Tracing Analytic Ray Curves for Light and Sound Propagation in Non-linear Media

Qi Mo*, Hengchin Yeh*, and Dinesh Manocha*

*Department of Computer Science, University of North Carolina, Chapel Hill

Abstract—The physical world consists of spatially varying media, such as the atmosphere and the ocean, in which light and sound propagates along non-linear trajectories. This presents a challenge to existing ray-tracing based methods, which are widely adopted to simulate propagation due to their efficiency and flexibility, but assume linear rays. We present a novel algorithm that traces analytic ray curves computed from local media gradients, and utilizes the closed-form solutions of both the intersections of the ray curves with planar surfaces, and the travel distance. By constructing an adaptive unstructured mesh, our algorithm is able to model general media profiles that vary in three dimensions with complex boundaries consisting of terrains and other scene objects such as buildings. Our analytic ray curve tracer with the adaptive mesh improves the efficiency considerably over prior methods. We highlight the algorithm's application on simulation of visual and sound propagation in outdoor scenes.

1 Introduction

Non-linear media is ubiquitous in the physical world. The atmosphere, even under stable conditions, has spatially varying temperature, pressure, and humidity [1]. There can be wind field or other weather patterns that affect the atmosphere [2], [3], [4]. Similarly, the ocean displays spatial variations in its key properties such as temperature, pressure, and salinity [5]. The propagation speed of sound or light wave at a particular location is determined by the spatially varying properties of the media. Refraction refers to the change of propagation direction of a sound or light wave because of a speed gradient; propagation no longer follows linear paths under refraction. Such refractive media is therefore also known as *non-linear media*, and simulating propagation of light and sound in non-linear media remains a challenging problem.

Non-linear media lead to significant acoustic effects [6]. Take the diurnal change of sound propagation as an example: during the day, when the temperature is typically higher closer to the ground, sound waves are refracted upward, creating a shadow zone with very low level received sound (Figure 14a); when the temperature gradient is inverted at night, sound waves are refracted downward, intensifying the acoustic signals received by the listener. Downward refraction combined with a reflective ground creates a set of concentric circular patterns in the sound field around a source. Outdoor acoustic applications such as noise reduction, urban planning, and virtual reality for training purposes require the propagation simulation to account for those phenomena [7], [8].

Light propagation in outdoor scenes also follows non-linear paths, which can become apparent under certain conditions (e.g. the extreme temperature gradients that produce mirages [9], [10], [11].) For applications with high accuracy requirements, such as satellite laser range-finding [12], [13], [14] and solar radiation modeling [15], simulating the non-linear propagation paths can be critical.

Ray tracing is a powerful tool for simulating propagation. Traditionally, most ray tracing algorithms compute linear propagation paths that change directions only at boundary surfaces [16]. Many previous works (See Section 2.1) have adapted the linear ray tracer for non-linear propagation by taking piece-wise linear ray steps, effectively assuming a constant media within each step. The size of the ray steps therefore is constrained by the magnitude of variations within the media, which hinders the performance of propagating in greatly varying media or over large space. Cao et al. [17] applied analytic ray formulation from geometric optics on visual rendering, which shows promising performance advantage over ray stepping. However, their work does not target large scale general media like the atmosphere, neither has it fully explored the challenges of efficient propagation of both light and sound in complex outdoor scenes.

Some of the prior models and simulator for acoustic propagation [18], [5], [6] rely on the assumption of a stratified media, or a media profile that only varies in height and range, reducing the dimension of the problem and making the computation more practical. Given such assumptions, the propagation can be confined to a 2D plane to reduce the computational overhead, if the media boundary can also be kept simple, i.e. no complex 3D objects to reflect the sound waves off the propagation plane. However, the media profiles vary in a general manner in reality, and are often altered significantly by complex-shaped 3D objects like buildings or terrains.

Main Results: In this paper, we present a fast algorithm that traces analytic ray curves for propagation in non-linear media. Our algorithm improves upon existing methods as follows:

- We trace analytic ray curves as path primitives, which leads to propagation in larger and fewer segments of curves than linear rays. This is essentially an extension of the idea in [17], but we use different ray curve formulations (details in Sec. 3.1) that have simpler forms.
- We utilize the ray curve formulations to perform closed-form intersections with complex 3D objects, enabling fast propagation in large outdoor scenes with many obstacles.
- We construct adaptive unstructured tetrahedral mesh based on the underlying media profiles, and we make the media mesh conform to boundaries of scene objects, both of which contributes to efficient ray curve traversal.

Our approach can perform interactive propagation in fully general media and complex outdoor scenes on a single CPU core (See Sec. 6.6 and 6.7). We validate the accuracy and convergence of our ray tracer on media profiles with analytic solutions (defined in Sec. 3.1), and we demonstrate the performance and applications on complex outdoor benchmarks with realistic profiles (defined in Sec. 3.2). We show a better performance-accuracy trade-off as compared to ray stepping, higher order and adaptive numerical ray integration, and other analytic ray curve methods.

2 Prior Work

Ray tracing literature is vast because of its wide applications, including photorealistic rendering, geometric acoustics, and scientific visualization. We divide the discussion along two challenges for propagation in non-linear media: (1) computing curved paths (Sec. 2.1), and (2) modeling varying media (Sec. 2.2). We then review work that used analytic curves (Sec. 2.3).

2.1 Piecewise linear propagation paths

Early works in computer graphics [9], [10] rendered atmospheric phenomena by modeling the atmosphere with discrete layers. More general media is handled by effectively tracing linear ray segments at each step of a numerical integration of the ray equation, derived from either Eikonal equation [19], [20] or Fermat's principle [21], [22]. Similar methods [23], [24] have been proposed for modeling gravitational fields and dynamic systems. Piecewise linear approximation of curved paths are also at the heart of techniques such as non-linear photon mapping [25], explicit wavefront tracking [26], [11], and voxel-based ray marching [27]. Acceleration has been achieved by parallelism [28], [11], [26], and spatial and temporal caching [29].

However, the step size of linear ray tracing is inherently limited by the magnitude of media variations, hindering its scalability with the size and complexity of the media and the scenes. Higher order numerical methods are adopted to improve the efficiency [21], [22], [23], [24], [25], but the step size is still limited by the underlying media profiles. Furthermore, each advancement of the ray step with higher order solvers can no longer be assumed to be a straight line, making intersection tests with the scenes more complex.

