# WAVE: Interactive Wave-based Sound Propagation for Virtual Environments

Ravish Mehra, Atul Rungta, Abhinav Golas, Ming Lin, and Dinesh Manocha

**Abstract**—We present an interactive wave-based sound propagation system that generates accurate, realistic sound in virtual environments for dynamic (moving) sources and listeners. We propose a novel algorithm to accurately solve the wave equation for dynamic sources and listeners using a combination of precomputation techniques and GPU-based runtime evaluation. Our system can handle large environments typically used in VR applications, compute spatial sound corresponding to listener's motion (including head tracking) and handle both omnidirectional and directional sources, all at interactive rates. As compared to prior wave-based techniques applied to large scenes with moving sources, we observe significant improvement in runtime memory. The overall sound-propagation and rendering system has been integrated with the Half-Life 2 game engine, Oculus-Rift head-mounted display, and the Xbox game controller to enable users to experience high-quality acoustic effects (e.g., amplification, diffraction low-passing, high-order scattering) and spatial audio, based on their interactions in the VR application. We provide the results of preliminary user evaluations, conducted to study the impact of wave-based acoustic effects and spatial audio on users' navigation performance in virtual environments.

**Index Terms**—Sound propagation, dynamic sources, directivity, spatial sound, Helmholtz equation

## 1 Introduction

Realistic sound propagation is extremely important in VR applications for improving the sense of presence and immersion of the user in the virtual environment and augmenting the visual sense of the user resulting in increased situational awareness [3, 14, 2]. In combat training simulations [25], sound propagation cues can provide additional information about the environment (small vs. big, inside vs. outside) and about events happening outside the field-of-view (such as enemy sneaking from behind); they can also help localize the direction of gunfire. Since these cues would be available to soldiers in a real-life scenario, it is important for soldiers to train with these cues within the VR simulation as well [12]. In VR exposure therapy, patients are asked to experience the trauma-related VR environment, where they are exposed to 'trigger' stimuli (visual, auditory, olfactory, and tactile) in a controlled manner for therapeutic gain. Sound propagation is necessary to generate accurate auditory cues to improve the patient's sense of immersion in the virtual environment.

Most current games or VR systems use simple algorithms based on reverberation filters or ray tracing techniques to generate plausible sound. While these techniques may work for game-like scenarios, they may not provide accurate solutions for many VR applications. In this paper, we mainly focus on sound propagation techniques that generate realistic and accurate sound effects for virtual environments.

Sound propagation techniques can be classified into geometric and wave-based techniques. Geometric techniques, which assume that sound waves travel as rays (a valid assumption at higher frequencies), can handle large environments and both directional and dynamic sources and listeners [9]. However, modeling wave-effects such as diffraction and interference, which are prominent at low frequencies, remains a significant challenge with geometric techniques. Wave-based techniques, on the other hand, can accurately perform sound propagation at all frequencies, and can model all acoustic effects, including the wave effects [38, 29]. Prior wave-based techniques are primarily designed for scenes with static sources (although they can handle moving listeners). In order to handle moving sources at runtime, these algorithms precompute the pressure field at a large number of sampled source positions, considerably increasing the precomputation cost and storage overhead [29, 18].

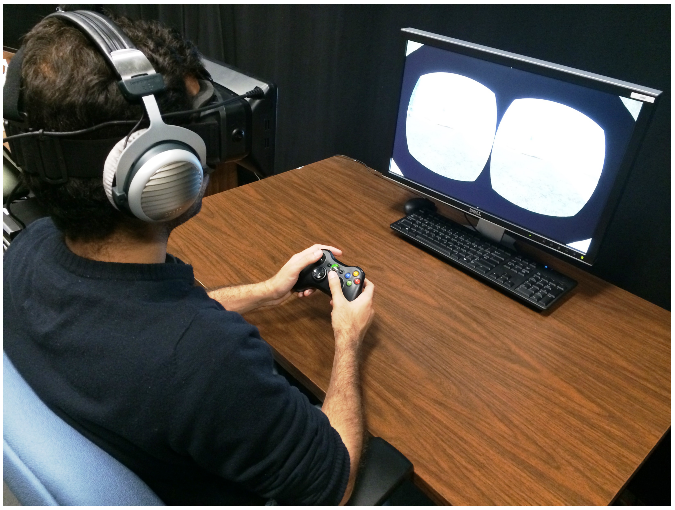

Fig. 1. User experiencing a VR environment with our WAVE sound propagation system. Our system tracks player's interactions using the Oculus Rift head-mounted display (HMD) and the Xbox 360 controller and delivers high-fidelity acoustics effects and spatial sound using headphones.

We present WAVE (Wave-based Acoustics for Virtual Environments), an interactive wave-based sound propagation system for efficiently generating accurate and realistic sound for VR applications. Our approach is based on a recent numerical solver for the acoustic wave-equation called the "Equivalent Source Method" (ESM) [19], which can handle large environments. We present a novel wave-based algorithm that can directly compute an accurate solution for dynamic sources and listeners. This results in significant improvement in runtime memory overhead over prior techniques. WAVE can handle both omnidirectional and directional sources and can interactively compute spatial audio that corresponds to the listener's motion and head rota-

- *Ravish Mehra is with UNC Chapel Hill. E-mail: ravishm@cs.unc.edu.*
- *Atul Rungta is with UNC Chapel Hill. E-mail: rungta@cs.unc.edu.*
- *Abhinav Golas is with UNC Chapel Hill and Samsung Research. E-mail: golas@cs.unc.edu.*
- *Ming Lin is with UNC Chapel Hill. E-mail: lin@cs.unc.edu*
- *Dinesh Manocha is with UNC Chapel Hill. E-mail: dm@cs.unc.edu*

*Manuscript received 18 Sept. 2014; accepted 10 Jan. 2015. Date of Publication 20 Jan. 2015; date of current version 23 Mar. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.*

tion. The key contributions of our work include:

1. Sampling-free sound propagation for dynamic sources and listeners. This was accomplished by formulating of the ESM propagation equation into matrix-vector product using the dynamic transfer operator.

2. Rank-based compression to reduce runtime memory via the singular value decomposition of the dynamic transfer operator. We use a load-balanced, hybrid CPU-GPU runtime algorithm to obtain interactive performance.

3. User evaluation to study the effect of wave-based acoustic effects and spatial audio on users' navigation performance in virtual environments.

Our system has been integrated with the Half-Life 2 game engine; users can experience wave-based acoustic effects and spatial audio based on their interactions in the virtual environments. The user's position and interactions are transferred to the system using the Oculus-Rift head mounted display (for user's head orientation) and the Xbox game controller (for user's position). We have performed preliminary user evaluation to study the effect of our system on user's navigation performance in virtual environments. The results of the experiments suggest that our approach allows users to perform source localization and navigation tasks more effectively compared to prior geometric approaches. The task performance is further improved when spatial audio rendering is used.

## 2 Motivation

A VR system trying to recreate a real-world environment needs to recreate the acoustics of the real world as well. This requires generating high-fidelity acoustic effects that accurately model real-world experiences. In this section, we explore the motivating scenarios and required features of sound propagation systems built for VR applications.

### 2.1 Dynamic sources and listeners

A typical VR scenario consists of multiple sources and listeners moving dynamically in a 3D environment. In the case of a first-person shooter VR game, the sources can be enemy gun ships, helicopters, or soldiers; the listener would typically be the player's character. For VR flight simulations, sources are other aircraft flying around the user-controlled aircraft. In most of these scenarios, both sound sources and listeners can move anywhere in the 3D space. Therefore, one of the most important features for a VR-centric sound-propagation system is support for fully dynamic sources and listeners in three dimensions.

