Interactive Sound Rendering

Dinesh Manocha | Ming C. Lin
Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599-3175
{dm,lin}@cs.unc.edu
http://gamma.cs.unc.edu/Sound

Abstract

Extending the frontier of visual computing, sound rendering utilizes auditory display to communicate information to a user and offers an alternative means of visualization. By harnessing the sense of hearing, sound rendering can further enhance a user’s experience in a multimodal virtual world. In addition to immersive environments, auditory displays can provide a natural and intuitive human-computer interface for many desktop applications. In this paper, we give a brief overview of recent work at UNC Chapel Hill on fast algorithms for sound synthesis and sound propagation. These include physically-based sound synthesis for rigid bodies and liquid sound generation, as well as numeric and geometric algorithms for sound propagation. We highlight their performance on different benchmarks and briefly discuss some problems for future research.

1. Introduction

Traditionally, the focus in interactive applications has been on generating realistic images. These developments have been well supported by high growth rates and programmability of current graphics hardware. At the same time it is also important to develop interactive algorithms for other forms of rendering, including sound and haptics. Sound rendering utilizes auditory display to communicate information to a user and offers an alternative means of visualization. By harnessing the sense of hearing, audio rendering can further enhance a user’s experience in a multimodal virtual world [17], [31]. Ultimately, the audio cues combined with visual rendering provide a far more immersive experience. In addition to immersive environments, auditory display can also provide a natural and intuitive human-computer interface for many desktop applications. Furthermore, audio interfaces are increasingly used to develop assistive technologies for the visually impaired. Despite the significance of hearing as one of the dominant senses, auditory displays have not received as much attention as graphical rendering in the literature.

The design of auditory display involves audio hardware technology, software rendering pipeline, modeling and simulation of acoustic spaces, signal processing, sonification techniques, perceptual evaluation, and application development. In this paper, we primarily focus on two compute-intensive components of auditory displays: sound synthesis and sound propagation. Sound synthesis deals with how sound is generated using physical principles [16], [55], [35], [40], [3], [34]. Sound propagation deals with computational modeling of acoustic spaces that takes into account the knowledge of sound sources, listener locations, 3D models of the environments, and material absorption data to compute impulse response (IR) of the space [19], [48]. These IRs are convolved with recorded or synthetically produced sound signals to generate spatialized sound effects, i.e. auralization.

The driving impetus of recent research in sound rendering has mainly come from interactive applications, including computer gaming, training systems, desktop interfaces, education, scientific visualization and computer-aided design. All these applications need capabilities for real-time sound synthesis and propagation techniques. Realistic sound quality can directly impact the perceived realism and full immersion of players and users of these interactive applications. Furthermore, interactive modeling and simulation of acoustic spaces can significantly enhance numerous scientific and engineering applications [11]. For examples, designers of man-made structures, such as urban layouts, habitats [54], and mechanical CAD structures [21], can benefit tremendously from interactive acoustic simulation technologies in terms of improved design and virtual prototyping. The current market for acoustic simulation software for engineering design alone is estimated to be around $300M a year.

The audible sonic frequencies for humans range from 20 Hz to 20,000 Hz with wavelengths that fall exactly in the range of dimensions of common objects, i.e. a few centimeters to a few meters. Consequently sound diffracts (bends) appreciably, especially at low frequencies. Due to its low speed, sound reflections is also easily perceptible in a room in terms of delays and echos. The combination of sound diffraction and reflections makes interactive sound propagation a major computational challenge [48]. As a result, most interactive applications currently use sound effects based on fixed models of propagation or statically designed environment reverberation filers.

In terms of synthesis, most interactive applications use pre-recorded sound clips for providing sounds corresponding to object interactions in a scene. Although this
approach has the advantages of sounds being realistic and the generation process is quite fast, there are many physical effects which cannot be captured by such a technique. The resulting sound often lacks variation in tones and timbers or typically appears repetitive. Physics-based sound synthesis can reproduce sound using physical principles, adding noticeable realism by introducing natural variations due to object interaction, but poses substantial computational challenges for interactive applications.