In atmospheric and underwater acoustics, seismic modeling, and related fields, techniques that trace piece-wise linear rays have also been proposed (survey in [6], [5]) and adopted in practical tools [18]. The ray step size limitation can again be a bottleneck; simplifying assumptions on the media (e.g. profiles only vary with height and range), or the scene objects (e.g. conical hills) are often made ([18] e.g. by BELLHOP) to keep computation costs feasible.

2.2 Data structures for non-linear media

Traditional ray tracing acceleration focuses on building tight-fitting hierarchical structures to enclose the surfaces in the scenes (see surveys [16], [30]), assuming a homogeneous medium. A noted exception is the use of constrained Delaunay tetrahedralization (CDT) by Lagae and Dutré [31], which adapts to the density of surfaces in the scene without being hierarchical. On the other hand, rendering participating media faces the same challenge as that of simulating non-linear media: both must characterize volumetric media in addition to surfaces ([32], [33]). Adaptive structures such as kd-trees [34], [35], adaptive grids [36], and manually-graded tetrahedral mesh [37] have been used to facilitate ray marching and/or sampling of scattering events through participating media.

In volume rendering for scientific visualization, polyhedral meshes are commonly used with either ray casting [38], [39], [40], [41] or particle tracing [42]. Polyhedral meshes provide smooth interpolation of the volumetric field [43] with its continuous structure, in contrast to structures like octrees that can have neighboring cells with different resolutions. Unstructured polyhedral mesh also lends itself to adaptive cell sizes, which can either be constructed using a global scheme [44] that varies cell sizes in the entire mesh, or can be built dynamically using a top-down or bottom-up approach, resulting in a multi-resolution representation [45], [46]. Our algorithm uses a global approach similar to [44] to construct the tetrahedral mesh as a pre-process before ray traversal, while the latter methods can be useful for modeling dynamic media. A key difference between our approach and the methods in the context of visualization is that, volumetric ray casting generally does not account for the non-linear refractive paths that the light follows.

In the separate context of meteorology and Earth circulation modeling, unstructured mesh is advantageous due to its adaptive nature and its flexibil-

ity in terms of handling irregular domains. Consequently unstructured meshes have been increasingly adopted to replace regular grids in recent operational models [47], [48]. Models that compute atmospheric flow fields at high resolution[47], [48] can provide detailed media profiles to serve as initial conditions for propagation. Therefore, using unstructured mesh in propagation algorithms makes it possible to couple seamlessly with an atmospheric flow model as input.

2.3 Analytic curve trajectories

Analytic light paths have been derived in the context of geometric optics for simple profiles of refractive index [49], [50], [51]. Cao et al. [17] is perhaps the first work in visual rendering to use the analytic ray formulation derived in Qiao [52]. Cao et al. [17] demonstrated the performance advantage over piecewise linear ray tracing, and used octrees for further acceleration. However, their ray formulation does not have a closed-form solution for intersections with planar surfaces; instead they used bisection methods.

Analytic rays with a polynomial formulation is proposed in [53] for artist-controlled lighting with curved rays. The light paths are not physically-based and cannot be easily extended to more realistic kinds of light bending from refraction. Grave et al. [54] visualize the effects of general relativity using an analytic solution derived for the Gödel universe.

In computational acoustics, closed-form ray trajectories have been derived for constant gradient in the propagation speed, and in the squared refractive index. The term *cell method* refers to acoustic ray tracing that subdivides media into cells and assumes analytic ray paths in each cell, but it has only been used for 2D varying media modeled by regular triangular grid with no obstacles [55], [56]. Our algorithm can be seen as an extension of cell methods to a more general propagation algorithm that can handle 3D varying media and complex scene objects. Furthermore, we improve the efficiency based on closed-form ray intersections and use of an adaptive unstructured mesh.

3 Background

A non-linear media profile can be described by spatially-varying propagation speed $c(\mathbf{x})$, or equivalently by index of refraction $n(\mathbf{x}) = c_0/c(\mathbf{x})$, for each location $\mathbf{x}$, where c_0 is the reference propagation speed. In this section, we first introduce a few simple profiles with analytic ray solutions, two of which we have adopted as the foundation of our ray curve tracer. We then introduce models of general non-linear media that corresponds to physical reality.

3.1 Profiles with analytic ray solutions

In ray tracing for wave propagation, rays are defined as normal to the wavefront. The Eikonal equation for ray trajectories is derived from the wave equation as:

$$\frac{d}{ds}\left(\frac{1}{c(\mathbf{x})}\frac{d\mathbf{x}}{ds}\right) = -\frac{1}{c(\mathbf{x})^2}\nabla c(\mathbf{x}), \tag{1a}$$

$$\frac{d}{ds}\left(n(\mathbf{x})\frac{d\mathbf{x}}{ds}\right) = \nabla n(\mathbf{x}), \tag{1b}$$

$\mathbf{x} = \{x, y, z\}$ is the Cartesian coordinates and s is the arc-length along the ray.

The analytic ray trajectories are known for a set of profiles with constant media gradient, and we give the trajectories in a local coordinate system aligned with the gradient direction. If we place the origin of the coordinate system at the ray origin $\mathbf{x}$, and take the media gradient direction as the z-axis, the ray trajectory is a plane curve that lies in the plane formed by the z-axis and the initial ray direction $\mathbf{d}$, i.e. the *ray plane*. We then take the direction perpendicular to the z-axis as the r-axis within the ray plane. Figure 1 plots the analytic ray curves for the following profiles (see Appendix A for detailed derivations):

- **c-linear:** $c(z) = c_0 + \alpha z$, $\alpha = \|\nabla c\|$, c_0 is c at ray origin. Let $\xi_0' = \frac{cos\theta_0}{c_0}$, where θ_0 is the angle between $\mathbf{d}$ and r axis, the ray trajectory in r-z coordinates is derived from Equation (1a) to be:
$$r(z) = \frac{\sqrt{1-\xi_0'^2 c_0^2} - \sqrt{1-\xi_0'^2\left(c_0+\alpha z\right)^2}}{\xi_0'\alpha}, \tag{2}$$
which is a circular curve in the ray plane.
- **n^2-linear:** $n^2(z) = n_0^2 + \alpha z$, $\alpha = \|\nabla n^2\|$, n_0 is n at ray origin. We establish a similar coordinate system with origin at $\mathbf{x}$, and take the direction of ∇n^2 as z-axis. Let $\xi_0' = n_0 cos\theta_0$, where θ_0 is the angle between $\mathbf{d}$ and r axis, the ray trajectory is:
$$r(z) = \frac{2\xi_0'}{\alpha}\left(\sqrt{-\xi_0'^2 + n_0^2 + \alpha z} - \sqrt{-\xi_0'^2 + n_0^2}\right), \tag{3}$$
which is a parabolic curve in the ray plane.
- **n-linear:** The analytic ray curve for constant ∇n was used in [17], although unlike the previous two ray curves, it does not have an analytic solution for intersection tests with planar surfaces.