### 2.2 Source directivity

Most sound sources that we come across in the real world are directional sources; typical examples include the human voice, speakers, and musical instruments. This directivity is frequency-dependent and, in many cases, time-varying, e.g., a rotating siren or a person covering his/her mouth. Source directivity can have a significant effect on the acoustics of the environment, such as a talking person walking towards/away from the listener, the positioning of music instruments in an orchestra, etc. Therefore, a sound propagation system for VR applications needs to support both omnidirectional and directional sources.

### 2.3 Spatial audio

The human auditory system obtains significant directional cues from the subtle differences in the sound received by the left and the right ear. These differences are caused by the scattering of sound around the human head, and they change with the direction of incoming sound. Spatial audio entails generating the appropriate sound signals at the listener's ears, using speakers or headphones, giving the listener the impression that the sound is coming from the intended direction. Spatial audio can be used to enhance user's immersion in a VR application by providing additional cues about actions happening outside user's field-of-view, such as an enemy sneaking from behind in a combat-training simulations. It can also help the player locate objects and sound sources in the virtual world. Therefore, spatial audio is another important feature that needs to be supported by a sound propagation system designed for VR applications.

## 3 Related Work

In this section, we discuss the prior work in the field of sound propagation.

### 3.1 Sound Propagation techniques

Most current sound propagation systems used for interactive applications are based on geometric techniques. The most commonly used geometric technique for modeling specular reflections is the image source method which can be accelerated using beam tracing [8]. Other techniques based on ray tracing and radiosity have been developed for modeling specular and diffuse reflections, respectively [9]. Techniques based on acoustic radiance transfer [31] can model arbitrary surface interactions for wide-band signals. The two main approaches for modeling diffraction in a geometric acoustics framework are based on the uniform theory of diffraction (UTD) [39] and the Biot-Tolstoy-Medwin (BTM) formulation [34]. UTD is an approximate formulation, while the BTM is an offline technique that yields accurate results, but with significant performance cost. Recent work has incorporated UTD-based diffraction in interactive geometric techniques [36, 30].

Wave solvers can be classified into *frequency-domain* and *time-domain* approaches. The most common among frequency-domain techniques include the finite element method (FEM) [37] and the boundary element method (BEM) [6]. Of the time-domain approaches, the most popular is the finite difference time domain (FDTD) method [35]. The compute and memory usages of these techniques scale linearly with the volume or surface area of the scene.

### 3.2 Dynamic sources

Most interactive wave-based techniques can handle either dynamic sources or dynamic listeners but not both simultaneously [19, 42]. Some wave-based techniques that can handle both dynamic sources and listeners precompute simulation results at sampled positions on a uniform grid, and interpolate these samples at runtime [29, 18]. However, in order to reduce the runtime memory overhead, these techniques restrict either sources or listeners to move on a 2D grid and use a coarse-sampling scheme. When both sources and listeners are allowed to move in 3D, precomputation time and runtime memory requirements increase significantly, making these approaches infeasible. The coarse-sampling scheme can create interpolation artifacts, especially in the shadow regions close to the objects where the sound might appear to come from the wrong position.

### 3.3 Directional sources

Real-world sound sources have characteristic directivities that vary with frequency [20], and these source directivities have a significant effect on the propagation of sound in an environment [41]. Interactive geometric techniques can incorporate high-frequency source directivities at runtime [9], while interactive wave-based sound propagation techniques can handle low-frequency directional sources. Recent wave-based techniques [29, 19] can handle static, elementary source directivities such as monopoles, dipoles, and their linear combinations. Other approaches have been proposed to incorporate static, measured directivities in wave-based techniques [10]. Recently, Mehra et al.[18] proposed an interactive wave-based technique to handle time-varying source directivity at runtime.

### 3.4 Spatial Sound

Scattering of sound around the human head produces significant directional cues for the human hearing system [3]. These scattering effects are represented using the *head-related transfer function* (HRTF). Measurements to compute HRTF are performed in controlled environments and the recorded data is available online [1]. Integrating HRTFs into wave-based techniques for generating spatial audio requires computing the direction of sound field propagation using *plane wave decomposition*. Prior plane-wave decomposition techniques either use spherical convolution [27] or solve a linear system [43], and are computationally expensive. To avoid the expensive calculations required to integrate HRTF, some interactive wave-based techniques resort to simpler listener directivity models based on a spherical head and a cardioid function [29]. However, these simplified models are not accurate for sound localization and externalization, both of which are necessary for immersion in virtual environments [3]. Recently, an interactive plane-wave decomposition technique was proposed that computes the decomposition efficiently using derivatives of the pressure field at the listener position [18]. The most common method for delivering spatial audio is by means of headphones. However, there has also been work on delivering spatial audio using speaker arrays. These *wave-field synthesis* approaches enable a listener-position independent spatial audio reconstruction for a multi-user audio-visual environment [7, 33].

### 3.5 Sound in VR

Sound research in VR has focused mainly on how sound quality, spatialization, and binaural rendering impact the users sense of presence. Hendrix and Barfield [11] showed that incorporating sound in virtual environment (VE) increased the overall sense of presence for users. In a broader study, Ozawa et al [24] evaluated the effect of sound quality, sound information, and sound localization on the user and observed that the latter two had a strong correlation with presence sensed by users. Similarly, use of spatial sound resulted in significantly higher presence ratings than non-spatial sound in virtual environments [11, 21]. For spatial sound, Väljamäe et al. [40] found that individualized HRTF had a more positive influence on presence than generic HRTFs. Larsson et al. [15] compared a virtual environment containing anaechoic audio to a virtual environment with room acoustic cues and showed that room acoustic cues resulted in a significant increase in presence. There has also been work that examines how source signal type and spatial audio techniques impact navigation performance in virtual environments [17, 16]. However, no one has yet studied the effect on navigation performance of different sound propagation techniques (geometric vs. wave-based) in virtual environments.

## 4 WAVE System

We start by providing a brief background of the equivalent source method, which forms the basis of our sound simulation system.

### 4.1 Background: Equivalent Source Method (ESM)

The equivalent source method in Mehra et al. [19] is a frequency-domain approach for modeling wave-based sound propagation in large, open environments. It is based on the equivalent source formulation proposed for acoustic radiation and scattering problems [22]. It works on the principle of decomposing the global sound field into per-object and inter-object interactions. Per-object interactions encapsulate the manner in which the object modifies (reflects, scatters, diffracts) the sound field. These interactions are captured by the *per-object* transfer function that maps an arbitrary incoming field impinging the object to the corresponding outgoing scattered field. Inter-object interactions encapsulate the effect of the outgoing sound field of one object on another object. These interactions are modeled by the *inter-object* transfer function that maps the outgoing scattered field of an object to the incoming field of another object. In the equivalent sound formulation, both these transfer functions are stored compactly using efficient basis functions called the equivalent sources. The per-object transfer function is represented by the scattering matrix $\mathbf{T}$ and the inter-object transfer function is represented by the interaction matrix $\mathbf{G}$. The size of the matrices $\mathbf{T}$ and $\mathbf{G}$ depends on the number of objects in the scene and the simulation frequency. The sound field emitted by a source is represented by vector $\mathbf{S}$. The global sound field for the entire scene is computed by solving the linear system

$$(\mathbf{I} - \mathbf{TG})\mathbf{C} = \mathbf{TS}, \tag{1}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{C}$ is strength vector of the propagated sound field for the entire scene. This technique is an object-centric approach for wave-based sound propagation as compared to previous volumetric or surface area-based approaches [35, 6, 37].

