagation paths generated by RESound arrive at the listener with a physically accurate directional component.

8.3 Limitations

RESound has several limitations. The accuracy of our algorithm is limited by the use of underlying GA algorithms. In practice, GA is only accurate for higher frequencies. Moreover, the accuracy of our frustum-tracing reflection and diffraction varies as a function of maximum subdivision. Our diffraction formulation is based on the UTD and assumes that the edge lengths are significantly larger than the wavelength. Also, frustum tracing based diffraction also is limited in the types of diffraction paths that can be found. Our approach for computing the diffuse IR is subject to statistical error [11] that must be overcome with dense sampling. In terms of audio rendering, we impose physical restrictions on the motion of the source, listener, and scene objects to generate an artifact free rendering.

9. CONCLUSION AND FUTURE WORK

We have presented an interactive sound rendering system for dynamic virtual environments. RESound uses GA methods to compute the propagation paths. We use a ray-based underlying representation that is used to compute specular reflections, diffuse reflections, and edge diffraction. We also use statistical late reverberation estimation techniques and present an interactive audio rendering algorithm for dynamic virtual environments. We believe RESound is the first interactive system that can generate plausible sound rendering in complex, dynamic virtual environments.

There are many avenues for future work. We would like to further analyze the accuracy of our approach. It is possible to further improve the accuracy of edge diffraction by using the BTM formulation, as opposed to UTD. Similarly, the accuracy of diffuse reflections can be improved based on better sampling methods. Many interactive applications such as games or VR need $30 - 60$ Hz update rates and we may need faster methods to achieve such a performance on current commodity hardware. We are also investigating using frustum tracing for very accurate GA simulations [4]. Finally, we would like to use RESound for other applications such as tele-conferencing and design of sound-based user interfaces.

10. ACKNOWLEDGMENTS

This research is supported in part by ARO Contract W911NF-04-1-0088, NSF award 0636208 , DARPA/RDECOM Contracts N61339-04-C-0043 and WR91CRB-08-C-0137, Intel, and Microsoft.

11. REFERENCES

[1] V. Algazi, R. Duda, and D. Thompson. The CIPIC HRTF Database. In *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[2] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, April 1979.

[3] F. Antonacci, M. Foco, A. Sarti, and S. Tubaro. Fast modeling of acoustic reflections and diffraction in complex environments using visibility diagrams. In *Proceedings of 12th European Signal Processing Conference (EUSIPCO '04)*, pages 1773–1776, September 2004.

[4] A. Chandak, L. Antani, M. Taylor, and D. Manocha. Fastv: From-point visibility culling on complex models. *Eurographics Symposium on Rendering*, 2009.

[5] A. Chandak, C. Lauterbach, M. Taylor, Z. Ren, and D. Manocha. AD-Frustum: Adaptive Frustum Tracing for Interactive Sound Propagation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1707–1722, Nov.-Dec. 2008.

[6] R. Ciskowski and C. Brebbia. *Boundary Element methods in acoustics.* Computational Mechanics Publications and Elsevier Applied Science, 1991.

[7] P. R. Cook. *Real Sound Synthesis for Interactive Applications.* A. K. Peters, 2002.

[8] B. Dalenbäck. Room acoustic prediction based on a unified treatment of diffuse and specular reflection. *The Journal of the Acoustical Society of America*, 100(2):899–909, 1996.

[9] B.-I. Dalenbäck, M. Kleiner, and P. Svensson. A Macroscopic View of Diffuse Reflection. *Journal of the Audio Engineering Society (JAES)*, 42(10):793–807, October 1994.

[10] N. Durlach and A. Mavor. *Virtual Reality Scientific and Technological Challenges.* National Academy Press, 1995.

[11] J. J. Embrechts. Broad spectrum diffusion model for room acoustics ray-tracing algorithms. *The Journal of the Acoustical Society of America*, 107(4):2068–2081, 2000.

[12] C. F. Eyring. Reverberation time in "dead" rooms. *The Journal of the Acoustical Society of America*, 1(2A):217–241, January 1930.

[13] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proc. of ACM SIGGRAPH*, pages 21–32, 1998.

[14] T. Funkhouser, N. Tsingos, and J.-M. Jot. Survey of Methods for Modeling Sound Propagation in Interactive Virtual Environment Systems. *Presence and Teleoperation*, 2003.

[15] M. Hodgson. Evidence of diffuse surface reflection in rooms. *The Journal of the Acoustical Society of America*, 88(S1):S185–S185, 1990.

[16] B. Kapralos, M. Jenkin, and E. Milios. Acoustic Modeling Utilizing an Acoustic Version of Phonon Mapping. In *Proc. of IEEE Workshop on HAVE*, 2004.

[17] R. G. Kouyoumjian and P. H. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proc. of IEEE*, 62:1448–1461, Nov. 1974.

[18] A. Krokstad, S. Strom, and S. Sorsdal. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration*, 8(1):118–125, July 1968.

[19] K. Kunz and R. Luebbers. *The Finite Difference Time Domain for Electromagnetics*. CRC Press, 1993.