There are also analytic solutions for the profiles that produce superior and inferior mirages [57]. The two profiles are also described in [17], and we use their analytic solutions to validate our ray tracer:

- **Inferior mirage (V-IM)**, with the squared refractive index: $n^2(z) = \mu_0^2 + \mu_1^2(1 - exp(-\beta z))$,
- **Superior mirage (V-SM)**, with the squared refractive index: $n^2(z) = \mu_0^2 + \mu_1^2 exp(-\beta z)$, with constants $\mu_0 = 1.000233, \mu_1 = 0.4584, \beta = 2.303$.

3.2 Realistic profiles

The atmosphere and the ocean are two of the most prominent non-linear media in outdoor scenes. We hereby focus our discussion on atmospheric properties, but we would like to point out that media properties and propagation in the ocean are analogous.

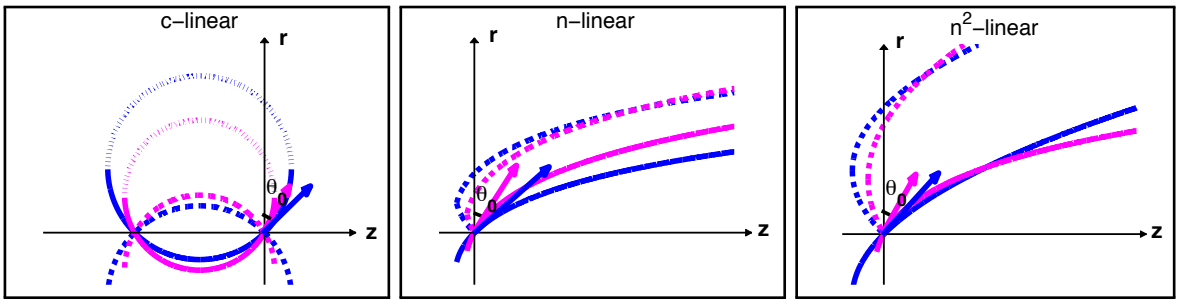


Fig. 1: **Analytic ray curves** in c-linear (circular curves), n-linear, and n^2-linear (parabolic curves) media profiles. Our algorithm uses the circular and the parabolic curves as ray tracing primitives, while the n-linear curve is used in Cao et al.[17]. The r-z plane is the *ray plane* defined in Section 3.1. Red and blue curves represent different θ_0 (30°and 45°, respectively). The dashed curves trace out the ray paths if the launch angles are flipped around the r-axis. The dotted part of c-linear curve shows the circular shape only and does not represent actual trajectories (the ray will proceed in straight line after the direction becomes parallel to the r-axis).

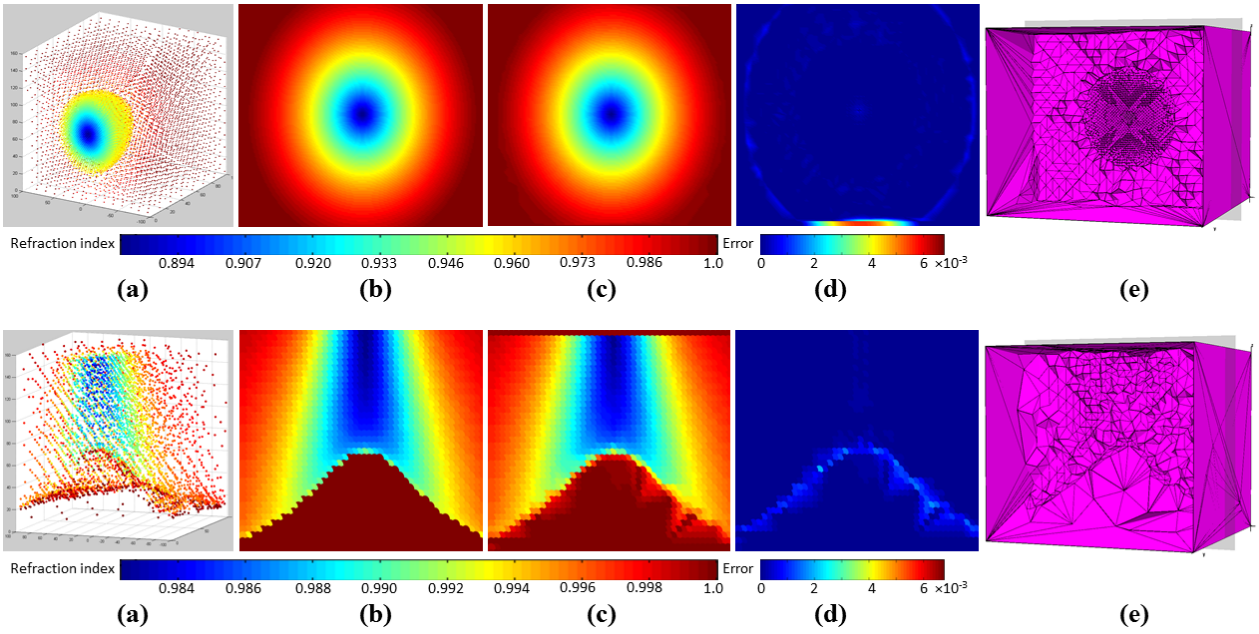


Fig. 2: **Adaptive meshes.** Here we show two meshes generated for the hot spot (**A-HS**) in the top row and upwind-over-hill (**A-UW**) profiles in the bottom row, respectively (both defined in Sec. 3.2). **(a)** Resampled media points (showing half of the points to expose the sectional view), **(b)** Input media profiles, **(c)** Interpolated media profiles from the meshes, **(d)** Absolute approximation errors, **(e)** Adaptive meshes. The input media grid has 6.4×10^6 ($200 \times 200 \times 160$) points for **A-HS**, and 8×10^5 ($100 \times 100 \times 80$) points for **A-UW**. The meshes are constructed from a resampled 4.3×10^4 points for **A-HS**, and 9.8×10^3 points for **A-UW**. With $100\times$ fewer sample points than the input the adaptive meshes are able to achieve low approximation errors.