The computation of per-object and inter-object transfer functions and the solution of linear system is performed at the pre-processing stage for a *fixed source position with static directivity*. The strength vector $\mathbf{C}$, which is stored for runtime use, captures the propagated sound field corresponding to the incoming sound field $\mathbf{S}$ produced by the source. At runtime, the strength vector $\mathbf{C}$ is used to generate the sound field for that fixed source with static directivity at a moving listener position. Note that for each different source position and each different source directivity, this linear system must be solved again, and the resulting strength vectors must be stored for runtime use. The principle of acoustic reciprocity which states that one can reverse the sense of source and listener without changing the acoustic response [26, p. 195-199], can be used to support dynamic sources and a static listener. However, this technique, in its current form, cannot handle both dynamic sources and dynamic listeners in three dimensions at runtime.

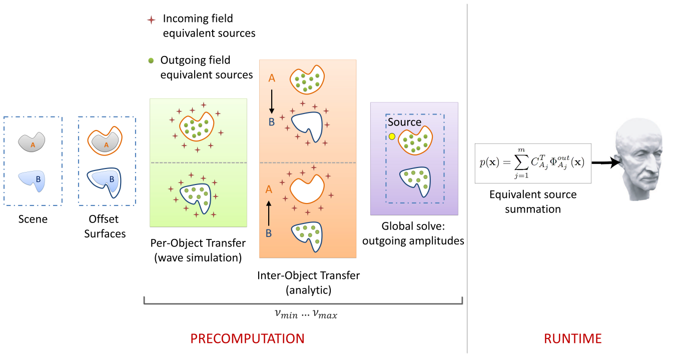
Incoming field equivalent sources
Outgoing field equivalent sources
A
B
Scene
A
B
Offset Surfaces
Per-Object Transfer (wave simulation)
A
B
A
B
Inter-Object Transfer (analytic)
Source
Global solve: outgoing amplitudes
$p(\mathbf{x}) = \sum_{j=1}^{m} C_{A_j}^T \Phi_{A_j}^{out}(\mathbf{x})$
Equivalent source summation
$\nu_{min} \cdots \nu_{max}$
PRECOMPUTATION
RUNTIME

Fig. 2. **Stages of the equivalent source method in Mehra et al.[19].** Given a scene with multiple objects, the per-object and inter-object transfer functions are precomputed for each object and each object pair in the scene, respectively. Based on the sound source position, the global sound field is solved for the entire scene and stored as strengths of the equivalent sources. At runtime, the global propagated sound field is evaluated at the listener position by fast summation of sound fields due to all the equivalent sources weighted by their precomputed strengths.

### 4.2 WAVE: Overview

Figure 3 gives an overview of our approach. Our sound simulation system consists of two stages: *pre-processing* and *runtime*. In the pre-processing stage, we start with the 3D model of the scene along with its material absorption coefficients as input. Next, we compute the per-object transfer functions for all the objects in the scene and compute inter-object transfer functions for all the object pairs in the scene. Using these transfer functions, we compute the *dynamic transfer operator* for the scene as a whole. This operator encapsulates the sound propagation behavior of the entire scene for any incoming sound field produced by a sound source (moving or with varying directivity). The dynamic transfer operator is compressed and stored for runtime use.

At runtime, the VR application provides the instantaneous position and directivity of the sound source; this is used as input to the "dynamic source" and "source directivity" modules of our sound simulation system, respectively. Next, the game controller and head-mounted display provide the instantaneous position and orientation of the listener, which are used as inputs to the "spatial sound" module. The dynamic transfer operator is applied to the incoming sound field of the moving, directional source to generate the strength vector $\mathbf{C}$ of the propagated sound field. Finally, this strength vector is used to compute the binaural responses at the moving listener, which are then rendered to the user over headphones.

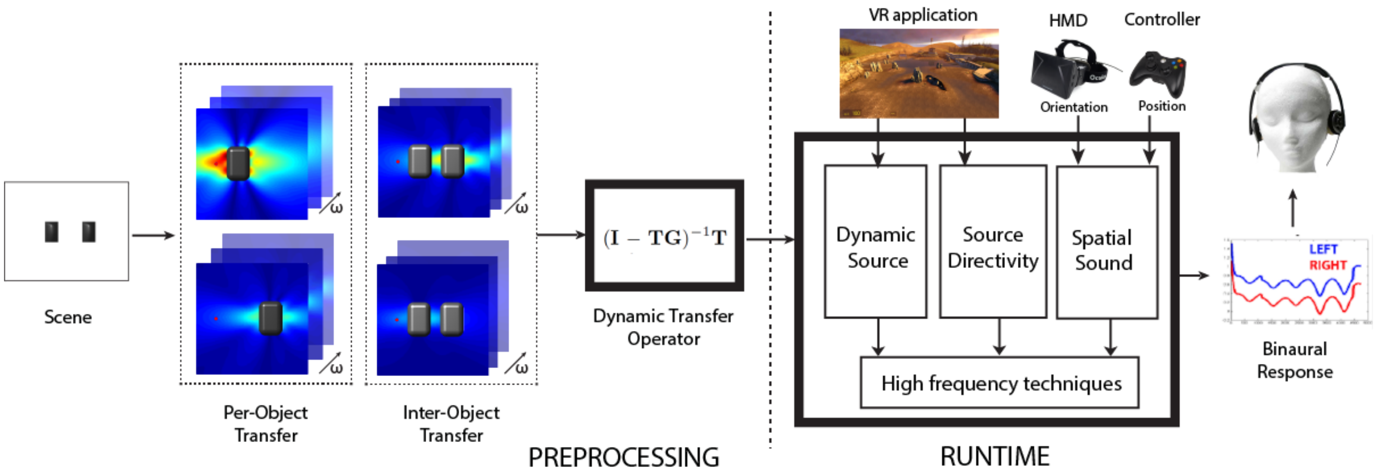

Fig. 3. **Overview of our WAVE system.** It consists of two stages: preprocessing and runtime. During the preprocessing stage, we compute the dynamic transfer operator of the scene using the per-object and inter-object transfer functions. At runtime, the VR application, the controller, and the head-mounted display provide instantaneous source position and directivity, listener's position and orientation as input to our sound simulation system, respectively. Our system then computes high-fidelity acoustics effects and spatial audio, which is rendered to the player over headphones.

## 4.3 Dynamic Transfer Operator

The main intuition behind our approach for handling dynamic (moving) sources and listeners is the following: the scattering behavior of an object can be precomputed and stored as per-object transfer function; as long as the object remains rigid, and its material properties do not change, the sound propagation due to an object remains the same. Similarly, the acoustic interactions between the objects can also be precomputed and stored using the inter-object transfer function; if the objects do not move or rotate, the acoustic interactions between objects remain unchanged. Therefore, even in cases of dynamic sources or time-varying directivity, the only changes are those of the incoming sound field generated by the source. This change requires recomputing the solution of the linear system (Equation (1)). Note that in equation (1)), only the right-hand side vector of the linear system $\mathbf{S}$ changes; the matrices on the left-hand side remain the same. Therefore, the above linear system can be transformed into a matrix vector product as

$$\mathbf{C} = (\mathbf{I}-\mathbf{TG})^{-1}\mathbf{TS} = \mathbf{DS}, \tag{2}$$

where matrix $\mathbf{D} = (\mathbf{I}-\mathbf{TG})^{-1}\mathbf{T}$ is called the *dynamic transfer operator*. This operator is precomputed for the scene and stored for runtime use.