Challenges in Sound Rendering: The current systems for sound synthesis and propagation can not offer interactive performance in complex, dynamic scenes or may not be able to synthesize different type of sounds. We need significant advances and also need to integrate the resulting algorithms with different applications. These include:

- **Sound Synthesis:** Most of existing synthesis approaches have focused on sound mixing or sound generation due to colliding solids in the air. We need to develop new algorithms for interactive sound synthesis techniques for complex environments consisting of hundreds of objects and sound generated in liquid medium and coupled fluid-solid interaction.

- **Sound Propagation:** The sensation of sound is due to small variations in air pressure. The 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. We need clever domain decomposition techniques for low- and medium-frequency sources and novel geometric propagation algorithms for high-frequency sources. These algorithms can be further accelerated by using the capabilities of multi-core and many-core commodity processors for interactive computations.

In the rest of this paper, we will describe some of our recent work on interactive sound synthesis and sound propagation.

2. Real-time Sound Synthesis

Sound is produced by surface vibrations of an object under external forces. These vibrations travel through the surrounding medium. The pressure waves within the human audible range are sensed by the ear and render the perception of sound. One may consider the sound synthesis pipeline consists of two parts: the interaction model and the resonators. The forces from object interactions drive the resonators.

The pipeline of our sound synthesis system is shown in Figure 1. The green boxes indicate the components that allow user input. The blue boxes refer to the pre-processing parts. The orange boxes show the processes that are computed at run-time. The interaction handling module detects different types of interactions among the objects and converts them into excitation forces for the sound synthesis module, which is composed of modal synthesis, vibration calculation, and the real-time audio engine. The sound synthesis module approximates the oscillating responses of surfaces when external forces are applied and generates sound from the approximate vibration.

![Figure 1. System overview. The green boxes indicate the parts users can interact with. The blue boxes indicate the pre-processing parts. The orange boxes are the runtime components.](image)

2.1. Interactive Synthesis for Complex Scenes

Typically, the number of modes of an object with a few thousand vertices is in the range of a few thousands and the basic synthesis procedure runs in real-time. But as the number of objects increases beyond two or three, the performance can degrade. In this case, our goal is to decrease the number of modes mixed and utilize the listener’s perception from noticing the difference. Here we briefly discuss a few techniques to improve the performance and ensure that the method works well for interactive applications. For more details, we refer to [40].

2.1.1. Mode Truncation

The sound of a typical object on being struck consists of a transient response composed of a blend of high frequencies, followed by a set of lower frequencies with low amplitude. The transient attack is essential to the quality of sound as it is perceived as the characteristic “timbre” of the object. The idea behind mode truncation is to decrease the number of modes mixed and utilize the initial transient response with a set of lower frequencies. Since mode truncation preserves the initial transient response of the object when \( \tau \) is suitably set, the resulting degradation in quality is minimal.

2.1.2. Mode Compression

A perceptual study described in [45] showed that humans have a limited capacity to discriminate between frequencies which are close to each other. That is, if two “close enough” frequencies are played in succession, the average human listener is unable to tell whether they were two different frequencies or the same frequency played out twice. [40] lists the frequency discrimination at different frequencies. For example, at 2 KHz the frequency discrimination is more than 1 Hz. That is, a human subject cannot tell apart 1999 Hz from 2000 Hz. Note that the frequency discrimination deteriorates...
of our goals is to generalize and expand the physics-based sound synthesis techniques to all media, including liquids, and all type of interaction, including fluid-solid interaction, and thereby make the paradigm applicable to a much broader class of scenarios.

Sound in liquid is primarily due to bubble formation. We refer the reader to Leighton’s excellent text on acoustics due to bubble resonance for more detail [30]. Minnears formula, which derives the resonant frequency of a perfectly spherical bubble in an infinite volume of water from the radius, provides the physical basis for generating sounds in liquid. Recently, Moss et al. [34] have developed the first automatic sound synthesis algorithm. It is based on the fact that the sound generated by a bubble is dominated by the resonant frequency, since all other frequencies will rapidly die out. The resonant frequency is dependent on the restoring force, which is the result of the pressure in the bubble. The resulting approach takes into account both the spherical and non-spherical bubbles.