For both light and sound propagation speed, temperature is a key determining factor. The atmospheric sound speed given temperature, for example, is

$$c = \sqrt{\gamma R_d T_v}, \tag{4}$$

where $\gamma = c_p/c_v$ is the ratio of the specific heats, R_d is the gas constant of dry air, T_v is the virtual temperature considering humidity, and can typically be approximated by the absolute temperature T.

A standard profile of atmospheric temperature and pressure is available with the 1976 USA Standard Atmosphere [1]. It can be de-standardized with the following model for localized heat sources:

- **Hot spot (A-HS)** is computed by Eq. 4 with combined temperature from [1] and Eq. 5,

$$T = T_0 + (T_s - T_0)exp(-d/d_0), \tag{5}$$

where $T_0 = 273K$, T_s is the temperature at the hot spot, d is the distance to the hot spot, and d_0 is the dropoff length.

Alternatively, we can adopt a widely-used empirical models of the atmosphere [6] that gives the sound speed directly. The sound speed is modeled with a stratified component c_{str} and a fluctuation component c_{flu}, so that $c = c_{str} + c_{flu}$. The stratified component follows a logarithmic profile of the altitude z:

$$c_{str}(z) = c_0 + b \ln\left(\frac{z}{z_g} + 1\right)), \tag{6}$$

where c_0 is the sound speed at the ground, and z_g is

the roughness length of the ground surface. Different values of the parameter b lead to different profiles:

- **Stratified profile, upward (A-LU) or downward (A-LD) refractive,** computed by Eq. 6 with n_0 = 1, c_0 = 340 m/s, and z_g = 1 m. We take $b = 1m/s$ for A-LD and $b = -1m/s$ for A-LU.

The fluctuation component models the random atmospheric temperature and wind speed turbulence:

$$c_{flu}(\mathbf{x}) = \sum_i G(\mathbf{k}_i)\cos(\mathbf{k}_i \cdot \mathbf{x} + \varphi_i), \quad (7)$$

where $\mathbf{k}_i$ is the wave vector describing the *spatial* frequency of the fluctuation, φ_i is a random angle $\in [0, 2\pi]$, $G(\mathbf{k_i})$ is a normalization factor, and we have:

- **Stratified-plus-fluctuation (A-LU+F, A-LD+F)** A-LU or A-LD combined with Equation 7.

For sound propagation, the wind profile plays a role that is as important as the temperature[58], [59], and the wind profile is significantly modified above undulating terrains. For example, Jackson and Hunt [60] derived closed-form wind profile for a hill of the shape: $f(\frac{x}{L}) = \frac{1}{1+(\frac{x}{L}^2)}$, where x is the horizontal distance from the apex of the hill, L is the radius of the base of the hill. According to the Monin-Obukhov similarity theory [2], the mean wind velocity follows the logarithmic law with height z: $u(z) = \frac{u_*}{K}\ln\frac{z}{z_g}$, where K is the von-Karmann constant, z_g is the aerodynamic roughness length, and u_* is the friction velocity [3], [61]. The horizontal component of the wind velocity over this particular hill shape, in addition to the mean velocity $u(z)$, is given as:

$$\Delta u = u_0(z=L)\frac{h}{L}\frac{\ln(\frac{L}{z_0})}{\ln^2(\frac{l}{z_0})}\Big(\frac{1-(\frac{x}{L})^2}{1+(\frac{x}{L})^2}\ln(\frac{\Delta z}{z_0}) - \Big(\frac{2(x/L)}{(1+(x/L)^2)^2}\Big(\frac{\Delta z - z_0}{l}\Big)\ln(\frac{\Delta z}{z_0})\Big), \quad (8)$$

where δz is the distance above the hill, l is the thickness of the hill's influence region, in which the flow above the ground is perturbed, and we have:

- **Wind over hill (A-UW for upwind, A-DW for downwind)** $u(z) + \Delta u$ is combined with temperature-induced sound speed profile based on the 1976 USA Standard Atmosphere [1].

4 ALGORITHM

In this section, we present our ray tracing algorithm with analytic ray curves and adaptive media mesh.

4.1 Mesh-based Ray Curve Tracer

The idea of our algorithm is to spatially decompose the volume of a media profile with an unstructured tetrahedral mesh, so that within each cell of the mesh the media profile can be assumed to have constant gradient. Based on the formulation of analytic ray curves presented in Sec. 3.1, the propagation paths within each cell of such a mesh follow one of the analytic forms, depending on the particular media gradient used to build the mesh. We thereby compute the propagation paths consisting of segments of analytic ray curves, and those segments traverse the media cell-by-cell based on mesh connectivity.

The pseudo-code for the traversal of ray curves through a tetrahedral mesh is given in Algorithm 1. Given a ray origin, we first locate the tetrahedral cell that contains the origin. This is commonly referred to as *point location*, and it can be relatively expensive for complex models when there are a large number of cells. However, in most scenarios, each primary ray originates from the same point (light or sound source), and each secondary ray (after interacting with boundary surfaces) originates from the same cell where the previous ray that spawned it ends. The point-location query is performed once per frame, and the cost is amortized over all the rays.

```
1  Point Location: locate ray origin P in cell T;
2  Compute analytic ray from media property of T;
3  Intersect ray curve with T to find exit face F;
4  if ray encounters boundary surface in T then
5  |   Surface interaction (e.g. reflection);
6  |   Go to Step 2 with T unchanged;
7  else if there is a tetrahedron T' incident to F then
8  |   Go to Step 2 with T = T';
9  else
10 |   ray exits the scene;
11 end
```

Algorithm 1: Ray Curve Traversal.

Once the initial cell is located, we retrieve the precomputed per-cell media gradient ∇m (see details in Sec. 5.3). The direction of ∇m and the origin and initial direction of a ray are used to define the *ray plane*, and we can compute the curved trajectory within the cell by Eq. 2 or 3 (Line 2, Algorithm 1).

The ray curves we use have closed-form intersection solutions with planar surfaces, e.g. the four faces of the tetrahedral cell. The intersection point closest to the ray origin is chosen as the exit point from the cell, and the neighboring cell incident to the exit face is taken as the next cell in the traversal (Line 7 in Algorithm 1). We use its media properties to compute the next segment of the curved ray path .