The key difference between the previous ESM approach [19] and our current approach is the stage at which the propagation equation is solved. In previous ESM technique [19], the propagation equation is solved during the preprocessing stage; only the strength vectors $\mathbf{C}$ are stored for runtime. While this has a smaller memory footprint, it supports only fixed source position and directivity. In the current approach, the propagation equation is solved at runtime using the dynamic transfer operator which must be precomputed and stored during the preprocessing step to be used at runtime. Though it is more memory-intensive, our formulation supports dynamic sources and time-varying directivity at runtime, as described below.

### 4.3.1 Handling dynamic sources and listeners

As the sound source moves, the incoming sound field changes and we need to recompute the solution of the linear system (Equation (1)). As shown above, the linear system computation can be transformed as a matrix-vector multiplication. With our approach, as the source moves, we update the incoming sound field vector $\mathbf{S}$ and perform a matrix vector product with the dynamic transfer operator $\mathbf{D}$ at runtime. This gives us the strength vectors $\mathbf{C}$, which can be used to compute the sound field at a moving listener in a similar manner as Mehra et al. [19]. Note that the dynamic transfer operator formulation solves the wave equation for the exact source and listener positions, and, unlike previous sampling-based approaches [29, 18], does not use any interpolation.

### 4.3.2 Dynamic source directivity

In case of dynamic (time-varying) directivity, the incoming sound field vector $\mathbf{S}$ changes with changing directivity. The dynamic transfer operator formulation is agnostic of the manner in which $\mathbf{S}$ changes. Therefore, the same dynamic transfer operator used to compute sound propagation for dynamic sources can be used to handle time-varying source directivity, computing it directly at runtime. Note that our approach computes the propagated sound field due to the time-varying source directivity directly at runtime.

### 4.3.3 Spatial audio

We use the plane-wave decomposition approach proposed by Mehra et al. [18] to generate spatial audio for the listener. This approach computes the spherical harmonic coefficients of the plane-wave decomposition using derivatives of the sound pressure field at the listener position. The spatial sound is then computed as a dot product of the spherical harmonic coefficients of the plane-wave decomposition and the head-related transfer function (HRTF). This approach supports listener motion and head rotation at interactive rates. The key change that we made to the Mehra et al. [18] approach is in the computation of the strength vector $\mathbf{C}$ used to evaluate the derivatives of sound field: in our method, $\mathbf{C}$ is now computed at runtime using the dynamic transfer operator. This allows us to compute spatial audio corresponding to fully dynamic and directional sources and listeners free to move anywhere in three dimensions.

## 4.4 Memory compression

To reduce the runtime memory requirement of our algorithm, we use an accurate and compact approximation of the dynamic transfer operator $\mathbf{D}$. Specifically, we propose the use of the $\epsilon$-approximate operator $\mathbf{D}_\epsilon$ where $\|\mathbf{D}-\mathbf{D}_\epsilon\| < \epsilon$. As is widely established by literature, the best way to compute $\mathbf{D}_\epsilon$ is to use the Singular Value Decomposition (SVD). By choosing the first $k$ singular values of the SVD of the dynamic transfer operator $\mathbf{D} = \mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}\mathbf{V}^t_{n\times n}$, we can express $\mathbf{D}_\epsilon$ as

$$\mathbf{D}_\epsilon = \mathbf{U}_{m\times k}\mathbf{\Sigma}_{k\times k}\mathbf{V}^t_{k\times n} \quad , \tag{3}$$

where $k$ is chosen to satisfy the $\epsilon$ error bound, and subscripts denote matrix dimensions, $m < n$. The error bound $\epsilon$ is typically chosen to obtain higher accuracy. We use the notion of error threshold as a means of controlling quality. The value of error threshold was chosen based on existing literature [13, 29]. This decomposition can be stored as two matrices $W = U\Sigma$, and $V$. The resulting compression factor is $\frac{mn}{(m+n)k}$. The matrix- vector multiplication $\mathbf{DS}$ becomes an $O((m+n)\,k)$ operation, as opposed to $O(mn)$ for the exact representation.

### 4.5 Parallelization on CPUs and GPUs

In the context of matrix-vector multiplication, GPUs, with their higher memory bandwidth, are ideally suited for large linear systems. CPU architectures, on the other hand, have deep caches, and are therefore better suited for small linear systems. Using this insight, we compute solutions for larger matrices on GPUs and smaller ones on CPUs. To minimize memory transfers across devices and the associated lag, matrices $W$ and $V$ are kept on the same device (CPU or GPU). For the sake of clarity, from this point forth, we consider the two matrix-vector products as one operation. In addition, the distribution of work is performed statically, and not changed at runtime. Since rendering and auralization pipelines pose temporally-consistent demands on resources, we find this assumption to be sufficient for interactive performance. Each processor (CPU or GPU) is thus responsible for a range of matrices in the list, which is sorted based on matrix size. At runtime, the incoming sound field vectors $\mathbf{S}$ for each frequency are partitioned and sent to the appropriate processor. Then, matrix-vector multiplications are performed and results are copied back to assemble the solution vector $\mathbf{C}$.

**Work Distribution** In order to obtain the best performance, work should be distributed such that all computations are completed as soon as possible. In the simplified case of only CPU and GPU resources, this distribution can be modeled as a partition of the list of matrices – sorted in descending size – under memory and computational constraints. Thus, GPUs are used to compute the results for the first $p$ matrices $[0,\ p]$ and CPUs are responsible for matrices numbered $[p+1,\ T]$, where $T$ denotes the total number of matrices (equal to number of frequencies). The computation of the optimal value of $p$ can then be viewed as a search in a constrained 3-dimensional space, with the number of CPU and GPU threads being the other two dimensions. Available memory and computation on each device constrains the search space, since these are also required for other operations. The optimal partition $p$ must be computed only once for any given scene on each unique hardware configuration, and can be pre-computed. Owing to the speed of testing a particular configuration, a naïve brute force method is sufficiently fast for this purpose.

### 4.6 High frequencies

Wave-based techniques are computationally practical for low frequencies of sound ($<$ few kHz). However, this is the range of frequencies where wave-effects such as diffraction and interference are most dominant. In order to handle high frequencies using wave-based techniques, typically two techniques are employed: *spectral extrapolation* and *hybrid propagation*. In spectral extrapolation, propagation results at low frequencies are used to generate plausible results for higher frequencies [29, 19]. These spectral extrapolation techniques are computationally efficient but generate approximate solutions which converge to exact solutions for single-edge diffraction configurations. In other scenarios, the extrapolation techniques generate plausible results for high frequencies. A general spectral extrapolation technique for arbitrary scenes with guarantees on extrapolation error is an important area of future research. Hybrid propagation techniques combine the results of wave-based techniques at low frequencies and geometric techniques at high frequencies to generate the full frequency results [32, 42]. A typical value of the crossover frequency is $1-2$ kHz and low-pass-high-pass filter combination is usually used to combine the results. These techniques generate accurate results over the complete frequency range but are computationally more expensive at runtime. In our current implementation, we use a spectral extrapolation technique similar to that used in Mehra et al. [19] to generate the responses for high frequencies.

## 5 Implementation

Next, we discuss the implementation details of our WAVE system.

