Efficient HRTF-based Spatial Audio for Area and Volumetric Sources

Carl Schissler, Aaron Nicholls, and Ravish Mehra

Fig. 1: Our spatial audio technique can efficiently produce plausible sound for large complex area-volumetric sources in interactive scenes: (left) Waterfalls; (right) Island.

Abstract— We present a novel spatial audio rendering technique to handle sound sources that can be represented by either an area or a volume in VR environments. As opposed to point-sampled sound sources, our approach projects the area-volumetric source on the spherical domain centered at the listener and represents this projection area compactly using the spherical harmonic (SH) basis functions. By representing the head-related transfer function (HRTF) in the same basis function, we demonstrate that spatial audio which corresponds to an area-volumetric source can be efficiently computed as a dot product of the SH coefficients of the projection area and the HRTF. This results in an efficient technique whose computational complexity and memory requirements are independent of the complexity of the sound source. Our approach can support dynamic area-volumetric sound sources at interactive rates. We evaluate the performance of our technique on large complex VR environments and demonstrate significant improvement over the naïve point-sampling technique. We also present results of a user evaluation, conducted to quantify the subjective preference of the user for our approach over the point-sampling approach in VR environments.

Index Terms—Spatial audio, HRTF, area sources, volumetric sources, spherical harmonics

1 Introduction

Renewed interest in virtual reality (VR) and the introduction of head-mounted displays with low-latency head tracking necessitate high-quality spatial audio effects. Spatial audio gives the listener a sensation that sound comes from a particular direction in 3D space and helps to create immersive virtual soundscapes [5].

A key component of spatial audio is the modeling of *head-related transfer functions* (HRTF). An HRTF is a filter defined over the spherical domain that describes how a listener's head, torso and ear geometry affects incoming sound from all directions [6]. The filter maps incoming sound arriving towards the center of the head to the corresponding sound received by the user's left and right ears. In order to auralize the sound for a given source direction, an HRTF filter is computed for that direction, then convolved with dry input audio to generate binaural audio. When this binaural audio is played over headphones, the listener hears the sound as if it comes from the direction of the sound source.

- *Carl Schissler is with Oculus & Facebook and UNC Chapel Hill. E-mail: carl.schissler@gmail.com.*
- *Aaron Nicholls is with Oculus & Facebook. E-mail: aaron.nicholls@oculus.com.*
- *Ravish Mehra is with Oculus & Facebook. E-mail: ravish.mehra@oculus.com*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx/

Although many current systems support the use of HRTFs for point sources [23], few systems handle sound sources represented by an emissive area or volume in space. In these scenarios, the sound heard by the listener is a combination of sound from many directions, each with a different HRTF filter. For instance, an area sound source such as a river emits sound from the entire water surface. This gives the listener the impression that the source is extended in space along the direction of the river's flow, rather than being localized at a single point. In a forest, a listener might hear wind blow through the trees, creating a broad soundscape.

Environments such as rivers or forests contain large area or volume sound sources that are difficult to recreate with traditional spatial audio techniques. A naïve point-sampling approach might approximate a large source by a collection of many evenly-distributed point sources. The spatial audio filter would then be computed as a weighted sum of the interpolated HRTF filters for the direction corresponding to each point source. However, this method is impractical for very large sources. For the island scene (Figure 1, right), the ocean coastline sound source occupies an area of roughly 100,000m^2. Representing this sound source at a 1 meter resolution with points would require about 100,000 point sources and would take at least 600 ms to compute. Users perceive this latency in head-tracked spatial audio in terms of source-lag and source-motion as the users's head rotates and this latency significantly detracts from users' experience in interactive VR applications [10, 9]. On the other hand, a coarser point sampling would cause audible artifacts in the sound: When the listener is close to or inside the sound source, they would hear discrete sound coming from closest point sources rather than accurate directional audio coming from a region of space. A more sophisticated approach is needed

to handle these challenging sources at the low latency required for head-tracked spatial audio in virtual reality applications.

We present a novel technique for computing spatial audio for area and volumetric sound sources in dynamic VR environments. The key contributions of this work are:

1. **Source complexity-independent** spatial audio technique whose computational and memory requirements are independent of the area or volume of the source.
2. A novel **analytical formulation** to compute HRTF-based spatial audio for spherical sound sources.
3. **Fast update rate and low latency** spatial audio rendering that enables **dynamic area-volumetric sound sources**.

We have implemented this technique on consumer VR hardware and the Unity game engine. Our system can compute HRTF filters for large complex sound sources such as the ocean (Figure 1) in less than a millisecond. We have also conducted a user evaluation to study the effect of area-volumetric sources on user's subjective preference in virtual environments.

2 Related Work

2.1 Head-related Transfer Function (HRTF)

The HRTF uses a linear filter to map the sound arriving from a direction (θ,ϕ) at the center of the head to the sound received at the entrance of each ear canal of the listener. In spherical coordinates, the HRTF is a function of three parameters: azimuth ϕ, elevation θ, and either time t or frequency ν. We denote the time-domain HRTF for the left and right ears as $h_L(\theta,\phi,t)$ and $h_R(\theta,\phi,t)$. The frequency-domain HRTF is denoted by $h_L(\theta,\phi,\nu)$ and $h_R(\theta,\phi,\nu)$. In the frequency domain, the HRTF filter can be stored using the real and imaginary components of the Fourier transform of the time-domain signal, or can be represented by the magnitude response and a frequency-independent inter-aural delay. In the second case, a causal minimum-phase filter can be constructed from the magnitude data using the min-phase approximation [19] and the inter-aural delay. HRTFs are typically measured over evenly-spaced directions in anechoic chambers using specialized equipment. The output of this measurement process is an impulse response for each measured direction (θ_i,ϕ_i). We refer to this HRTF representation as a *sampled* HRTF. Another possible HRTF representation is one where the sampled HRTF data has been projected into the *spherical harmonic* basis [14, 27].

2.2 Spatial audio techniques

Spatial audio techniques aim to approximate the human auditory system by filtering and reproducing sound localized in 3D space. The human ear determines the location of a sound source by considering the differences between the sound heard at each ear. Interaural time differences (ITD) occur when sound reaches one ear before the other, while interaural level differences (ILD) are caused by different sound levels at each ear [6]. Listeners use these cues for localization along the left-right axis. Differences in spectral content, caused by filtering of the pinnae, resolve front-back and up-down ambiguities.

Amplitude Panning: The simplest approaches for spatial sound are based on amplitude panning, where the levels of the left and right channels are changed to suggest a sound source that is localized toward the left or right. However, this stereo sound approach is insufficient for front-back or out-of-plane localization. Conversely, *Vector-based amplitude panning* (VBAP) allows panning among arbitrary 2D or 3D speaker arrays [26].