4.2 Adaptive Mesh

To further accelerate ray traversal, our goal is to construct a tetrahedral mesh with graded cell sizes that adapts to the spatial distribution of media properties, hereafter referred to as an *adaptive mesh*. The cost of computing a ray curve and its intersection within each media cell is constant, therefore an adaptive mesh leads to fewer cells and thus faster ray traversal. We want to vary the cell sizes according to media variations, based on the heuristics that the range of

validity for the assumption of a constant media gradient is inversely proportional to the magnitude of the gradient. Given an input profile of propagation speed $c(\mathbf{x})$ for each grid point $\mathbf{x}$, we compute the *slowness* $k(\mathbf{x}) = \frac{1}{c(\mathbf{x})}$, and the gradient of the slowness $\nabla k(\mathbf{x})$ by finite difference. We then compute the spatially-varying target cell sizes $d(\mathbf{x})$ such that $\sigma = \frac{1}{4}\nabla k d^2(\mathbf{x})$, with a global σ that controls the overall variation allowed in each cell. Details of mesh construction that realize the target cell sizes are discussed in Sec. 5.1.

4.3 Embed Media Boundaries

During propagation, objects in the scenes (such as terrains, buildings, and sound barriers) affect the propagation paths of ray curves. Unlike axis-aligned structures commonly used in ray tracing, (e.g. octrees), tetrahedral mesh has flexible structures that can embed surfaces of arbitrary orientations. We choose to leverage this capability to embed surfaces in the mesh.

With surface-embedded mesh, no separate intersections with surfaces are computed during the ray traversal, and Line 4 in Algorithm 1 is merged with Line 3. When a ray exits from a face of the cell that corresponds to an embedded face, the current ray traversal terminates and a secondary ray is spawned reflecting off the embedded face.

While we observe that embedding the surfaces often brings speedup, because it unifies mesh traversal and surface intersections and eliminates extra computation, inserting constrained surfaces adds considerable computational overhead to mesh construction. We evaluate this trade-off between construction efficiency and traversal efficiency individually for each input scene, and we discuss the details in Sec. 5.2.

4.4 Surface Interactions

Given the ray trajectories in Equation 2 and 3, the tangent direction at arbitrary point (e.g. an intersection point) along the ray curve can be evaluated analytically for the circular curve:

$$\frac{dr}{dz} = \frac{(\xi_0'(\alpha z + c_0))}{\sqrt{1 - \xi_0'^2(\alpha x + c_0)^2}}, \tag{9}$$

and for the parabolic curve:

$$\frac{dr}{dz} = \frac{\xi_0'}{\sqrt{-\xi_0'^2 + \alpha x + n_0^2}}. \tag{10}$$

This gives the incident ray location and direction when intersecting a surface, then different surface interactions including reflection, Snell's law refraction, or BRDF-based sampling can be employed to generate the direction of the next ray (Line 4, Algorithm 1). Furthermore, the circular and parabolic ray curves both have closed-form arc lengths, which can be used to compute attenuation of propagated energy for light (e.g. [62]) and for sound ([5]). A closed-form arc length is also convenient for free path sampling to simulate media scattering [34], [36]. While we focus on the refractive characteristics of non-linear media and specular boundary reflections and do not perform BRDF sampling or media scattering in our benchmark results, our ray formulation is compatible with more complex surface interactions and media participation.

5 Implementation

There are multiple ways to construct an adaptive mesh, to incorporate boundary surfaces, and to compute per-cell media gradients. In this section, we give details of our implementation and also discuss some alternatives. Further discussions with experimental results are available in Appendix B and C.

5.1 Construction of Adaptive Mesh

We assume that the input media profiles are available on a three-dimensional uniform grid. The data points on the grid are generated from real-world measurements or from sampling a characteristic profile. We will now describe our method of resampling an input profile to generate a point set distributed according to local magnitude of media variations; tetrahedralization on such a point set generates an efficient structure for both media representation and ray traversal.

```
1  Given the input media grid points G, initialize a
   flag array that marks each point in G as false;
2  Initialize an empty point set S for output, and a
   queue of points T, enqueue the center point x_i;
3  while T is not empty do
4      dequeue x_i;
5      if x_i lies within the bounds of the profile then
6          compute a spherical region with center x_i
           and radius d(x_i);
7          if all samples in the spherical region are
           marked false then
8              mark all such samples true;
9              add x_i to S; enqueue ideal sites of x_i
               with spacing d(x_i) in T;
10         end
11     end
12 end
```

Algorithm 2: Media sample redistribution for mesh construction. The *ideal sites* are the locations of neighbors in a FCC lattice [44].

As mentioned in Sec. 4.2, we compute a desired cell size $d(\mathbf{x})$ for each point location $\mathbf{x}$ within the profile. Now we want to vary the spacing between sample points according to $d(\mathbf{x})$ when we resample the input media profile. We use Algorithm 2 to obtain the set of resampled points S from the profile G, in a manner similar to the *Atomic Meshing* process [44]. Basically, a *face-centered-cubic* (FCC) lattice is grown from the center of the space outward, placing each new point away from existing samples by the spacing $d(\mathbf{x})$. The approximation errors that are introduced by the resampling process are quantified in Section 6.2.

5.2 Embedding of Boundary Surfaces

To embed surfaces, we insert them as boundary constraints and construct a constrained tetrahedral mesh. Given a point set S with adaptive spacing computed by Algorithm 2, and given the optional constrained surfaces of the objects (P) in the scene, we use the method proposed by Si and Gärtner [63] and implemented in the *TetGen* software package, to build a Constrained Delaunay Tetrahedralization (CDT) with S and P. The resulting CDTs have adaptively graded cell sizes due to the distribution of S, and we observe well-shaped tetrahedral mesh with a maximum radius-edge ratio below 2.0 in our benchmarks.

In our benchmarks, the resolution of surface primitives is compatible with nearby media variation, leading to compact constrained meshes that are efficient to traverse. However, if there are over-tessellated objects, the constrained mesh generation algorithm can end up choosing overly small cell sizes close to the objects' surfaces and this affects the traversal performance.

There are two options to compensate this. We could convert the explicit triangles of the object boundaries to implicit surfaces, then construct a tetrahedral mesh that conforms to the implicit surfaces, effectively retessellating them (as in [64], input scenes already in the forms of parametric or other implicit surfaces can be handled similarly). This method keeps the performance benefit of a unified traversal while achieving proper cell size and compactness of the mesh. Or we could link each cell to a list of boundary surfaces that it overlaps with, similar to [17]; the ray traversal of each cell then needs to iterate through this list to compute intersections. This approach has the benefit of simplicity, but might compromise traversal performance. Furthermore, generating such links comes with its own computational overhead. In Appendix C, we report the experimental performance of both mesh construction and traversal for the two options.