### 5.1 Hardware

The WAVE system is implemented on a desktop machine with Intel Xeon E5-2687W CPU (3.1 GHz), 64 GB RAM, and NVIDIA Geforce GTX Titan GPU. We use the Oculus Rift Development Kit 1 Head-Mounted Display (HMD) with a resolution 640x800 per eye and 110 degrees diagonal field of view as the display device. It contains a combination of 3-axis gyros, accelerometers, and magnetometers to compute head orientation without drift. Audio was rendered using the Beyerdynamic DT 990 headphones for each user. The head orientation given by the HMD is used for both visual rendering and spatial audio auralization. The movement of the player in the VR application is controlled using an Xbox 360 controller.

### 5.2 Software

To obtain interactive performance using our algorithm, efficient implementations of the matrix-vector multiplication routines are essential. To this end, we use optimized BLAS libraries for both CPUs and GPUs – the Intel ® Math Kernel Library (MKL), and NVIDIA ® CUBLAS. At runtime, all required memory allocations are made initially to avoid expensive reallocations. For further efficiency on CPUs, this memory is aligned to the 64 byte boundary. Parallelism is controlled via the use of streams on GPUs and threads on CPUs. For GPUs, all operations, including memory transfers and computations, are performed asynchronously, with events used to enforce dependencies between operations. Matrices are statically assigned to a stream/thread for computation. This parallel implementation on CPU and GPU is essential for interactive performance. Our system uses 2 CPU threads, and 16 GPU streams on our hardware setup. We have integrated our sound simulation system with the Valve's Source$^{TM}$ game engine framework. We used standard KEMAR dataset for the HRTF computation.

## 6 User Evaluation

We conducted a user study to evaluate the effect of the WAVE system on users' navigation task performance in the virtual environment.

### 6.1 Study Design

We present two experiments to study the impact of our WAVE system on how well users navigate in virtual environments. In the first experiment, we compare our wave-based approach to an interactive geometric acoustics approach. Both the approaches had monaural audio (spatial audio disabled). The second experiment compares our wave-based approach with and without spatial audio. In the first experiment, our main goal was to quantify the improvement in user's task performance with respect to acoustic effects generated by our wave-based system vs. a pure geometric system. To avoid the additional effect of spatial audio, we used monaural audio for both the systems in this experiment. In the second experiment, we compared the performance of our wave-based approach with and without spatial audio rendering; this was done to study what additional effect wave-based spatial audio has on source localization and navigation performance.

We conducted between-group experiments. Experiments were conducted with three groups (geometric, wave, wave+spatial) and two conditions (geometric vs. wave, wave vs. wave+spatial). Subjects were randomly and independently sorted into the three groups. Each subject in the group was asked to perform the same navigation task: attempt to locate and navigate to the sound source in a VR environment. To avoid any learning effects, each subject performed only one condition and one iteration of the task. The starting positions of source and listener were kept the same for all the subjects. The source was kept stationary and visualized as a radio set. The audio used in the experiment was a music clip. The task-completion time was measured as the elapsed time from the moment the subject starts walking to the subject's arrival at the source. The between-groups independent variables were the sound propagation systems (geometric/wave-based) in the first experiment and wave-based spatial sound rendering

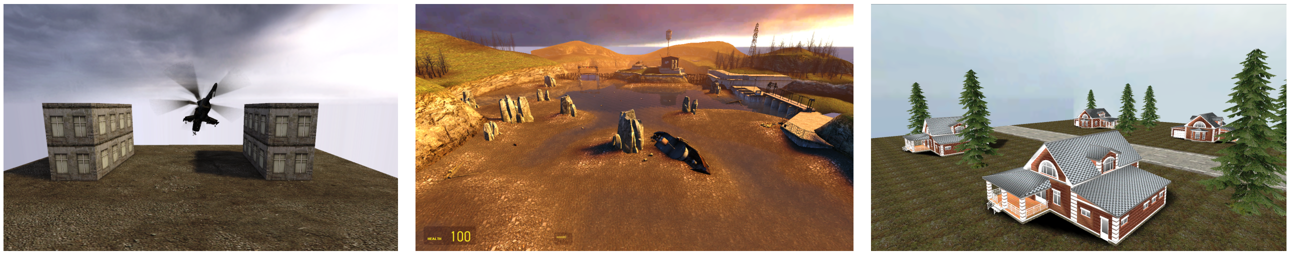

Fig. 4. Different scenes used to evaluate the performance of our sound propagation algorithm and system: (left) Parallel walls, (center) Reservoir, and (right) Suburban scene.

(active/inactive) in the second experiment. The dependent variable in both was the task completion time. We chose between-group study design instead of within-group to avoid any learning effects in the subjects.

### 6.2 Scenario

We created a 3D scene in the shape of a typical suburban street with houses (see Figure 4, the "suburban scene"). The houses act as obstacles for sound, generating multiple propagation effects including reflections, diffraction, and scattering. The scene was set up in the Valve's Source$^{TM}$ game engine. The subject's character is spawned behind one of the houses, and the sound source (a radio set playing music) is placed behind another house. The same dry audio signal is used for all three user groups. The subject has a first-person view of the virtual environment, rendered to the subject using the HMD. Task completion time was measured by triggering start and finish events in the game engine code and using timing counters to measure the difference.

### 6.3 System Details

We chose the commercial system Acoustect SDK developed by Impulsonic$^{TM}$ as the geometric acoustic technique for the first user group in the study. Acoustect SDK is a geometric acoustic propagation system developed for interactive applications [5] . It uses the image source method [4] for simulating reflections in the environment and a ray tracing algorithm to reduce the number of image sources that need to be created. In addition to reflections, Acoustect SDK also incorporates edge diffraction based on the Uniform Theory of Diffraction [39]. Our WAVE system was used as the wave-based approach for the second and third user groups. For the second user group, spatial audio rendering of the WAVE system was *disabled* and monaural audio was delivered to the subjects. For the third user group, spatial audio rendering of the WAVE system was enabled and binaural audio was generated. In order to make a fair comparison, we maintained a similar runtime performance for the Acoustect system and our WAVE system. We achieved this by choosing the following parameters for the Acoustect system: number of primary rays: 1024, number of secondary rays: 32, maximum order of reflections: 4, and maximum order of diffractions: 2. To ensure the validity of ray approximation (object size $>>$ wavelength) for the geometric system, we used a simplified model of the 3D scene.

### 6.4 Procedure

The study was conducted on a total 30 subjects, all between the age of 19 and 34. There were 27 males and 3 females, and the mean age of the group is 25.5 years. All the subjects had normal hearing and normal or corrected-to-normal vision. 13 subjects had prior experience with VR environments, either with HMDs or CAVE. Before starting the study, the subjects filled in a background questionnaire and were given detailed instructions. All the subjects completed the study.

The subjects participated in two training sessions: one to get them comfortable with the HMD and with navigation using the Xbox controller, and one to familiarize them with spatial audio. In the first training session, we asked the subjects to walk around in the virtual environment. We allowed the subjects as much time as they needed, until they were able to navigate fluidly. In the second training session, we asked the subjects to navigate in the direction they perceived the sound to be coming from. We ensured that the subjects were "sufficiently trained" by making them navigate a training map and measuring the time it took each user to navigate the entire map. If the time taken was less than the base-case time, we asked the user to repeat the training. The base-case time was computed as the average time taken by skillful users who are adept at using HMDs and game controllers. In our experience, it took the users 2-3 iterations to achieve the base-case time. Next, the subjects were randomly sorted into three experiment groups of 10 subjects each. The subject performed the main task and their task-completion times were recorded. The subjects were allowed to take as much time as needed.