[20] H. Kuttruff. *Acoustics*. Routledge, 2007.

[21] S. Laine, S. Siltanen, T. Lokki, and L. Savioja. Accelerated beam tracing algorithm. *Applied Acoustic*, 70(1):172–181, 2009.

[22] V. Larcher, O. Warusfel, J.-M. Jot, and J. Guyard. Study and comparison of efficient methods for 3-d audio spatialization based on linear decomposition of hrtf data. In *Audio Engineering Society 108th Convention preprints*, page preprint no. 5097, January 2000.

[23] C. Lauterbach, A. Chandak, and D. Manocha. Adaptive sampling for frustum-based sound propagation in complex and dynamic environments. In *Proceedings of the 19th International Congress on Acoustics*, 2007.

[24] C. Lauterbach, A. Chandak, and D. Manocha. Interactive sound rendering in complex and dynamic scenes using frustum tracing. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1672–1679, Nov.-Dec. 2007.

[25] C. Lauterbach, S. Yoon, D. Tuft, and D. Manocha. RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs. *IEEE Symposium on Interactive Ray Tracing*, 2006.

[26] J. Lehtinen. Time-domain numerical solution of the wave equation, 2003.

[27] R. B. Loftin. Multisensory perception: Beyond the visual in visualization. *Computing in Science and Engineering*, 05(4):56–58, 2003.

[28] T. Moeck, N. Bonneel, N. Tsingos, G. Drettakis, I. Viaud-Delmon, and D. Alloza. Progressive perceptual audio rendering of complex scenes. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 189–196, New York, NY, USA, 2007. ACM.

[29] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford University Press, 2003.

[30] J. F. O'Brien, C. Shen, and C. M. Gatchalian. Synthesizing sounds from rigid-body simulations. In *The ACM SIGGRAPH 2002 Symposium on Computer Animation*, pages 175–181. ACM Press, July 2002.

[31] N. Raghuvanshi, N. Galoppo, and M. C. Lin. Accelerated wave-based acoustics simulation. In *ACM Solid and Physical Modeling Symposium*, 2008.

[32] N. Raghuvanshi and M. C. Lin. Interactive sound synthesis for large scale environments. In *Symposium on Interactive 3D graphics and games*, pages 101–108, 2006.

[33] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen. Creating interactive virtual acoustic environments. *Journal of the Audio Engineering Society (JAES)*, 47(9):675–705, September 1999.

[34] D. Schröder and T. Lentz. Real-Time Processing of Image Sources Using Binary Space Partitioning. *Journal of the Audio Engineering Society (JAES)*, 54(7/8):604–619, July 2006.

[35] D. Schröder and A. Pohl. Real-time Hybrid Simulation Method Including Edge Diffraction. In *EAA Auralization Symposium*, Espoo, Finland, June 2009.

[36] K. Shoemake. Plücker coordinate tutorial. *Ray Tracing News*, 11(1), 1998.

[37] S. Siltanen, T. Lokki, S. Kiminki, and L. Savioja. The room acoustic rendering equation. *The Journal of the Acoustical Society of America*, 122(3):1624–1635, September 2007.

[38] S. Siltanen, T. Lokki, and L. Savioja. Frequency domain acoustic radiance transfer for real-time auralization. *Acta Acustica united with Acustica*, 95:106–117(12), 2009.

[39] J. E. Summers, R. R. Torres, and Y. Shimizu. Statistical-acoustics models of energy decay in systems of coupled rooms and their relation to geometrical acoustics. *The Journal of the Acoustical Society of America*, 116(2):958–969, August 2004.

[40] P. Svensson and R. Kristiansen. Computational modelling and simulation of acoustic spaces. In *22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, June 2002.

[41] U. P. Svensson, R. I. Fred, and J. Vanderkooy. An analytic secondary source model of edge diffraction impulse responses . *Acoustical Society of America Journal*, 106:2331–2344, Nov. 1999.

[42] M. Taylor, A. Chandak, Z. Ren, C. Lauterbach, and D. Manocha. Fast Edge-Diffraction for Sound Propagation in Complex Virtual Environments. In *EAA Auralization Symposium*, Espoo, Finland, June 2009.

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

[44] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proc. of ACM SIGGRAPH*, pages 545–552, 2001.

[45] N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. Technical Report RR-4734, INRIA, REVES/INRIA Sophia-Antipolis, Feb 2003.

[46] N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. *ACM Trans. Graph.*, 23(3):249–258, 2004.

[47] K. van den Doel. *Sound Synthesis for Virtual Reality and Computer Games*. PhD thesis, University of British Columbia, 1998.

[48] K. van den Doel, P. G. Kry, and D. K. Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 537–544, New York, NY, USA, 2001. ACM Press.

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

[50] I. Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004.

[51] M. Wand and W. Straßer. Multi-resolution sound rendering. In *SPBG'04 Symposium on Point - Based Graphics 2004*, pages 3–11, 2004.

[52] E. Wenzel, J. Miller, and J. Abel. A software-based system for interactive spatial sound synthesis. In *International Conference on Auditory Display (ICAD)*, Atlanta, GA, April 2000.