5.3 Gradient estimation

Given the spatial decomposition of the media profile with our tetrahedral mesh, we need accurate estimation of the media gradient within each tetrahedral cell to compute the analytic ray trajectories entering that cell. We adopt a cell-centered linear regression-based method to estimate the gradient.

For media property m (e.g. c or n^2) defined over the domain, and a mesh cell C with centroid $\mathbf{x}_0$, the cell gradient ∇m should satisfy the equation system:

$$\begin{bmatrix} (\mathbf{x}_1-\mathbf{x}_0)^\intercal \\ (\mathbf{x}_2-\mathbf{x}_0)^\intercal \\ (\mathbf{x}_3-\mathbf{x}_0)^\intercal \\ (\mathbf{x}_4-\mathbf{x}_0)^\intercal \end{bmatrix} \nabla m = \begin{bmatrix} m(\mathbf{x}_1)-m(\mathbf{x}_0) \\ m(\mathbf{x}_2)-m(\mathbf{x}_0) \\ m(\mathbf{x}_3)-m(\mathbf{x}_0) \\ m(\mathbf{x}_4)-m(\mathbf{x}_0) \end{bmatrix}, \quad (11)$$

where $\mathbf{x}_k, k = 1, ..., 4$ are the centroids of the 4 neighbors of C, $m(\mathbf{x}_k)$ is the media property values at those centroids. Written in matrix form:

$$\mathbf{X}\nabla m = \mathbf{b}, \quad (12)$$

Optionally, the system can be weighted to take into consideration of the irregular shapes of the mesh:

$$\mathbf{W}\mathbf{X}\nabla m = \mathbf{W}\mathbf{b} \quad (13)$$

where $\mathbf{W} = diag\{w_i\}$ is a 4×4 diagonal matrix containing the weights of neighbor k of cell C. This can be solved with linear least square (See Appendix D for the explicit solution of the estimated gradient).

The average-based gradient estimation method used in prior work [17] is faster to compute, but the regression-based method, especially the weighted version with inverse centroid distance, has been shown to provide better accuracy for irregular shaped mesh elements, and adapts well to lower-quality meshes [65], [66]. Accuracy of the estimated gradient is particularly important for outdoor propagation, when artifacts such as false caustics have been shown to happen with discontinuous gradients [67].

6 Results and Analysis

In this section, we highlight the applications of our algorithm on light and sound propagation on benchmarks with different media profiles and geometric primitives. We analyze the errors of the adaptive mesh approximating the underlying media (Sec. 6.2), and the subsequent accuracy and convergence characteristics of our ray curve tracer (Sec. 6.3). We compare performance with numerical ray integration methods (Sec. 6.4), as well as with the prior work[17] that traces a different ray curve for visual rendering (Sec. 6.5). Our ray curve tracer enables fast propagation in complex 3D scenes (Sec. 6.6), which is demonstrated by outdoor sound simulation at a performance that has not been achieved before (Sec. 6.7).

Except Fig. 3 right generated by the circular curve, the results in this section is generated by the parabolic curve. It is our observation that both curves yield similar accuracy and performance, and we omit the circular curve results to avoid duplications.

6.1 Benchmarks

Propagation in non-linear media is important for both visual and acoustic applications, therefore we tested our algorithm on visual (Figures 4, 11, 6, 10, , 7, and Table 2) and acoustic (Figures 2, 5, 12, 13, 8, 14, 9, 15, and Table 3) benchmarks. Each of the benchmarks consists of a media profile, and triangulated geometric representation of the boundary surfaces.

6.1.1 Media profiles

We tested our algorithm on all the profiles listed in Sec. 3. In particular, we perform validation and convergence tests of the ray curve tracer with the profiles that have analytic ray solutions (c**-linear**, n**-linear**, n^2**-linear**, Inferior mirage **V-IM**, and Superior mirage **V-SM**). Visual results are generated for **V-IM**, **V-SM**, and for the texture-based profile used in [17] (details in Sec. 6.5). Acoustic results are obtained for Hot spot

A-HS, Upward/downward refractive atmosphere **A-LU, A-LD**, atmosphere with fluctuations **A-LU+F, A-LD+F**, and Wind over hill **A-UW, A-DW**.

6.1.2 Geometric models

For validation tests, we exclude geometry in the scene except a flat ground as required by the analytic solutions. For visual comparisons we use the same 3D elephant model used in [17]. For acoustic applications, the Desert, Christmas, and Reservoir represent large-volume outdoor scenes that have complex surface geometry (e.g. varying terrains and buildings).

Scene	# surfaces	# medium points.	# tetrahedra
Elephant	1,500	1,532	5,538
Desert (m)	8,000	23,632	144,976
Desert (h)	16,000	132,742	674,434
Christmas (m)	8,000	44,862	227,851
Christmas (h)	16,000	179,382	1,169,353
Reservoir	4,000	34,771	248,806

TABLE 1: **Benchmarks.** Desert and Christmas are tested at two different resolutions, respectively, marked by (m) for median and (h) for high resolution.

6.2 Error analysis of adaptive mesh

Assume that the media profile in refractive index n is available as input on a regular grid of points $\mathbf{x}_G$, so that $n_G = n(\mathbf{x}_G)$. In our adaptive mesh construction (Algorithm 2), we resamples the media profile using a smaller set of points S, and we construct a tetrahedral mesh from the resampled set. The *approximated* refractive index $\tilde{n}(\mathbf{x})$ at an arbitrary position $\mathbf{x}$ within the domain is then obtained by Barycentric interpolation with Equation 40. The approximation error is defined as the difference $E = n_G - \tilde{n}_G$, where $\tilde{n}_G = \tilde{n}(\mathbf{x}_G)$. The relative error is

$$E_{rel} = \frac{\|n_G - \tilde{n}_G\|}{\|n_G\|}, \tag{14}$$

where $\|\cdot\|$ denotes a 2-norm. The error is a function of the size of S, which is controlled by the global δ.

We perform error analysis with two media profiles (**A-HS** and **A-UW**) (see Fig. 2), which illustrate the capability of our adaptive mesh to capture different profiles with accuracy. We plotted the point set S color-coded by n_S in Fig. 2(a), a slice of input n_G in Fig. 2(b), a slice of the approximated $\tilde{n}_G$ defined by S in Fig. 2(c), and the relative error in Fig. 2(d). The constructed meshes for those profiles are shown in Fig. 2(e), where finer cells can be seen concentrated in regions of greater variations within the media.