HRTF-based rendering: To compute spatial audio for a point sound source using the HRTF, we first determine the direction from the center of the listener's head to the sound source (θ_S,ϕ_S). Using this direction, the HRTF filters $h_L(\theta_S,\phi_S,t)$ and $h_R(\theta_S,\phi_S,t)$ for the left and right ears are interpolated from the nearest measured impulse responses. If the dry audio for the sound source is given by $s(t)$, and the sound source is at a distance d_S from the listener, the sound signals at the left ear $p_L(t)$ and the right ear $p_R(t)$ can be computed as follows:

$$p_L(t) = \frac{1}{1+d_S^2} h_L(\theta_S,\phi_S,t) \bigotimes s(t) \quad (1)$$

$$p_R(t) = \frac{1}{1+d_S^2} h_R(\theta_S,\phi_S,t) \bigotimes s(t) \quad (2)$$

where $\bigotimes$ is the convolution operator and $\frac{1}{1+d_S^2}$ is the distance attenuation factor. If there are multiple sound sources, the signals for each source are added together to produce the final audio at each ear. For the sake of clarity, from this point forth, we drop the subscripts L and R of the HRTF. The reader should assume that the audio for each ear can be computed in the same way.

Ambisonics: Ambisonics is a spatial audio technique first proposed by that Gerzon [15] that uses first-order plane wave decomposition of the sound field to capture a playback-independent representation of sound called the *B-format*. This representation can then be decoded at the listener's playback setup which can be either headphones, 5.1, 7.1 or any general speaker configuration.

Wave-field Synthesis Wave-field synthesis is a loudspeaker-based technique that enables spatial audio reconstruction that is independent of listener-position. This approach typically requires hundreds of loudspeakers and used for multi-user audio-visual environments [31].

2.3 Area-volumetric Sources

Audio: Previous work on sound for virtual scenes has frequently focused on point sources. Although directional sound sources can be modeled for points in the far-field [21], these approaches cannot produce the near-field effects of large area or volume sources. The diffuse rain technique [28] computes an approximation of diffuse sound propagation for spherical, cylindrical, and planar sound sources, but does not consider spatial sound effects. Other approaches approximate area or volume sources using multiple sound emitters, or use the closest point on the source as a proxy when computing spatial audio. However, none of these techniques accurately model how an area-volumetric sound source interacts with the HRTF to give the impression of an extended source.

Graphics: The challenge of computing spatial audio for area or volume sound sources is similar to the challenge of computing direct illumination from area light sources. Because closed-form solutions for most light geometries do not exist, numerical approaches like Monte Carlo ray tracing are preferred [12, 29]. Radiosity algorithms face a similar challenge when determining the form factor for a surface element. In this case, the light recieved from all visible surface patches must be considered. A common solution is to use *hemi-cube* rasterization to project the contribution of each patch onto a 5-faced half-cube surrounding the point of interest [11]. Although these approaches work well for representing illumination, they cannot adequately capture spatial audio for area-volumetric sources because they do not incorporate the directionally-varying phase and frequency filtering inherent in HRTFs.

2.4 Spatial audio perception

A key goal for VR systems is to help users to achieve a sense of presence in virtual environments. Experimentally, self-reported levels of immersion and/or presence have been shown to increase or decrease in line with auditory fidelity [17, 32]. Head-tracking and spatialization further increase self-reported realism of audio and the sense of presence [18, 17, 2]. In addition, head-tracked HRTFs greatly improve localization performance in virtual environments [2]. Multiple studies have demonstrated that with sufficient simulation quality, HRTF-based audio techniques can produce virtual sounds indistinguishable from real sound sources [20, 7].

Minimizing latency in head-tracking and audio/visual processing is one of the key challenges in virtual reality. Research has shown that

the presence of significant latency in a head-tracked auditory environment impacts localization performance and degrades reported quality of experience. Although some studies [8, 34, 35] have suggested that latencies of 150 – 500 ms in audio environments minimally impact localization performance or perceived latency, subsequent research has demonstrated a significant effect exists once timing is taken into account. Brungart et al.[10] controlled for stimulus presentation and/or reaction time and demonstrated that as latency increases above 73 ms, both localization performance and reaction speed in localization tasks degrade significantly. Furthermore, [9] demonstrated that listeners were able to consistently detect latencies greater than 82 ms. The addition of a reference sound (which provided a low-latency cue) further lowered the average latency detection threshold more than 20 ms to approximately 61 ms. Furthermore, they witnessed variability in latency detection thresholds between their subjects, and the best-performing listeners were able to reliably detect latency as low as 60 ms without a reference sound and 36 ms with one present.

In light of this, controlling latency is important for producing high-fidelity audio environments in virtual reality. Virtual environments can feature both visual cues and low-latency sounds such as unspatialized or VBAP sources, which may be presented with less latency than HRTF-processed sounds. These provide reference cues of latency in spatialized audio, similar to the reference sounds of [9]. For these reasons, a target of no more than 30 ms end-to-end latency (between head movement and corresponding update of HRTF-spatialized audio) is applicable even in virtual reality applications.

2.5 Spherical harmonics in audio

Orthogonal basis functions defined on the spherical domain have been frequently used in audio rendering. Several approaches have proposed the use of spherical harmonics for efficient HRTF representations [37, 27]. Spherical basis functions can also be used to represent the directivity of sound sources. One approach combines a set of elementary spherical harmonic source directivities to synthesize directional sound sources using a 3D loudspeaker array [33]. Noisternig et. al. [25] use the discrete spherical harmonic transform to reconstruct radiation patterns in virtual and augmented reality. In wave-based sound simulations, spherical harmonics have been used with the plane-wave decomposition of the sound field to produce dynamic source directivity as well as spatial sound [21]. These basis functions have also been used for spatial sound encoding in near-field using higher-order ambisonics [22, 13].

3 Area-volumetric Sources

In this section we describe our spatial audio technique for handling area and volumetric sound sources.

3.1 Theoretical derivation

To simplify the discussion, we start with the following scenario illustrated in Figure 2: an area-volumetric sound source (S) and a listener (L). One method to compute spatial audio produced by this source at the listener is to sample the source with N discrete points. These point sources are at a distance $[d_1, d_2, ..., d_N]$ from the listener in the directions $[(\theta_1,\phi_1),(\theta_2,\phi_2),...,(\theta_N,\phi_N)]$. The spatial audio from the collection of point sources can be computed as the summation of spatial audio produced by the individual point source (equation 1) to give:

$$p(t) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{1+d_i^2}\,h(\theta_i,\phi_i,t)\right) \bigotimes s_i(t), \tag{3}$$

where $h(\theta,\phi,t)$ and $s_i(t)$ are the HRTF and the dry audio corresponding to the point source i, respectively. The factor $(1/N)$ is applied to normalize the amplitude of the area-volumetric sound source. Under the assumption that all point sources are emitting the same dry audio (i.e. $s_i(t) = s(t)$), the above equation becomes

$$p(t) = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{1+d_i^2}\,h(\theta_i,\phi_i,t)\right) \bigotimes s(t), \tag{4}$$

$$= h_{\text{env}}(t) \bigotimes s(t), \tag{5}$$

where $h_{\text{env}}(t) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{1+d_i^2}\,h(\theta_i,\phi_i,t)$ is the spatial audio filter corresponding to the area-volumetric source. This equation shows that spatial audio filter for an extended source can be expressed as a weighted summation of the HRTFs of the constituent point sources. This is a discrete approximation and converges to the exact solution as $N \to \infty$. The continuum solution can be written as:

$$h_{\text{env}}(t) = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} f(\theta,\phi)\,h(\theta,\phi,t)\,\sin(\theta)\,d\theta\,d\phi, \tag{6}$$

where $f(\theta,\phi)$, also called the projection function, is a direction-dependent normalized weight function applied to the HRTF.