The resulting simulator is coupled with a fluid simulator that computes liquid dynamics, which subsequently continues to drive the bubble resonance in liquid media. There are many challenging computational issues in this coupling. The first issue is what type of fluid simulation should be used. Three broad categories exist for fluid dynamics computation in visual simulation: grid-based methods, smoothed particle hydrodynamics (SPH), and shallow-water approximations and for different applications, all these three type of simulators are used. In practice, our overall approach can generate plausible liquid sounds for different scenarios, as described in [34].

3. Physics-Based Sound Propagation

The most common approach to acoustic simulation is a two-stage process: the computation of impulse responses (IR) representing an acoustic space, and the convolution of the impulse responses with dry (anechoically recorded or synthetically generated) source signals. The IR computation relies on an accurate calculation for modeling the sound field. The sensation of sound is due to small variations in air pressure.

Physics-based approaches solve the wave equation numerically to obtain the exact behavior of wave propagation in a domain. We assume that the input to our acoustics simulation is a 3D geometric model, along with the boundary conditions and the locations of the sound sources and listener. The propagation of sound in the domain is governed by the Acoustic Wave Equation:

\[
\frac{\partial^2 p}{\partial t^2} - c^2 \nabla^2 p = F(x,t),
\]

This equation captures the complete wave nature of sound, which is treated as a time-varying pressure field \( p(x,t) \) in space. The speed of sound is \( c = 340 \text{m/s} \) and \( F(x,t) \) is the forcing term corresponding to sound sources present in the scene. The operator \( \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \).
\[ \frac{\partial^2}{\partial t^2} + \nabla^2 \] is the Laplacian in 3D.

3.1. Related Work

Numerical methods for solving the wave equation rely on discretizing space and time for solution on a computer. Depending on the underlying discretization method, numerical approaches for acoustics may be roughly classified into: Finite Element Method (FEM), Boundary Element Method (BEM), Digital Waveguide Mesh (DWM), Finite Difference Time Domain (FDTD), and Functional Transform Method (FTM) [36], [38]. The FEM and BEM have traditionally been employed mainly for the steady-state frequency domain response, as opposed to a full time domain solution of the wave equation. FEM is applied mainly to the interior and BEM to exterior scattering problems [26]. DWM approaches [56], on the other hand, are specific to the Wave Equation and use discrete waveguide elements, each of which is assumed to carry waves along its length along a single dimension [24], [43]. However, such approaches suffer from directional dispersion of sound, that is, sound doesn’t travel with the same speed in different directions on the spatial grid. The most commonly used method for time-domain wave simulations is the Finite Difference Time Domain (FDTD) method, owing to its simplicity and efficiency. FDTD has been an active area of research for more than a decade [4], [5] and was first proposed for electromagnetic simulations [46]. FDTD works on a uniform Cartesian grid and solves for the field values at each simulation cell over time. Initial investigations into FDTD were hampered by the lack of computational power and memory, limiting its application to mostly small models in 2D. Over the last decade, the possibility of applying FDTD to medium sized models in 3D has been explored [42]. However, the computational and memory requirements for FDTD are beyond the capability of most desktop systems today [41], requiring days of computation on a small cluster even for medium-sized 3D models for simulating frequencies below 1 kHz. Other methods are based on spectral techniques [6], which are a class of very high order numerical schemes in which the complete field is expressed in terms of global basis functions, which virtually eliminates spatial approximation errors.