Moreover, Fig. 12 in Appendix shows the approximation error with the profile (**A-LU+F**). Given an input grid of 2.09×10^5 points spanning a space of $160\text{m} \times 160\text{m} \times 160\text{m}$ with 1.25m grid spacing, we obtain the resulting S with $23,462$ points by resampling with $\delta = 0.001$. With 100 times fewer points than the input grid, the approximated $\tilde{n}_G$ is able to capture the features of n_G, and the relative error is below 4×10^{-4}. As shown in Fig. 12(e), the relative error decreases with increasing size of S, controlled by δ.

6.3 Error analysis of ray curve tracer

Close approximation of the media profile by the adaptive mesh leads to accuracy in the ray tracing results. We use the ray hit points and the travel distance along the ray paths, both crucial for light and sound propagation, to quantify the ray tracing accuracy.

For validation purposes, we choose the profiles defined in Sec. 3.1 that have known analytic ray solutions, among which c**-linear**, n**-linear**, n^2**-linear**, **V-IM**, and **V-SM** have analytic solutions for trajectories (and therefore ray hit points), and c**-linear** and n^2**-linear** have analytic solutions additionally for ray arc-length. In Fig. 3 and 4 we show the results of our ray curve tracer on those profiles, which converge to the corresponding analytic solutions with increasing number of ray curve segments controlled by δ.

The ray tracing results are computed for a source $1.5m$ above the ground, and rays are traced for elevation angles $\in [-\pi/4, \pi/4]$. For visual profiles, results are compared at a propagation range of $100m$, while for acoustic profiles, results are compared after propagating for $10s$ in time. The media gradient magnitude used are: $\alpha = 0.1$ for c-linear, $\alpha = 0.02$ for n-linear, and $\alpha = 0.01$ for n-linear profiles.

6.4 Comparison to numerical ray integration

We compare the accuracy and convergence of our ray curve tracer with numerical ray integration methods including higher-order solver (4th order Runge-Kutta) and an adaptive scheme (Dormand-Prince).

Fig. 4 and 5 demonstrate the advantages of the ray curve tracer over numerical integration for visual profiles **V-IM**, **V-SM**, and acoustic profiles **A-LU**, **A-LD**. We use the analytic solutions of **V-IM** and **V-SM** as ground truth to compute the relative error in Fig. 4, and for **A-LU** and **A-LD** (Fig. 5) we use a converged solution with 1×10^{-7} tolerance of error computed by our ray curve tracer as the ground truth. In both figures our ray tracer results approach the ground truth with increasing number of segments controlled by δ. The 4th order Runge-Kutta and Dormand-Prince methods, however, have difficulty achieving comparable accuracy even with decreasing step sizes.

Under the **V-IM** and **V-SM** profiles, we also performed same-quality and same-speed comparisons with rendered images. The image difference rendered by the ray curve tracer and the numerical ray integration using Euler method (piece-wise linear rays) is shown in Fig. 11 in Appendix. The performance of curved ray tracer is an order of magnitude faster under the same-quality comparison, while the piece-wise linear rays lead to noticeable artifacts when running at competitive speed (same-speed comparison).

6.5 Comparison to prior analytic ray curve

Cao et al.[17] proposed a ray tracer based on the n-linear analytic ray curve and spatial decomposi-

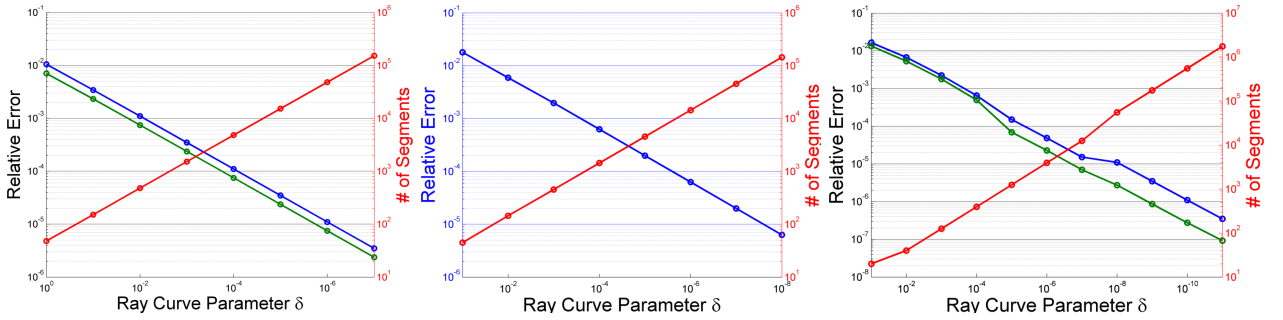


Fig. 3: **Convergence of ray curve tracer.** Given the analytic ray trajectories known for c**-linear**, n**-linear**, and n^2**-linear** profiles, our ray curve tracer converge to the analytic solution (decreasing relative error) with increasing number of segments controlled by δ. Additionally, the relative error of ray arc-length are plotted for c**-linear** and n^2**-linear** profiles (green lines), for which the analytic solutions of arc-length are also known.

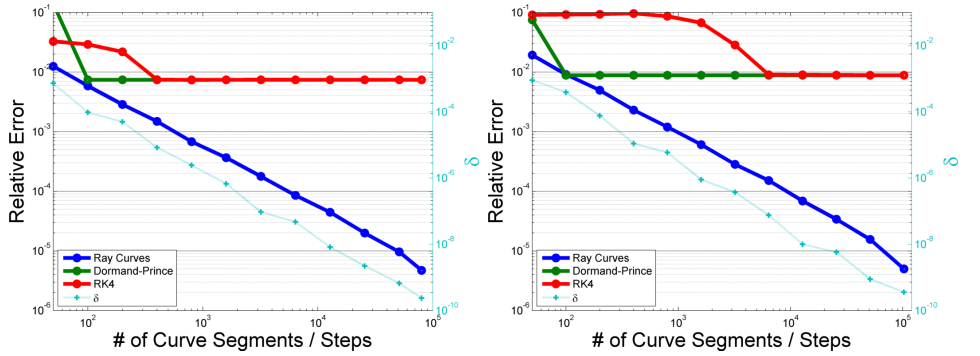


Fig. 4: **Comparison to numerical solvers on mirages.** For inferior mirage **V-IM** (left) and superior mirage **V-SM** (right) profiles, the results of our ray curve tracer (hitpoints plotted here) converge to the analytic solutions with adaptively increasing segments controlled by δ. The 4th order Runge-Kutta and Dormand-Prince methods fail to approach the analytic solutions beyond a relative error of 1×10^{-2}.