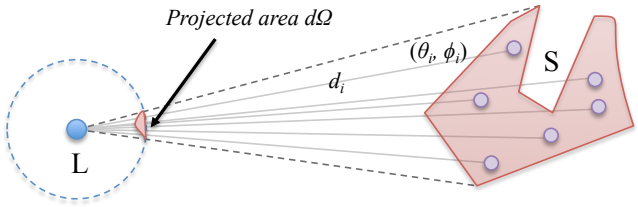


Fig. 2: Visualization to illustrate the projection of area-volumetric sound source S over a sphere around the listener L.

One way to think about the projection function is to visualize the projection of the area-volumetric source onto an imaginary sphere around the listener (Figure 2). The area-volumetric source will project onto an area of the listener's sphere (denoted by $d\Omega$). The projection function would have a non-zero value for all directions inside this projected area and have a zero value for directions outside. The projection value at any single direction inside the projection area $d\Omega$ depends on the sound radiation of the source and its distance attenuation in that direction. A significant advantage of considering the projection of the source, rather than doing a point-based sampling approach, is that the complexity of the approach is not based on the size of the sound source, only the projected area.

To compute the spatial audio filter for an area-volumetric sound source, we have to solve the integral equation (6). Solving this integral directly can be computationally expensive. The key insight of our work is to use orthonormal basis functions to solve this integral efficiently. The projection function $f(\theta,\phi)$ and HRTF $h(\theta,\phi,t)$ are both functions defined over a spherical domain (θ,ϕ). Similar to how a 1D signal can be expressed in terms of orthonormal Fourier bases, functions defined over a spherical domain can also be expressed in terms of orthonormal basis functions (see supplemental material). We express the projection function $f(\theta,\phi)$ and HRTF $h(\theta,\phi,t)$ in orthonormal basis Ψ_{lm} as follows:

$$f(\theta,\phi) = \sum_{l=0}^{n}\sum_{m=-l}^{l} c_{lm}\;\Psi_{lm}(\theta,\phi) \tag{7}$$

$$h(\theta,\phi,t) = \sum_{l=0}^{n}\sum_{m=-l}^{l} d_{lm}(t)\;\Psi_{lm}(\theta,\phi) \tag{8}$$

Using the properties of orthonormal basis functions (see supplemental material), we can simplify the projection integral equation to give:

$$h_{\text{env}}(t) = \sum_{l=0}^{n}\sum_{m=-l}^{l} c_{lm}\;d_{lm}(t). \tag{9}$$

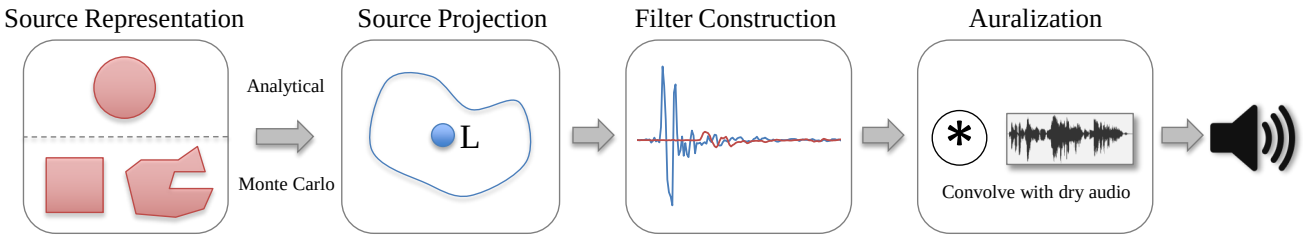


Fig. 3: Overview of our spatial sound pipeline for area and volume sound sources. The pipeline is duplicated for each sound source in a scene. At runtime, the set of shapes for a source is first projected into the spherical harmonic basis using either the analytical formulation (spheres) or Monte Carlo ray sampling (boxes, meshes). This produces a set of basis coefficients that approximate the sound contribution from all shapes of the source. Next, HRTF filters are constructed for the left and right channels based on this projection. Finally, these filters are convolved with the dry input sound to produce the final audio for that sound source.

The same derivation holds for a frequency-domain HRTF representation $h(\theta,\phi,\nu)$ or an equivalent representation (such as minphase). Therefore, the spatial audio filter corresponding to the area-volumetric source can be computed as a dot product of the basis coefficients of the projection function and the listener's HRTF. To summarize, the steps required are as follows:

1. Projection
   (a) Compute projection of the source onto the listener sphere.
   (b) Compute basis coefficients of the projection function.
2. HRTF filter construction
   (a) Take dot product of basis coefficients of projection function and HRTF function.

Note that the basis coefficients of the HRTF do not change at runtime and can be precomputed and stored. On the other hand, the basis coefficients of the projection function change with listener orientation, source-listener distance, source directivity, etc., and must be recomputed at runtime. The above equations can use any orthonormal basis functions defined for the spherical domain, such as spherical harmonics or spherical wavelets. We chose the spherical harmonics as the orthonormal basis functions in this work.

3.2 System Overview

Figure 3 shows an overview of our technique. We start with an area-volumetric sound source, represented as a collection of geometric shapes. During the preprocessing step, the spherical harmonic (SH) coefficients of the HRTF are precomputed and stored for runtime use. At runtime, given a set of input shapes that constitute an area-volumetric source, we determine the projection of each shape on the spherical domain centered at the listener. Next, the projection coefficients computed for each shape individually are then summed up for all the shapes constituting the source. The spatial audio filter for this area-volumetric source is computed as a dot product of the SH coefficients of the projection function and the HRTF. The spatial audio filter for each sound source is then convolved with the input dry audio of the source to generate spatial audio for the source. This pipeline is duplicated for each area-volumetric sound source in the scene and the spatial audio for all the sources are summed together to generate the final sound to be played over the headphones. We now discuss each step in detail.

3.2.1 Source Representation

In our technique, an area-volumetric source is defined as a collection of one or more geometric shapes that emit sound from an area or volume. During the scene design phase, an artist or a game designer could either (a) place these geometric shapes in the scene and create an area-volumetric sound source as their collection or (b) select part of the scene geometry (river, forest) and assign it as an area-volumetric sound source. The geometric shapes associated with an area-volumetric source are: (a) sphere, (c) box, and (c) arbitrary mesh. Shapes (a) and (b) are volumetric sources, whereas (c) could be an area (open mesh) or volumetric source (closed mesh). The union of multiple shapes can describe complex sound sources (see Figure 4).

For an area source, sound is emitted uniformly from all surfaces with distance attenuation based on the distance to the surface. If a sound source is a closed volume (e.g. sphere, box, arbitrary mesh), the sound is emitted uniformly within the volume, with distance attenuation outside the volume. Each area-volumetric source has a spatial audio filter (that need to be computed) and a stream of dry unprocessed audio samples. At runtime, each source results in one convolution operation between its spatial audio filter and the dry audio.

Fig. 4: This visualization shows the sound sources for the windmill and city scenes in red. In the windmill scene, box sound shapes are used to represent the windmill sails, spheres are used for trees, and a triangle mesh is used for the nearby river. In the city, the train and car sound sources are represented by boxes, while scrolling advertisements are represented using meshes that correspond to the visual geometry.