### 6.5 Research Hypothesis

This study had two main research hypotheses: (1) That in the first experiment, the performance of the users using our wave-based system would be considerably higher than those using the geometric acoustic system. This would imply that the acoustic effects produced by our system result in better task performance in the VR environment than those produced by the geometric acoustics-based propagation system. (2) That in experiment two, subjects using the spatial audio rendering in our WAVE system would outperform those using the monaural audio rendering. This would mean that the users were able to perceive and utilize the directional cues inherent in wave-based spatial audio to more quickly navigate to the source position.

## 7 Results and Discussions

Figure 4 shows the different scenes used to demonstrate the performance of our WAVE system. Please refer to the supplementary video to listen to the auralizations. In Table 1, we show the precomputation timings of WAVE system. This includes the time to compute the per-object and inter-object transfer functions and the dynamic transfer operator. Table 2 demonstrates the runtime performance and memory requirements of our sound simulation system.

The audio processing pipeline consists of two stages: sound propagation and spatialization; these stages correspond to two filters which are updated asynchronously. In the case of source or listener motion, the sound propagation filter is recomputed for the new positions. This computation is performed within 56-126 ms which is sufficient for updating acoustic effects in audio applications [23]. Spatialization requires computation of the spherical harmonic (SH) coefficients of the plane-wave decomposition and the HRTF, and a dot product between these SH coefficients (Section 4.3.3) . Source or listener motion requires a recomputation of SH coefficients of the plane-wave decomposition and a dot product which can take from 6 ms to 30 ms (Table 2 in Mehra et al. [18]). Listener head rotation, on the other hand, only requires an update of the SH coefficients of the HRTF and a dot product. This can be performed by efficient SH rotation techniques in a few ms.

The exact size of the dynamic transfer operator depends on following parameters: (a) the number of objects in the scene (b) the complexity of sound propagation in the scene, and (c) simulation frequency.

| ESM<br>scenes | air<br>volume | surface<br>area | # freq. | WAVE-sim<br>(wall-clk) |
|---|---|---|---|---|
| Parallel walls | $85^3m^3$ | $142m^2$ | 250 | 210 min |
| Suburban scene | $85^3m^3$ | $558m^2$ | 250 | 1400 min |
| Reservoir | $85^3m^3$ | $950m^2$ | 250 | 850 min |

Table 1. **Precomputation cost** Abbreviations: “#freq.” is the number of frequency samples in the range (0-1 kHz), and “WAVE-sim” is the total wall-clock time to compute the per-object and inter-object transfer functions and the dynamic transfer operators. The sound-propagation computations are performed in parallel for all the frequencies on a 64-node cluster with 8 cores per node.

| Scene | pres. eval. | mat-vec | total | compr.<br>effic. | memory |
|---|---|---|---|---|---|
| Parallel walls | 20 ms | 36 ms | 56 ms | 20X | 0.2 GB |
| Suburban scene | 30 ms | 56 ms | 86 ms | 27X | 0.5 GB |
| Reservoir | 50 ms | 76 ms | 126 ms | 6X | 15 GB |

Table 2. **Runtime performance:** Abbreviations: “pres. eval” denotes the computation time for the incoming sound field of the source, “mat-vec” is the matrix-vector multiplication time, and “total” shows the total computation time. “compr. effic.” denotes the compression efficiency of the SVD on dynamic transfer operator, and “memory” is total runtime memory used. WAVE is able to achieve more than one order of magnitude compression efficiency, resulting in significantly reduced runtime memory requirement.

To give an idea of the size variation of the dynamic transfer operator (without compression), it varies from 50 x 300 at 20 Hz to 5000 x 15000 at 1 kHz for the Parallel Walls scene. The Parallel walls scene has 2 objects, suburban scene has 4 objects and reservoir has 5 objects in total. Figure 5 shows the decay rate of the singular values of the dynamic transfer operator. This result demonstrates that a small number of singular values and vectors are able to capture most of the energy (magnitude). Therefore, an SVD-based compression technique works efficiently for compressing the dynamic transfer operator (as shown in Table 2). Figure 6 demonstrates the observed speed-up using the work distribution scheme; with all matrix-vector multiplications performed on the CPU, it takes somewhere between 300ms to 350ms (maximum value of the plot), while an optimal work-distributed configuration based on our optimization takes around 50ms (minimum value of the plot), resulting in a speed-up of 6X to 7X.

### 7.1 User Evaluation Results

In Figure 7, we show the average task completion time for the different sound simulation systems used in the study: Acoustect system, our WAVE system without spatial audio, and our WAVE system with spatial audio. In experiment 1, there was 27% increase in the task performance, which is consistent with hypothesis 1. In experiment 2, subjects that used our WAVE system with spatial audio had a mean time of 43sec, outperforming those with non-spatial auralization (mean time of 60sec). This demonstrates a 28% further increase in the task performance. This is consistent with hypothesis 2. We conducted independent t-tests for both the experiments. For experiment 1, we found the t-ratio equal to +1.82 and a p-value of 0.043. In experiment 2, the t-ratio was +1.77 with a p-value of 0.047. Along with the independent t-tests, ANOVA tests were performed: F(2,27)=6.45, p-value = 0.005132. This shows the statistical significance of the results for both the experiments.

Our explanation for these results is as follows: A typical geometric system cannot model wave effects (such as diffraction) accurately; this creates discontinuity in the perceived sound field including, in many cases, no sound at all in occluded regions. Because of this, participants using the geometric system had trouble locating the sound source causing reduced task performance. In the second experiment, spatial audio added directional cues to the propagated sound, making it easier for the participants to localize the direction of sound and navigate to the sound source.

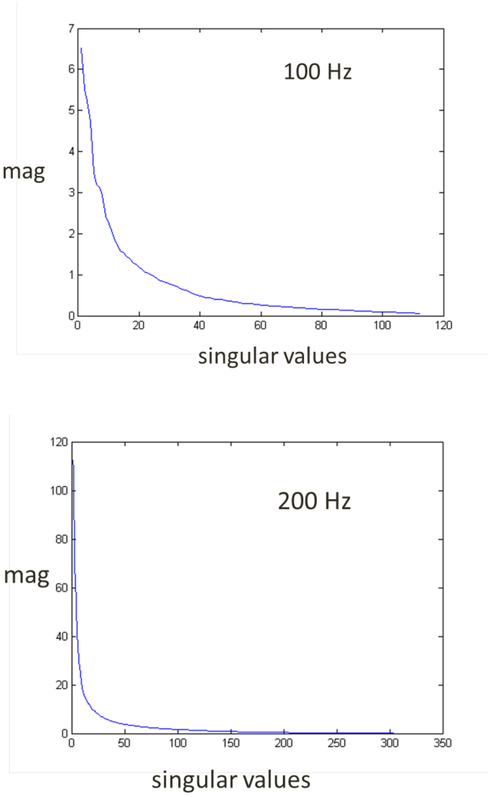

Fig. 5. Decay of the singular values of the dynamic transfer operator. This demonstrates that most of the energy is captured by small number of singular vectors and the dynamic transfer operator is amenable to SVD-based compression.