3.2. Adaptive Spatial Decomposition

Physics-based approaches for sound propagation attempt to directly solve the Acoustic Wave Equation numerically, which governs all linear sound propagation phenomena, and are thus capable of performing a full transient solution which correctly accounts for all wave phenomena, including diffraction, elegantly in one framework. However, as mentioned above, computational costs for acoustic simulation can be prohibitively high. Recently Raghuvanshi et al. [39] have developed a novel approach based on adaptive rectangular decomposition of the free space in the domain to address this issue. Fig. 2 gives an overview of this approach. Many problems of interest for the purpose of acoustic simulation necessarily have large empty spaces in their interior. Consider a simulation on a large scene like an auditorium with an impulse triggered near the floor. With FDTD, this impulse would travel upwards and would accumulate numerical dispersion error at each spatial cell it crosses. Typically, an impulse would cross thousands of cells just to travel from one end of the small scene to the other, accumulating a lot of error in the process. However, if we fit a rectangle in the scene extending from the bottom to the top, the impulse would have no propagation error, since we can utilize the analytical solution within a rectangle. This observation is the main motivation for computing a rectangular decomposition.

In order to achieve high spectral accuracy, both in time and space, in the interior of the rectangular partitions, Raghuvanshi et al. [39] use sixth-order spatial accuracy on the interface between partitions. Fig. 3 shows the results of our acoustic simulations after 5, 20, 50, 100 msec respectively for sound propagation within a cathedral scene.

From the perspective of parallelizing numerical solvers, Domain Decomposition Methods (DDM) have been widely studied for scientific computation [14], [37], [51]. The rectangular partitioning also provides a natural domain decomposition and therefore, is a good candidate for parallelization on GPU clusters or many-core processors. We plan to further investigate the parallelization of numerical acoustic simulations using commodity hardware.

4. Geometric Sound Propagation

The exact solution of sound propagation reduces to solving the wave equation, as described in the previous discussion. However, numerical methods to solve wave equation are mainly limited to static scenes. Given that the underlying complexity of numerical methods increases as fourth power of sound frequency, they are mostly useful for low or medium frequency
The dimensions of this scene are 35 m × 15 m × 26 m. We are able to perform numerical sound simulation on this complex scene on a desktop computer and pre-compute a 1 second long impulse response in about 29 minutes, taking less than 1 GB of memory. A commonly used approach that we compare against, Finite Difference Time Domain (FDTD), would take 1 week of computation and 25 GB of memory for this scene to achieve competitive accuracy. The auralization, or sound rendering at run-time consists of convolution of the calculated impulse responses with arbitrary source signals, that can be computed efficiently.

The alternate methods for handling high-frequency sound sources are based on sound field decomposition or geometric propagation [19], [47], [48]. These algorithms model the propagation of sound based on rectilinear propagation of waves and can accurately model the early reflections (up to 4−6). These equivalent sources conceptually radiate in free space and are, in principle, supposed to fulfill the boundary condition of the acoustic space. The various equivalent sources are used to compute a list of elementary waves that arrive at the receiver. This includes the source data, propagation data that consists of frequency dependent attenuation due to distance spreading, accumulated absorption loss at wall reflections, accumulated scattering loss at wall reflections, and air absorption.

4.1. Prior Work

Some of the earliest methods for geometric sound propagation were based on tracing sampled-rays [27] or sound-particles (phonons) [23], [2], [15] from a source to the listener. However, these methods are susceptible to aliasing problems. As a result, accurate methods for geometric sound propagation are either based on image source methods or volumetric propagation. Image source methods are the easiest and most popular for computing specular reflections [1], [13]. However, they can only handle simple static scenes or very low order of reflections at interactive rates [28]. Many hybrid combinations [12] of ray-based and image source methods have been proposed and used in commercial room acoustics prediction software (e.g. ODEON). But they are limited to rather simple static scenes and are not fast enough for interactive applications. The volumetric geometric methods trace pyramidal or volumetric beams to compute an accurate geometric solution. These include beam tracing that has been used for specular reflection and edge diffraction for interactive sound propagation [20], [28], [52]. The results of beam tracing can be used to guide sampling for path tracing [18]. However, they are only limited to very simple CAD models and it is rather difficult to make them work on complex CAD models due to robustness problems.