3.2.2 Source Projection

In this step, each source shape is projected into a spherical domain centered at the listener and the the spherical harmonic coefficients of the projection function are computed. In the special case of a spherical sound source, these coefficients can be computed analytically as a function of the radius of the sphere and its distance from the listener. Section 4.1 describes this analytical projection technique in detail.

In the case of a box or mesh, no such closed form solution exists and we have to compute the projection coefficients numerically. This

is computed by using an efficient Monte-Carlo integration approach (Section 4.2). The number of rays used in this approach is determined adaptively based on the size of the projection area of the area-volumetric source shape. In other words, a source shape at a greater distance has a smaller projection area, and thus fewer rays are traced compared to a nearby source shape.

3.2.3 Filter construction

The spatial audio filter construction process computes the dot product of the SH coefficients of the projection function of the source shape (determined in the previous step) with the SH coefficients of the HRTF. This step is repeated for each shape of sound source. The results sum to generate the filter for the corresponding area-volumetric source. This step is repeated for each ear to generate the spatial audio filter for the left and right ears.

3.2.4 Auralization

In this last step, the filters of the area-volumetric source are convolved with the dry audio associated with the source to generate binaural audio corresponding to that source.

4 Source Projection

The first step in computing the spatial audio filter for an area-volumetric source shape involves projecting the shape onto an imaginary sphere around the listener and computing the spherical harmonic coefficients of the projection function. Depending on the shape of area-volumetric source, this step can be performed in two ways:

4.1 Analytical Projection

In the special case of spherical source projection, the spherical harmonic coefficients of the projection function can be computed analytically. We present the main results here and strongly encourage the reader to refer the Appendix (Section 9.1) for detailed derivation.

Let's take a scenario in which we have a spherical area-volumetric source of radius R at a distance d from the listener (Figure 5). We choose the listener's coordinate frame such that it is centered at the listener with the z axis oriented toward the listener-source direction.

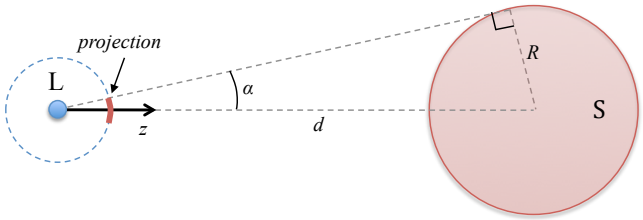


Fig. 5: The geometry of the analytical source projection for listener L and spherical source S. The projection depends solely on the angle $\alpha = \sin^{-1}\left(\frac{R}{d}\right)$, where d is the distance to the sphere's center, and R is the sphere's radius. We choose a coordinate system for the projection with the z axis oriented in the direction of the sphere in order to simplify the derivation.

Note that a sphere's projection over another sphere is a circular projection area. The radius of this projection area is independent of the orientation of the source sphere and depends only on radius R and distance d. Using trigonometry, we can relate R and d to the the half-angle of the projection, $\alpha = \sin^{-1}(R/d)$. The SH coefficients c_{lm} of the projection function $f(\theta,\phi)$ can be shown to be as follows:

$$c_{lm} = \begin{cases} 0 & : m > 0 \\ z_l & : m = 0 \\ 0 & : m < 0 \end{cases} \tag{10}$$

where z_l are the zonal harmonic coefficients defined as

$$z_l = \frac{1}{1+d^2} \quad \frac{4\pi}{1-\cos\alpha} \quad \sum_{k=0}^{l} \beta_{lk} \left[\frac{1-\cos\alpha}{k+1} - D(1,k) + D(\cos\alpha, k) \right] \tag{11}$$

Here β_{lk} is a constant and $D(a,b)$ is a function (see appendix).

When the listener's head is at a different orientation $(\theta_{list}, \phi_{list})$ with respect to the orientation assumed in Figure 5, the spherical harmonic coefficients of the projection function can be computed by using the zonal harmonics rotation equation as follows:

$$c_{lm} = \sqrt{\frac{4\pi}{2l+1}} \; z_l \; Y_{lm}(\theta_{list}, \phi_{list}). \tag{12}$$

The outcome is a set of analytically computed SH coefficients c_{lm} of the projection function that can be used to construct the spatial audio filter for a spherical sound source.

The above case only works when the listener is outside the spherical source. The case when the listener is inside must be handled separately because the value of $\alpha = \sin^{-1}\left(\frac{R}{d}\right)$ is undefined. Inside the source, sound arrives at the listener from all directions and the directivity of the source is reduced. There is a smooth transition in the spatial audio as a listener moves from the edge of the sphere toward the center, with more directivity at the edge and less near the center. To acheive this effect inside the source, we first compute the analytical SH projection coefficients c_{lm} as if the listener was at the closest point on the sphere's surface where $\alpha = \frac{\pi}{2}$. This produces coefficients with strong directivity. Then we attenuate the resulting coefficients c_{lm} by the factor $\frac{d}{R}$ for $l > 0$, leaving the DC coefficient c_{00} with constant directivity unchanged. Toward the center, $c_{lm} \to 0$ for $l > 0$. As a result the directionality of the sound source reduces naturally as the listener approaches the sphere center.

4.2 Monte Carlo Projection

While spheres are rotationally invariant and allow for an analytical projection formulation, this is not true for more complex sound sources. For computing the spherical harmonic coefficients of the projection function for a complex source, we need to solve the following integral equation:

$$c_{lm} = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} f(\theta,\phi) Y_{lm}(\theta,\phi) \sin(\theta) \, d\theta d\phi. \tag{13}$$

In this section, we describe an efficient Monte Carlo based formulation to solve this integral.

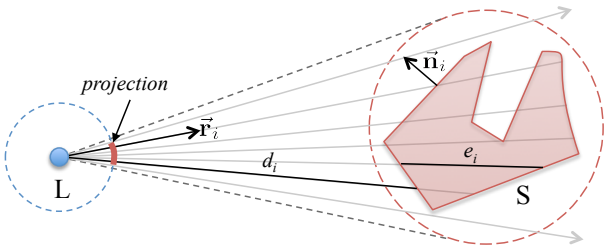


Fig. 6: The Monte Carlo projection uses rays to sample the sound contribution from arbitrarily-shaped sources. When the listener L is outside the bounding sphere of the sound source, S, we trace rays in the cone defined by the source's bounding sphere. Inside the bounding sphere, rays are traced in all directions uniformly. Each ray is given a weight w_i that is used to estimate the value of the projection integral. If a ray does not hit the source, that ray has $w_i = 0$.

4.2.1 Background

In Monte Carlo integration, a set of uniformly distributed random samples are used to numerically compute the integral of function. Each

sample is weighted according to its probability. An approximate value for the integral is computed by summing the weighted random samples. Due to the law of large numbers, the accuracy of the integral increases when more samples are taken. This approach has previously been applied for computing direct light for computer graphics [29], as well as for low-order spherical harmonic representations of lighting [16]. In our approach, we modify this formulation to efficiently compute the projection of an area-volumetric sound source.