Note that, even though our WAVE system has higher precomputation costs (as do other wave-based methods) than the geometric system, our main goal was to compare users’ navigation performance for a wave-based system and a geometric system with *similar* runtime performance. A higher precomputation cost is necessary to generate various wave effects (diffraction, scattering, and interference), effects which cannot be computed accurately by current geometric techniques. As shown in Larsson et al. [15], acoustics cues are necessary to achieve sense of presence in virtual environments. We believe that wave-based effects form an important part of these acoustic cues and need to be modeled accurately in virtual environments. Precomputation-based approaches, such as global illumination, are widely used for visual effects in games and computer graphics applications. Our wave-based sound simulation technique is similar in flavor to these global illumination approaches which also have a heavy precomputation stage and an interactive runtime stage. As demonstrated by these global illumination approaches, the benefit of an accurate solution justifies the precomputation cost involved. Based on the auralization shown in the supplementary video and the results of the user evaluation, we claim a similar benefit for accurate wave-based approaches over other approximate techniques (such as geometric methods).

### 7.2 Comparison

In this section, we compare our propagation technique, built on dynamic transfer operator, to prior wave-based propagation techniques.

The source directivity technique proposed in Mehra et al. [18] pre-

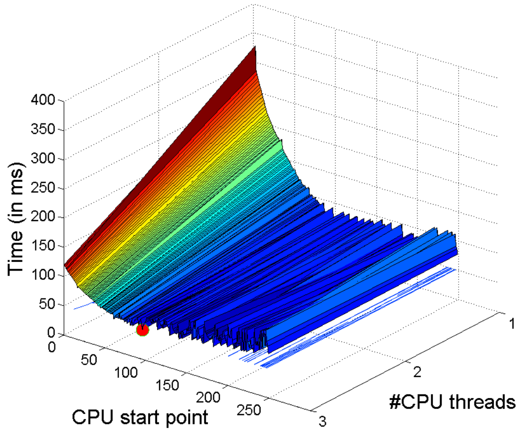

Fig. 6. Optimization for the parameters of the hybrid CPU-GPU matrix-vector multiplication method for the “parallel walls” scene.

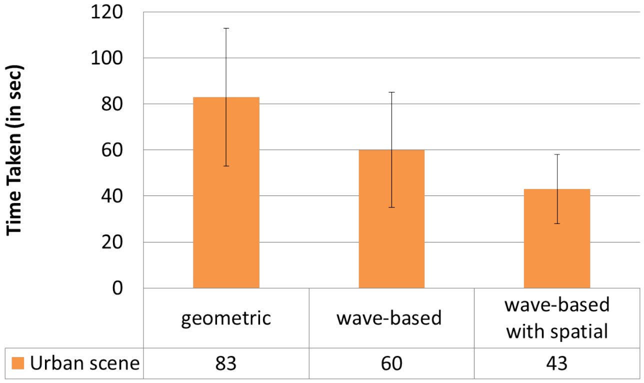

Fig. 7. Average task completion time (numbers below the bars) and standard deviation (lines on the bars) of the subjects’ navigation timings for the Acoustect system, our WAVE system with no spatial sound, and our WAVE system with spatial sound. The user’s performance increases as we go from Acoustect geometric system to our WAVE system without spatial sound and then to our WAVE system with spatial sound as exemplified by reduction in task completion time.

| Scene | Ragh. et al.[29] (sampl.) | Mehra et al.[18] (sampl.) | Ours (sampl.-free) |
|---|---|---|---|
| Parallel | 14 TB | 35 TB | 0.2 GB |
| Suburban Scene | 14 TB | 118 TB | 0.5 GB |
| Reservoir | 14 TB | 134 TB | 15 GB |

Table 3. Runtime memory requirements for sound propagation for dynamic sources using the sampling-based approach by Raghuvanshi et al. [29] and Mehra et al. [18], and our WAVE system at maximum simulation frequency $\nu_{\max} = 1018Hz$. The spatial resolution for the prior techniques is chosen to be 1m as proposed in Raghuvanshi et al. [29]. Our sound simulation system is at least three orders of magnitude more memory efficient as compared to prior techniques.

computes and stores the propagated sound field due to elementary spherical harmonic sources during the preprocessing stage. Then, at runtime, the time-varying source directivity is decomposed into respective SH coefficients and linearly combined with the stored propagated fields to generate the sound field due to directional source. The memory consumption of their technique grows as the *square* of the spherical harmonic order; it becomes too expensive for sharp directivity patterns as the SH order increases. Our approach solves the wave-equation at runtime using the dynamic transfer operator and does not need to precompute/store the propagated sound fields.

For the problem of dynamic (moving) sources, earlier work [29, 18] relied on a sampling-and-interpolation based approach that is feasible only for sources or listeners moving on a 2D plane. In Table 3, we show the runtime memory requirements of previous approaches for handling fully dynamic sources and listeners in three dimensions. Raghuvanshi et al. [29] reported a memory requirement of 620/155 = 4 MB per source sample for a scene of size 28m×60m×32m (Citadel) with 155 source samples placed on 2D plane. Memory requirements scale linearly with scene volume and source samples. Our scenes have size 85m×85m×85m and this approach would need 614,125 samples for 1 m sampling of source locations with 44 MB per source sample on our scenes; the storage overhead for 614,125 $m^3$ would be 14 TB. The memory overhead of Mehra et al.[18] is 60 MB per source for parallel walls scene. The size of the scene is 85m×85m×85m and this approach would need 614,125 samples for 1 m sampling of source locations. Therefore, total memory would be would be 35 TB. Similar estimates can be computed for other scenes. We also show the memory requirements of our dynamic transfer operator-based approach for the same scenes. Our sound simulation system is at least three orders of magnitude more memory efficient than previous approaches on these scenarios. Also, our system uses a sampling-free approach which performs sound propagation for exact positions of dynamic sources and listeners.

## 8 Conclusion and Future Work

In this paper, we have presented an accurate wave-based sound simulation system (WAVE) to generate wave-based acoustic effects for VR applications. We proposed a novel dynamic transfer operator-based approach that enabled sound propagation for fully dynamic sources and listeners, source directivity, and spatial audio in VR applications. We showed the benefit of our simulation system over prior wave-based techniques in terms of runtime memory. In addition to this, user evaluation results demonstrated improvement in task performance with our accurate, wave-based system as compared to a commercial geometric simulation system.

Currently, our dynamic transfer operator-based approach is limited to static environments. The runtime overhead of our approach increases with the maximum simulation frequency and currently requires GBs of memory. In future work, we would like to explore perceptually-driven compression schemes to further reduce the memory requirements [28]. Our wave-based method uses spectral extrapolation to generate full frequency spectra; we would like to combine our wave-based approach with the hybrid propagation models [42]. We would also like to evaluate our system for VR training simulations and integrate it with other VR systems. Our preliminary evaluation of wave-based vs. geometric simulation systems only demonstrates quantitative benefits in a single scenario. In future, we would like to conduct a detailed evaluation with more scene configurations and additional conditions. We would also like to incorporate additional evaluation strategies, such as a presence questionnaire, and conduct user evaluations with a larger number of users to better assess our method’s qualitative benefits.

## Acknowledgments

The authors wish to thank Anish Chandak and Lakulish Antani for simulating discussions, and the anonymous reviewers for comments and feedback. We would like to thank anonymous users for participating in the user study. This research was supported in part by the Link Foundation Fellowship in Advanced Simulation and Training, ARO Contracts W911NF-12-1-0430, W911NF-13-C-0037,W911NF-14-1-0437, and the National Science Foundation (NSF awards 1320644 and 1305286). The authors wish to thank Impulsonic Inc. for providing us with the Acoustect SDK. We would also like to thank Valve Corporation for permission to use the Source SDK and Half-Life 2 artwork for the Reservoir and Parallel walls scenes.

## References

[1] V. Algazi, R. Duda, D. Thompson, and C. Avendano. The CIPIC HRTF database. In *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, pages 99 –102, 2001.