4.2. Ray-Frustum Tracing

Recently, we have introduced a new volumetric approach, called “ray-frustum tracing” [29], [10]. This formulation uses a volumetric frustum tracing and utilizes the recent developments in interactive ray tracing to perform fast volumetric tracing using commodity processors. The overall approach is built on fast ray-tracing algorithms and represent each frustum using four corner rays. Unlike, prior beam tracing algorithm performs discrete clipping at the intersections, as shown in Fig. 4. The resulting approach can easily exploit the current hardware features in terms of data-parallelism and thread-level parallelism and thereby makes it possible to perform specular reflections and direct contributions at interactive rates. Moreover, it uses bounding volume hierarchies to accelerate the intersection tests and can also handle dynamic scenes. Recently, Chandak et al. [9] have presented a practical and conservative
visibility culling algorithm and used that for exact sound propagation. It is almost 10-20 times faster than prior beam-tracing algorithms and can accurately compute the propagation paths from the source to the listener.

The initial frustum-tracing algorithm been primarily used for handling specular reflections in complex, dynamic scenes. Recently, we combined the frustum-tracing formulation with the UTD formulation to perform edge-diffraction in complex virtual environments [50]. This approach is primarily designed for scenes with very large (or infinite-sized) edges, e.g., outdoor scenes with buildings. In terms of future work, it may be useful to combine this approach with the Biot-Tolstoy-Medwin (BTM) method [7], [25]. The BTM method is more accurate than UTD and can be formulated for use with finite edge [49]. However, it involves solving the BTM line integral over the edge length exposed to the sound field. As a result, BTM-based methods have a higher computational cost and recent research has focused on non-interactive acceleration techniques [32] used in offline simulations [33], [44] and edge subdivision techniques for interactive simulation of relative simple scenes [49], [8], [7].

4.3. Acoustic Levels-of-detail

The 3D models used in most desktop and CAD applications are highly detailed. Such detail is often needed for visually realistic rendering of the scene. Acoustic simulation, on the contrary, requires much simpler 3D models with lower level of detail. A detailed 3D model may increase the computation cost of the simulation without any perceptual differences in sound rendering [53]. In order to accelerate the computations, a separate acoustic geometric representation may be needed and used for interactive frustum tracing.

4.4. Audio Rendering and 3D Audio

Audio rendering is the process of producing the final output audio signal and is an important building block of interactive sound rendering. In order to render audio from geometric sound propagation we first construct an impulse response (IR) for each source–listener pair from the sound paths that reach the listener from the source. A source–listener pair’s IR is then convolved with the input audio signal of the source to produce the final output audio signal. In terms of extending this pipeline to complex, dynamic scenes, there are two main challenges; firstly, the IRs are constantly changing due to the dynamic nature of the application. Secondly, the incoming path to the listener has a direction relative to the listener and a path length associated with it. Therefore, we treat it as a 3D sound source for realistic audio rendering. The computation of 3D sound involves convolution of incoming sound with the head related impulse response (HRIR) for every path. This can become computationally expensive and can be challenging when sound propagation is performed for a large number of sound sources. Some exciting work has recently been done to perform 3D audio rendering of large number of sound sources using sampling-based techniques [57].

5. Conclusions

In this paper, we gave a brief overview of our recent work on sound synthesis and propagation. This includes fast algorithms to generate sound from rigid bodies and liquid sounds. Moreover, we also gave an overview of new algorithms for numeric and geometric propagation, which are considerably faster than prior approaches. We also highlighted a few areas for future research.

Acknowledgements: The work described in this paper represents joint work of the authors with Anish Chandak, Christian Lauterbach, William Moss, Nikunj Raghuvanshi, Zhimin Ren, Micah Taylor and Hengchin Yeh at UNC Chapel Hill. This research was supported in part by ARO Contract W911NF-04-1-0088, NSF awards 0429583 and 0636208, DARPA/RDECOM Contracts N61340-015-C-0043 and WR91CRB-08-C-0137, Intel, and Microsoft.

References


Figure 5. The left image shows the frusta generated by our adaptive volume tracing algorithm inside the 2.5 million triangle model of Soda Hall. The simulation is run with up to four specular reflections, edge diffraction and transmission. The right hand images show the new frusta generated on the fly, as we vary the dimensions of the room. The AD-FRUSTA algorithm can compute the propagation paths at five frames per second in this dynamic scene by using seven cores on a high PC.