4.2.2 Monte Carlo Projection for Arbitrary Shapes

We present a Monte Carlo numerical integration technique that computes an approximation of the SH coefficients of a source's projection function using a set of random rays. This operation is performed for each of a sound source's shapes independently and the results are added to produce the SH coefficients for the entire source. Our approach begins by generating a set of N uniformly-distributed rays with directions $\vec{\mathbf{r}}_i = (\theta_i, \phi_i)$ that sample the bounding sphere of a complex area or volumetric sound source shape. This process is illustrated in Figure 6. The rays are intersected with the geometry of the source and used to compute the projection of the source at the listener's spherical domain. Each ray is weighted by a factor f_i that specifies how much that ray contributes to the final projection. For area sound sources (e.g. triangle meshes), $f_i = f(\theta_i, \phi_i)$ is proportional to the inverse-square distance attenuation from the ray's intersection point to the listener, $\frac{1}{1+d_i^2}$, as well as the dot product of the ray direction $\vec{\mathbf{r}}_i$ with the surface normal vector $\vec{\mathbf{n}}_i$.

$$f_i = \left(\frac{1}{1+d_i^2}\right) \max(-\vec{\mathbf{r}}_i \cdot \vec{\mathbf{n}}_i,\ 0) \qquad \text{(area sources)} \qquad (14)$$

For volumetric sound sources, we choose f_i to also include the distance the ray travels through the source, e_i:

$$f_i = e_i \left(\frac{1}{1+d_i^2}\right) \max(-\vec{\mathbf{r}}_i \cdot \vec{\mathbf{n}}_i, 0) \qquad \text{(volume sources)} \qquad (15)$$

If a ray does not intersect a sound source or is blocked by an obstacle in the scene, we set $f_i = 0$ for that ray. The spherical harmonic coefficients c_{lm} of the projection can then be computed by the following equation:

$$c_{lm} = \frac{1}{\sum_{i=0}^{N} f_i} \sum_{i=0}^{N} f_i\, Y_{lm}(\theta_i, \phi_i). \qquad (16)$$

As an optimization, we trace fewer rays for distant sound sources, as shown in Figure 7. The number of rays, N, is chosen to be proportional to the solid angle of the source's bounding sphere. This saves computation for distant sound sources while maintaining the same sampling density from the listener's point of view. When the listener is inside the bounding sphere of the shape, the source is sampled using uniform random rays in all directions with the same sampling density.

5 Implementation

5.1 Hardware

Our system is implemented on a desktop machine with 3.4 GHz Intel Core i7-4930K CPU, 32 GB of RAM and NVIDIA GeForce GTX TITAN GPU. We use the Oculus Rift DK2 Head-Mounted Display (HMD) with a resolution of 960 x 1080 per eye and 100 degrees diagonal field of view as the display device. Audio is played using Sennheiser HD 700 open-ear headphones. The head position and orientation given by the HMD is used to update both the visual rendering and the spatial audio rendering.

5.2 Software

We have implemented our spatial audio system as a plugin for the Unity™ game engine. Spatial audio processing is applied to each sound source's dry audio as a custom Unity audio effect. The sound for all sources is then mixed for stereo reproduction using Unity's built-in audio mixing system. We use the standard KEMAR dataset for HRTF computation.

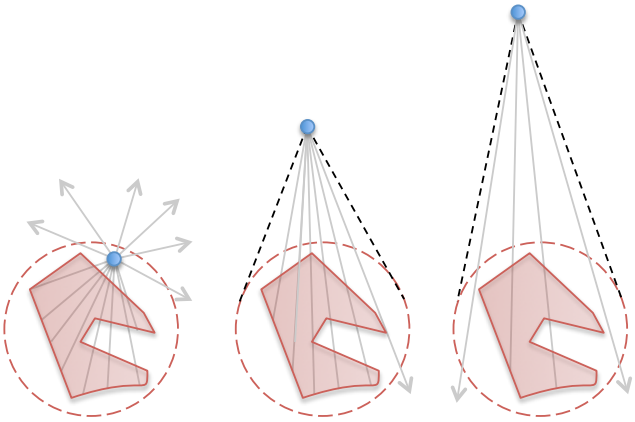


Fig. 7: The number of rays used to computed the Monte Carlo projection changes depending on the distance from the listener to the sound source's bounding sphere. If the listener is inside the bounding sphere (left), many rays are traced in all directions. Outside the bounding sphere, the number of rays that are used for Monte Carlo integration decreases proportional to the solid angle of the bounding sphere.

Spherical Harmonics (SH): For efficient evaluation of the real spherical harmonics, we use the fast SH code from [30]. This code uses aggressive constant propagation and a branchless cartesian formulation to evaluate the SH functions for fixed order n. It is more than an order of magnitude faster than naïve SH evaluation and substantially reduces the computational requirements for the Monte Carlo technique.

Rafaely and Avni [27] performed a detailed evaluation to study the effect of spherical harmonic order of the HRTF on spatial perception. Their results indicate the a minimum SH order of order 6 is required for frequencies up to 8 kHz. We choose a spherical harmonic expansion of order 9 in our implementation.

Ray Tracing: In the case of ray intersections with geometric primitives (spheres, boxes, meshes), we use specialized ray intersection tests. To handle efficient ray tracing of large mesh sound sources, we use a SIMD-aware stackless Bounding Volume Hierarchy (BVH) implementation [1].

Propagation Delay: We incorporate a simple model of sound propagation delay into our spatial sound formulation. Rather than using the actual delay to each audible point on sound source, we use the *minimum* delay to avoid comb-filtering artifacts which occur when the same dry source audio is played at slightly different delays. We compute the nearest sample on the source and use the delay to that point for sound rendering. This delay is then used in a fractional delay interpolation algorithm [36] to produce smoothly-varying sound that incorporates Doppler shifting effects. The dry sound for each source is resampled using linear interpolation and then sent to the spatial sound module for convolution with the spatial audio filter.

Convolution: To render the spatial audio for a sound source at interactive rates, we use a variant of the non-uniform partitioned block convolution algorithm [4]. The spatial audio filter is partitioned into blocks of power-of-two size, converted to frequency domain, and then convolved with a stream of frequency-domain dry source audio using an overlap-add method. To smoothly handle changes to the spatial sound filter, linear interpolation is performed in time-domain at the output of parallel convolutions of the previous and next filters [24]. Audio rendering is performed on a separate thread from the HRTF filter computation. When a new HRTF filter is ready for the

current scene state, the filter is asynchronously updated using atomic instructions.

6 User Evaluation

We have conducted a user evaluation to study the effect of area-volumetric sound sources on subjective preference of users in virtual environments. We compare the sound generated by a point-sampling technique, called the *base* method, with the sound generated by the Analytical-Monte Carlo technique (Section 4), called *our* method. In the point-sampling approach, we represent an area-volumetric source with a collection of discrete point sources. The number of point sources used to represent the area-volumetric source was chosen to ensure that the runtime computational requirements of the point-sampling technique matched our Analytical-Monte Carlo technique.

6.1 Study Design

The study uses a within-subject experiment design with an A-B session comparison protocol. The study has two comparison conditions: *base* vs. *our*, and *our* vs. *base*. Corresponding to each condition, a pair of VR sessions was generated with identical visual rendering techniques but with different spatial audio techniques. These two comparisons conditions were produced for each of 3 scenes (Island, Waterfall, Windmill), for a total of 6 scenarios. These 6 scenarios were presented to the participant in a random order. The participant was unaware as to which VR session (A and B) corresponded to which technique (*base* and *our* method).