[2] X. Amatriain, J. Castellanos, T. Höllerer, J. Kuchera-Morin, S. T. Pope, G. Wakefield, and W. Wolcott. Computer music modeling and retrieval. sense of sounds. chapter Experiencing Audio and Music in a Fully Immersive Environment, pages 380–400. Springer-Verlag, Berlin, Heidelberg, 2008.

[3] D. R. Begault. *3D Sound for Virtual Reality and Multimedia*. Academic Press, 1994.

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

[5] A. Chandak, L. Antani, and D. Manocha. Ipl sdk: Software development kit for efficient acoustics simulation. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 2012, pages 7938–7949. Institute of Noise Control Engineering, 2012.

[6] A. Cheng and D. Cheng. Heritage and early history of the boundary element method. *Engineering Analysis with Boundary Elements*, 29(3):268–302, Mar. 2005.

[7] W. P. de Bruijn and M. M. Boone. Application of wave field synthesis in life-size videoconferencing. In *Audio Engineering Society Convention 114*. Audio Engineering Society, 2003.

[8] 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 *ACM SIGGRAPH*, pages 21–32, 1998.

[9] T. Funkhouser, N. Tsingos, and J.-M. Jot. Survey of methods for modeling sound propagation in interactive virtual environment systems. *Presence and Teleoperation*, 2003.

[10] H. Hacihabiboglu, B. Gunel, and A. Kondoz. Time-domain simulation of directive sources in 3-d digital waveguide mesh-based acoustical models. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(5):934–946, 2008.

[11] C. Hendrix and W. Barfield. The sense of presence within auditory virtual environments. *Presence: Teleoperators and Virtual Environments*, 5(3):290–301, 1996.

[12] D. E. Hughes, J. Thropp, J. Holmquist, and J. M. Moshell. Spatial perception and expectation: Factors in acoustical awareness for mout training. *24th US Army Science Conference: Transformational Science and Technology for the Current and Future Force*, pages 339–343, 2006.

[13] D. L. James, J. Barbič, and D. K. Pai. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 987–995, New York, NY, USA, 2006. ACM.

[14] P. Larsson, D. Vastfjall, and M. Kleiner. Better presence and performance in virtual environments by improved binaural sound rendering. In *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, Jun 2002.

[15] P. Larsson, D. Västfjäll, and M. Kleiner. Effects of auditory information consistency and room acoustic cues on presence in virtual environments. *Acoustical Science and Technology*, 29(2):191–194, 2008.

[16] T. Lokki, M. Grhn, L. Savioja, and T. Takala. A case study of auditory navigation in virtual acoustic environments. In *PROCEEDINGS OF INTL. CONF. ON AUDITORY DISPLAY (ICAD2000*, pages 145–150, 2000.

[17] J. M. Loomis, R. G. Golledge, and R. L. Klatzky. Navigation system for the blind: Auditory display modes and guidance. *Presence: Teleoper. Virtual Environ.*, 7(2):193–203, Apr. 1998.

[18] R. Mehra, L. Antani, S. Kim, and D. Manocha. Source and listener directivity for interactive wave-based sound propagation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):495–503, April 2014.

[19] R. Mehra, N. Raghuvanshi, L. Antani, A. Chandak, S. Curtis, and D. Manocha. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Trans. Graph.*, Apr. 2013.

[20] J. Meyer and U. Hansen. *Acoustics and the Performance of Music (Fifth edition)*. Lecture Notes in Mathematics. Springer, 2009.

[21] C. D. Murray, P. Arnold, and B. Thornton. Presence accompanying induced hearing loss: Implications for immersive virtual environments. *Presence: Teleoper. Virtual Environ.*, 9(2):137–148, Apr. 2000.

[22] M. Ochmann. The full-field equations for acoustic radiation and scattering. *The Journal of the Acoustical Society of America*, 105(5), 1999.

[23] D. W. G. of Interactive 3D Audio SIG. Interactive 3d audio rendering guidelines, level 2.0, 1999.

[24] K. Ozawa, Y. Chujo, Y. Suzuki, and T. Sone. Psychological factors involved in auditory presence. *Acoustical Science and Technology*, 24(1):42–44, 2003.

[25] C. Pellerin. Immersive technology fuels infantry simulators. *American Forces Press Service*, 2011.

[26] A. D. Pierce. *Acoustics: An Introduction to Its Physical Principles and Applications*. Acoustical Society of America, 1989.

[27] B. Rafaely and A. Avni. Interaural cross correlation in a sound field represented by spherical harmonics. *The Journal of the Acoustical Society of America*, 127(2):823–828, 2010.

[28] N. Raghuvanshi and J. Snyder. Parametric wave field coding for precomputed sound propagation. *ACM Trans. Graph.*, 33(4):38:1–38:11, July 2014.

[29] N. Raghuvanshi, J. Snyder, R. Mehra, M. C. Lin, and N. K. Govindaraju. Precomputed Wave Simulation for Real-Time Sound Propagation of Dynamic Sources in Complex Scenes. *SIGGRAPH 2010*, 29(3), July 2010.

[30] C. Schissler, R. Mehra, and D. Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Trans. Graph.*, 33(4):39:1–39:12, July 2014.

[31] S. Siltanen, T. Lokki, and L. Savioja. Frequency domain acoustic radiance transfer for real-time auralization. *Acta Acustica united with Acustica*, 95(1):106–117, 2009.

[32] A. Southern, S. Siltanen, and L. Savioja. Spatial room impulse responses with a hybrid modeling method. In *Audio Engineering Society Convention 130*, May 2011.

[33] J. P. Springer, C. Sladeczek, M. Scheffler, J. Hochstrate, F. Melchior, and B. Frohlich. Combining wave field synthesis and multi-viewer stereo displays. In *Virtual Reality Conference, 2006*, pages 237–240. IEEE, 2006.

[34] 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.

[35] A. Taflove and S. C. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method, Third Edition*. Artech House Publishers, 3 edition, June 2005.

[36] M. T. Taylor, A. Chandak, L. Antani, and D. Manocha. RESound: interactive sound rendering for dynamic virtual environments. In *MM '09: ACM Multimedia*, pages 271–280, New York, NY, USA, 2009. ACM.

[37] L. L. Thompson. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America*, 119(3):1315–1330, 2006.

[38] N. Tsingos, C. Dachsbacher, S. Lefebvre, and M. Dellepiane. Instant Sound Scattering. In *Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering)*, 2007.

[39] N. Tsingos, T. Funkhouser, A. Ngan, , and I. Carlbom. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Computer Graphics (SIGGRAPH 2001)*, August 2001.

[40] A. Väljamäe, P. Larsson, D. Västfjäll, and M. Kleiner. Auditory presence, individualized head-related transfer functions, and illusory ego-motion in virtual environments. In *Proc. of 7th Annual Workshop Presence*, 2004.

[41] M. C. Vigeant. Investigations of incorporating source directivity into room acoustics computer models to improve auralizations. *The Journal of the Acoustical Society of America*, 124(5):2664–2664, 2008.

[42] H. Yeh, R. Mehra, Z. Ren, L. Antani, D. Manocha, and M. Lin. Wave-ray coupling for interactive sound propagation in large complex scenes. *ACM Trans. Graph.*, 32(6):165:1–165:11, 2013.

[43] D. N. Zotkin, R. Duraiswami, and N. A. Gumerov. Plane-wave decomposition of acoustical scenes via spherical and cylindrical microphone arrays. *IEEE Trans. Audio, Speech and Language Processing*, 18, 2010.