The virtual avatar for the participant was spawned at a position in each scenario and the participant was free to move and rotate their head. The position and orientation of the participant's head were tracked by the head-mounted display and the audio and visuals were updated correspondingly. Each scenario would last for one minute and the participant had the ability to toggle between the two sessions as many times as he/she wants. After completion of each scenario, the participant answered the following subjective questionnaire (see Figure 11):

1. In which session did the spatial extent of the sound better match the visuals?
2. In which session did you feel most enveloped by the soundscape?
3. Which session did you prefer?

The responses were recorded on a 5-point Likert scale: 5 meant strong preference for Session A, 3 meant no preference and 1 meant strong preference for session B.

6.2 System Details

Visual information was presented to the participants via the Oculus Rift DK2™ head-mounted display (HMD). Sound was produced through the Sennheiser HD 700 open-ear headphones. We used the standard KEMAR HRTF dataset for auralization.

6.3 Procedure

Before the experiment, participants filled a background questionnaire and were given detailed instructions. The participants were also trained on how to wear and use the equipment and were given one trial to get acquainted with the system. Then, participants were presented the 6 scenarios in a random order and asked to rate their preference after each scenario. Participants were allowed to take a break at any time if desired. After all the 6 scenarios, the experiment was completed. All the subjects completed the study.

6.4 Research Hypothesis

The research hypotheses of this study were: 1) The proposed technique improves audio-visual spatial extent match, sense of envelopment by soundscape, and general preference, in VR environments compared to the point-sampling technique. 2) The amount of improvement depends on the type of scene and the type of area-volumetric sound sources.

7 Results and Discussion

We evaluated the performance of our technique on four scenes with varying source complexity. The timings were measured on a single CPU thread and were averaged over 1000 iterations. The performance results are summarized in Table 1. For all scenes our method can update the spatial audio filters in less than 1 ms. We break down the total time into the time spent on the analytical source projection (spheres only) and Monte Carlo projection (boxes, meshes) for each scene. The source projection time scales linearly with the number of shapes for which the projection must be computed. On the other hand, the filter construction is done only once per source. The memory usage of our technique is small. The primary cost is the HRTF storage, which uses 100KB when stored in the SH domain up to 9th order.

In Table 1, we also compare the performance of our method to a naïve point-source approximation (Equation 4). Using the area and volume of the sound sources given in Table 1, we estimate the computation time of this technique for each scene. Computing an HRTF filter for a single point source takes about 0.006 ms. If the sound sources are sampled using points at a coarse 1 meter resolution, our method outperforms point sources for all scenes. The difference is most noticeable for large sound sources in the Waterfall and Island scene. These scenes would require greater than 100 ms to compute using the point-sampling approach and would result in perceivable latency for VR applications [9]. Our approach takes less than a millisecond for these scenes.

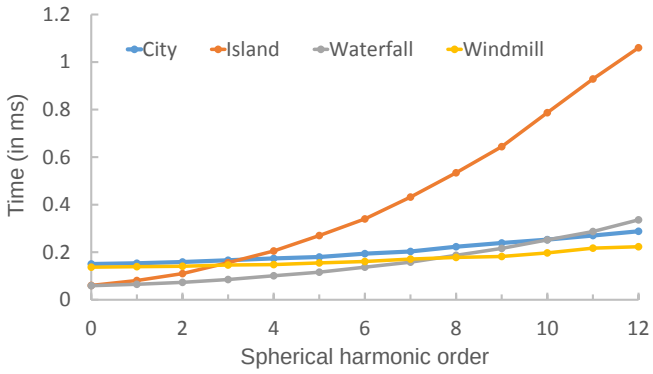


Fig. 8: The performance of our approach varies with respect to the spherical harmonic order for the four scenes.

In Figure 8 we show the performance of our approach with respect to the maximum spherical harmonic order, n, for the various scenes. Depending on the type of scene, the maximum spherical harmonic order can have a significantly effect on the time it takes to compute the spatial audio filter. For scenes where the majority of sources are spherical (Island, Waterfall), the effect is higher as the computational time of analytical projection is directly proportional to square of spherical harmonic order. On the other hand, for scenes which are dominated by box or mesh sources, the effect is significantly smaller as the computational time is dominated by the ray-tracing.

User Study: Figure 9 shows the results of our user study. The scores of the *base vs. our* condition were reversed and combined with the *our vs. base* condition. The comparison score is averaged over all the participants for each of the three questions and three scenes. A score less than 3 indicates a preference for point-sampling technique whereas a score greater than 3 indicates a preference for our Analytical-Monte Carlo technique. A score of 3 indicates no preference. Our Analytical-Monte Carlo technique performed better than the point-sampling technique for all the questions and all the scenes. These results demonstrate that participants perceive better match in the audio-visual spatial extent (of the area-volumetric source), increased sense of envelopment by the soundscape and higher preference with our technique compared to the point-sampling technique. Additionally, the amount of improvement depends on the scene type and the type of area-volumetric sound sources present in the scene.

Scene	Scene Complexity # Sources	# Shapes	Src area (m^2)	Render load (% CPU)	Our technique (ms) Analy. Proj.	M.C. Proj.	Filter const.	Total	Naïve-sampling. (ms) Total	Speedup (Naïve/Our)
City	7	B(3), M(4)	0.1K + 1.6K	3.30	-	0.09	0.09	**0.18**	**10**	**55**
Windmill	4	S(2), B(4), M(1)	8K + 1K	1.64	0.01	0.11	0.06	**0.18**	**54**	**300**
Waterfalls	4	S(10), M(2)	0 + 30K	1.57	0.13	0.13	0.05	**0.31**	**180**	**581**
Island	5	S(43)	100K + 0	2.04	0.55	-	0.07	**0.62**	**600**	**968**

Table 1: The performance results of our spatial sound system with varying number and complexity of sound sources. The number of sound source shapes in each scene is specified using the notation: S=spheres, B=boxes, M=meshes. We report the computational load of the audio rendering thread (Render load) performing the auralization step. We report the timings for both the analytical projection (Analy. Proj.) used for spherical sources and Monte Carlo (M.C. Proj) used for box and meshes along with the filter construction cost (Filter const.). For all scenes, our approach can compute spatial sound filters in less than 1 millisecond. We list the estimated volume and area of all sound sources in each scene, as well as the approximate time needed for the naive point-sampling approach. In case of latter technique, point sources were sampled at a 1 meter resolution (filter computation time per point source = 0.006 ms). Our spatial sound algorithm is 2-3 orders of magnitude faster than the naïve approach.

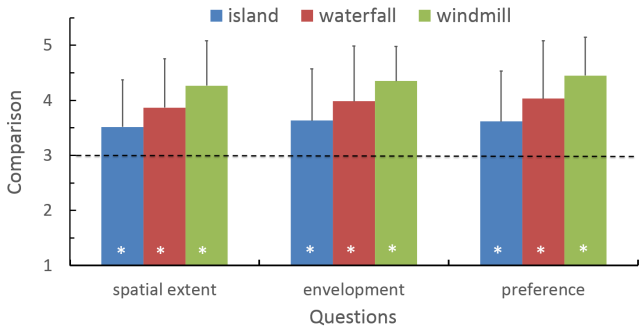


Fig. 9: User evaluation results for the subjective questionnaire. The comparison score is averaged over all the subjects and is plotted for each question and scene. Score of 1 represents a strong preference for the point-sampling technique and score of 5 represents strong preference for our analytical-monte carlo technique. The horizontal dashed line presents a score of 3 indicating no preference between the two techniques. Standard deviation is represented by the error bars. The symbol * denote the significance levels of $p < 0.001$.

8 Conclusions, Limitations, and Future Work

We have described an approach for computing spatial audio filters for scenes with large area and volumetric sound sources. Our technique is based on a listener-centric projection of the sources into the spherical harmonic basis and can compute plausible spatial audio filters in less than a millisecond, more than two orders of magnitude faster than a naïve sampling-based approach.

Our approach has some limitations. Since we perform a single projection for each source shape, we assume the sound of each shape is delayed equally, rather than in a directionally-dependent manner across the shape. Secondly, the analytical projection for spheres can become numerically unstable for small values of α. To avoid this problem, we treat very distant sources (e.g. $\alpha < 1^\circ$) as point sound sources and use double precision for the analytical projection. In addition, our current analytical approach cannot handle occlusion effects from obstacles in the scene, since it assumes the entire sphere is visible. We would like to add an occlusion factor into our projection function to incorporate this. Another limitation is aliasing of thin sources. In case of Monte Carlo approach, the projection of thin sources (e.g. a line) can lead to aliasing if none of the random rays intersect the source. This problems can be ameliorated, though not eliminated, by increasing the number of rays traced. We would like to explore more efficient formulations for such types of sources.

Secondly, we do not consider higher-order sound propagation effects like reflection and diffraction in our sound system. However, we would like to extend our approach to incorporate these phenomena in future. [3]

We would also like to conduct a detailed user evaluation with a larger number of participants and more scene configurations to better assess the qualitative benefits of our technique.

Acknowledgments

The authors wish to thank the anonymous reviewers for their constructive feedback. We would like to thank Amy Hong for her help with the demos, Tanya Micheletti, Erika Evans, and Lauri Kanerva for their help with the IRB paperwork, Tina Reinl and Lindsey Neby for their help with the user study, and Marina Zannoli for giving us feedback on the data analysis. We would also like to thank anonymous participants of the user study. This research was fully supported by Oculus & Facebook.

References

[1] A. T. Áfra and L. Szirmay-Kalos. Stackless multi-bvh traversal for cpu, mic and gpu ray tracing. In *Computer Graphics Forum*, volume 33, pages 129–140. Wiley Online Library, 2014.

[2] W. Ahnert, C. Moldrzyk, S. Feistel, T. Lentz, and S. Weinzierl. Head-tracked auralization of acoustical simulation. In *Proceedings of the 117th AES Convention*, 2004.

[3] 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, 1979.

[4] E. Battenberg and R. Avizienis. Implementing real-time partitioned convolution algorithms on conventional operating systems. In *Proceedings of the 14th International Conference on Digital Audio Effects. Paris, France*, 2011.

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

[6] J. Blauert. *Spatial hearing: the psychophysics of human sound localization*. MIT press, 1997.

[7] F. Brinkmann, A. Lindau, M. Vrhovnik, and S. Weinzierl. Assessing the authenticity of individual dynamic binaural synthesis. 2014.

[8] A. W. Bronkhorst. Localization of real and virtual sound sources. *The Journal of the Acoustical Society of America*, 98(5):2542–2553, 1995.

[9] D. S. Brungart, B. D. Simpson, and A. J. Kordik. The detectability of headtracker latency in virtual audio displays. In *Int. Conf. on Auditory Display*. Georgia Institute of Technology, 2005.

[10] D. S. Brungart, B. D. Simpson, R. L. Mckinley, A. J. Kordik, R. C. Dallman, and D. A. Ovenshire. The Interaction Between Head-Tracker Latency, Source Duration, and Response Time in the Localization of Virtual Sound Sources. In *Int. Conf. on Auditory Display*. Georgia Institute of Technology, 2004.

[11] M. F. Cohen and D. P. Greenberg. The hemi-cube: A radiosity solution for complex environments. In *ACM SIGGRAPH Computer Graphics*, volume 19, pages 31–40. ACM, 1985.

[12] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 137–145. ACM, 1984.

[13] J. Daniel. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new ambisonic format. In *Audio*

Engineering Society Conference: 23rd International Conference: Signal Processing in Audio Recording and Reproduction, May 2003.

[14] M. J. Evans, J. A. Angus, and A. I. Tew. Analyzing head-related transfer function measurements using surface spherical harmonics. *The Journal of the Acoustical Society of America*, 104(4):2400–2411, 1998.

[15] M. A. Gerzon. Periphony: With-height sound reproduction. *J. Audio Eng. Soc*, 21(1):2–10, 1973.

[16] R. Green. Spherical harmonic lighting: The gritty details. In *Archives of the Game Developers Conference*, volume 56, 2003.

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

[18] C. M. Hendrix. *Exploratory studies on the sense of presence in virtual environments as a function of visual and auditory display parameters*. PhD thesis, University of Washington, 1994.

[19] A. Kulkarni, S. Isabelle, and H. Colburn. On the minimum-phase approximation of head-related transfer functions. In *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, pages 84–87. IEEE, 1995.

[20] E. H. Langendijk and A. W. Bronkhorst. Fidelity of three-dimensional-sound reproduction using a virtual auditory display. *The Journal of the Acoustical Society of America*, 107(1):528–537, 2000.

[21] 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, 2014.

[22] D. Menzies and M. Al-Akaidi. Nearfield binaural synthesis and ambisonics. *The Journal of the Acoustical Society of America*, 121(3):1559–1563, 2007.

[23] H. Møller. Fundamentals of binaural technology. *Applied acoustics*, 36(3):171–218, 1992.

[24] C. Müller-Tomfelde. Time varying filter in non-uniform block convolution. In *Proc. of the COST G-6 Conference on Digital Audio Effects*, 2001.

[25] M. Noisternig, F. Zotter, and B. F. Katz. Reconstructing sound source directivity in virtual acoustic environments. *Principles and Applications of Spatial Hearing, World Scientific Publishing*, pages 357–373, 2011.

[26] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, 45(6):456–466, 1997.

[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] D. Schröder. *Physically based real-time auralization of interactive virtual environments*, volume 11. Logos Verlag Berlin GmbH, 2011.

[29] P. Shirley and C. Wang. Direct lighting calculation by monte carlo integration. In *Photorealistic Rendering in Computer Graphics*, pages 52–59. Springer, 1994.

[30] P.-P. Sloan. Efficient spherical harmonic evaluation. *Journal of Computer Graphics Techniques*, 2(2):84–90, 2013.

[31] 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.

[32] R. L. Storms. Auditory-visual cross-modal perception phenomena. Technical report, DTIC Document, 1998.

[33] O. Warusfel and N. Misdariis. Sound source radiation syntheses: From performance to domestic rendering. In *Audio Engineering Society Convention 116*. Audio Engineering Society, 2004.

[34] E. M. Wenzel. The impact of system latency on dynamic performance in virtual acoustic environments. *Target*, 135:180, 1998.

[35] E. M. Wenzel. Effect of increasing system latency on localization of virtual sounds with short and long duration. 2001.

[36] E. M. Wenzel, J. D. Miller, and J. S. Abel. A software-based system for interactive spatial sound synthesis. In *ICAD, 6th Intl. Conf. on Aud. Disp*, pages 151–156, 2000.

[37] D. N. Zotkin, R. Duraiswami, N. Gumerov, et al. Regularized hrtf fitting using spherical harmonics. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, pages 257–260. IEEE, 2009.

9 Appendix

9.1 Analytical Projection

In this section, we derive the SH coefficients of the projection function of a spherical sound source at the listener's position analytically. We start with the following scenario: let the listener position be $(0,0,0)$, the spherical source position be $(0,0,d)$, the source radius be R and the distance from the listener to the center of the spherical source be d (see Figure 10).

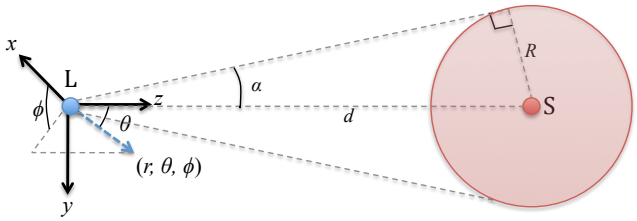


Fig. 10: Spherical projection scenario.

Due to rotational symmetry, the projection of sphere over another sphere is a circular area. The angular size of this circular projection area can be determine by basic trigonometry as $2\alpha = 2sin^{-1}(R/d)$ where α is the half-angle. Mathematically, the projection function $f(\theta,\phi)$ has the following form:

$$f(\theta,\phi) = f(\theta)\begin{cases} \neq 0 & : 0 \leq \theta < \alpha \\ 0 & : \text{otherwise} \end{cases}$$

In other words, the projection function is non-zero inside the projection area and zero outside. The expression to evaluate the spherical harmonic coefficients of projection function becomes:

$$c_{lm} = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\alpha} f(\theta)\, Y_{lm}(\theta,\phi)\, \sin\theta\, d\theta d\phi. \tag{17}$$

Since the definition of spherical harmonics changes with the order m, we have three cases:

Case: $m > 0$
Using the definition of spherical harmonics, we get

$$\begin{aligned} c_{lm} &= \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\alpha} f(\theta)\, \Gamma_{l|m|}\, P_l^{|m|}(\cos\theta)\, \cos(|m|\phi)\, \sin\theta\, d\theta d\phi \\ &= \Gamma_{l|m|}\int_{\theta=0}^{\alpha} f(\theta)\, P_l^{|m|}(\cos\theta)\, \sin\theta\, d\theta \int_{\phi=0}^{2\pi}\cos(|m|\phi)\, d\phi \end{aligned}$$

The right side expression $\int_{\phi=0}^{2\pi}\cos(|m|\phi)\ d\phi = \left[\frac{\sin(|m|\phi)}{m}\right]_0^{2\pi} = 0.$ Therefore, $c_{lm} = 0$ for $m > 0$.

Case: $m < 0$
Using similar derivation as above, it can be shown that $c_{lm} = 0$ for $m < 0$.

Case: $m = 0$
The SH coefficients for the case $m = 0$ are referred to as the zonal harmonics coefficients z_l.

$$c_{l0} = z_l = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\alpha} f(\theta)\, \Gamma_{l0}\, P_l^0(\cos\theta)\, \sin\theta\, d\theta d\phi$$

Substituting $z = \cos\theta$ gives us:

$$z_l = \int_{\phi=0}^{2\pi}\int_{z=\cos\alpha}^{1} f(\cos^{-1} z)\, \Gamma_{l0}\, P_l^0(z)\, dz\, d\phi$$

Separating variables:

$$z_l = \Gamma_{l0} \int_{z=\cos\alpha}^{1} f(\cos^{-1} z)\, P_l^0(z)\, dz \int_{\phi=0}^{2\pi} d\phi$$

Integrate by parts:

$$z_l = 2\pi\Gamma_{l0} \left[f(\cos^{-1} z)\, Q_l^0(z) - \int \frac{d}{dz}(f(\cos^{-1} z))\, Q_l^0(z)\, dz \right]_{\cos\alpha}^{1} \tag{18}$$

where

$$Q_l^0(z) = \int P_l^0(z)\, dz = 2 \sum_{k=0}^{l} \beta_{lk}\, B_{\frac{z+1}{2}}(1, k+1).$$

The notation $B_z(a,b) = \frac{1-(1-z)^b}{b}$ is the incomplete beta function and $\beta_{lk} = \frac{(-l)_k\ (l+1)_k}{k!\ k!}$ is a constant where $(x)_n$ is the Pochhammer symbol.

The right hand side expression in equation 18 above can be analytically integrated if the term $\frac{d}{dz}(f(\cos^{-1} z))$ is a constant. This implies that we can compute SH coefficients of the projection function analytically if the projection function is of the form $f(\cos^{-1} z) = cz + d$ where c and d are constants[1].

In case of spherical sources, a projection function with maxima at $\theta = 0$ and minima for $\theta = \alpha$ would ensure that the projection value is proportional to the depth of the source in that direction. For this purpose, we choose a function such that $\frac{d}{dz}(f(\cos^{-1} z)) = \frac{1}{1+d^2}\ \frac{1}{1-\cos\alpha}$, such that $f(\theta) = \frac{1}{1+d^2}\ \frac{\cos\theta - \cos\alpha}{1-\cos\alpha}$. Using this projection function in equation 18, simplifies it to:

$$z_l = \frac{1}{1+d^2}\ \frac{4\pi}{1-\cos\alpha}\ \sum_{k=0}^{l} \beta_{lk} \left[\frac{1-\cos\alpha}{k+1} - D(1,k) + D(\cos\alpha, k) \right], \tag{19}$$

where

$$D(a,b) = \frac{2^{-b-1}(1-a)^{b+2}}{(b+1)(b+2)} + \frac{a}{b+1}. \tag{20}$$

9.2 Comparison Questionnaire

[1]This logic can be applied recursively to support a function whose m^{th} order derivative is constant i.e. $\frac{d^m}{dz^m}(f(\cos^{-1} z))$ is constant. This would give projection functions of the form $f(\cos^{-1} z) = a_m z^m + a_{m-1} z^{m-1} + ... + a_0$.

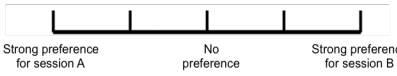
Session-comparison questionnaire

Subject # _____ Trial # _____

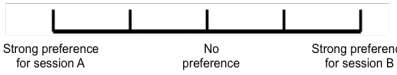
For each of the following questions, select a tab along the line (by drawing a circle) that best represents your final impression of both sessions of the trial.

Please respond to each question similar to the example above. Rate your opinions based on how you felt at the conclusion of each trial.

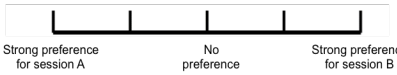
In which session did the spatial extent of the sound better match the visuals?



In which session did you feel most enveloped by the soundscape?



Which session did you prefer?



Comments:

Fig. 11: Session-comparison questionnaire used in our user evaluation.