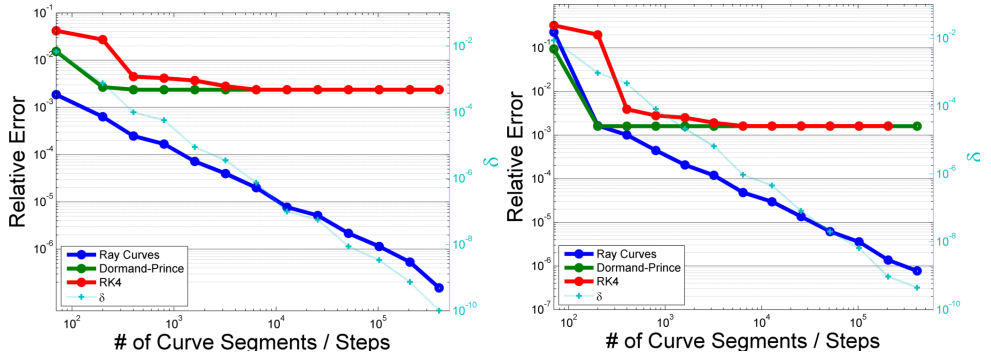
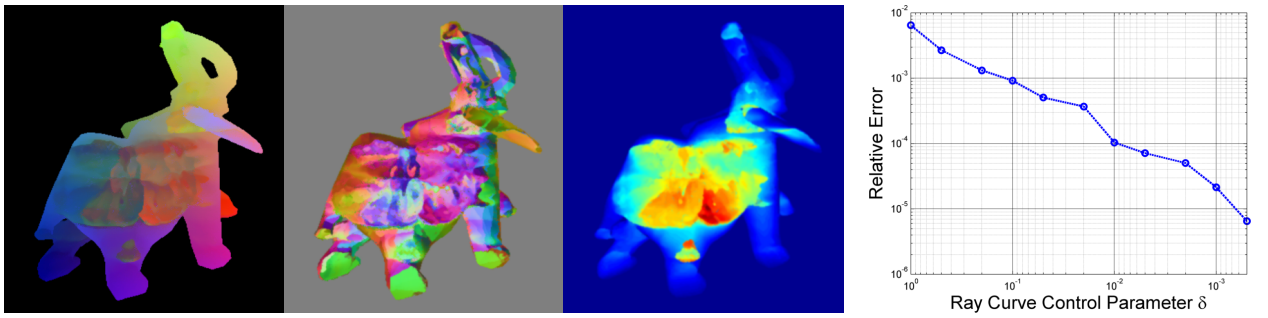


Fig. 5: **Comparison to numerical solvers on logarithmic acoustic profiles.** With downward refractive **A-LD** (left) and upward refractive **A-LU** (right) profiles, our ray curve tracer computes ray tracing hit points that converge with adaptively increasing segments controlled by δ. The 4th order Runge-Kutta and Dormand-Prince methods again stop at a relative error of 1×10^{-3} with arbitrarily large number of integration steps.

tion of the media with octrees, and we compare our algorithm to it on various aspects. By adopting different ray formulations, and by modeling general media with adaptive unstructured mesh, we highlight improvements in performance and accuracy.

We replicate the elephant benchmark used in [17] with the texture-based profile of refractive indices. For our ray tracer, we construct adaptive tetrahedral meshes using the algorithm given in Sec. 5.1, and we show two meshes constructed with different δ and therefore different resolutions in Fig. 10 in Appendix. The adaptive mesh captures the spatially-varying media gradients corresponding to the texture. By tuning δ, the mesh-based ray curve tracer strikes a balance between tracing performance and accuracy. This is demonstrated by Fig. 6, where the convergence of our ray curve tracing results is illustrated (Fig. 6(d)), together with a visualization of the converged ray hit points, exit directions, and path lengths (Fig. 6(a,b,c)). We then compare four ray tracing configurations:

- Parabolic ray curve and adaptive mesh controlled by δ (our algorithm),
- n-linear ray curve and adaptive mesh built assuming per-cell constant gradient of n,



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

Fig. 6: **Convergence of ray curve tracer results** for the elephant benchmark[17]. **(a,b,c)** Converged ray tracing results computed by our ray curve tracer, including visualized **(a)** hitpoints, **(b)** exit directions, and **(c)** path lengths. **(d)** The ray curve tracer converges with decreasing relative error between consecutively decreasing δ.

Method	Rays/path	Frame time
Parabolic ray + adaptive mesh	2.16	123 ms
n-linear ray + adaptive mesh	2.21	232 ms
n-linear ray + octree[17]	3.37	377 ms
Parabolic ray + non-adaptive mesh	8.91	619 ms

TABLE 2: **Same-quality comparisons.** We run our ray curve tracer with $\delta = 0.1$, and tune the alternative methods to achieve the same level of accuracy. The performance is measured by the average number of ray segments per path and by the total frame time.

- n-linear ray curve and octree, according to [17],
- Parabolic ray curve with a non-adaptive mesh.

We perform same-quality and same-speed comparisons among the four methods, in order to pinpoint the sources of performance difference. Ray tracing efficiency and accuracy are compared between parabolic and n-linear ray curves, between adaptive mesh and octree, and between adaptive and non-adaptive meshes, keeping the other configurations constant for each pair-wise comparison.

The same-quality comparison results are reported in Table 2. The latter three methods are tuned by their respective parameters to match the accuracy of our algorithm. In particular, the adaptive mesh built for n-linear ray curve is tuned similar to using δ in our algorithm; Cao et al.'s method is controlled by the parameters suggested in [17]; the non-adaptive mesh's resolution is increased uniformly. At comparable accuracy, our ray tracer takes significantly fewer segments per propagation path and/or less total frame time. The same-speed comparison results can be seen in Fig. 7. It is shown that the results of the other three methods are less accurate than of our ray tracer running with comparable computation time.

It is our observation that the difference between the parabolic and the n-linear ray, both with adaptive mesh, is mainly due to the intersection cost. Our ray formulation's closed-form intersections with planar surface contributes to the performance advantage over the n-linear ray[17], which requires bisection. The runtime breakdown in Table 3 shows that bisection takes up a large portion of the traversal time. We would also like to point out that our ray curves also have closed-form solution for tangent direction and arc length at any point along the ray. These are useful for computing boundary interactions as well as absorption and scattering, as explained in Section 4.3, which are not taken into consideration in [17].

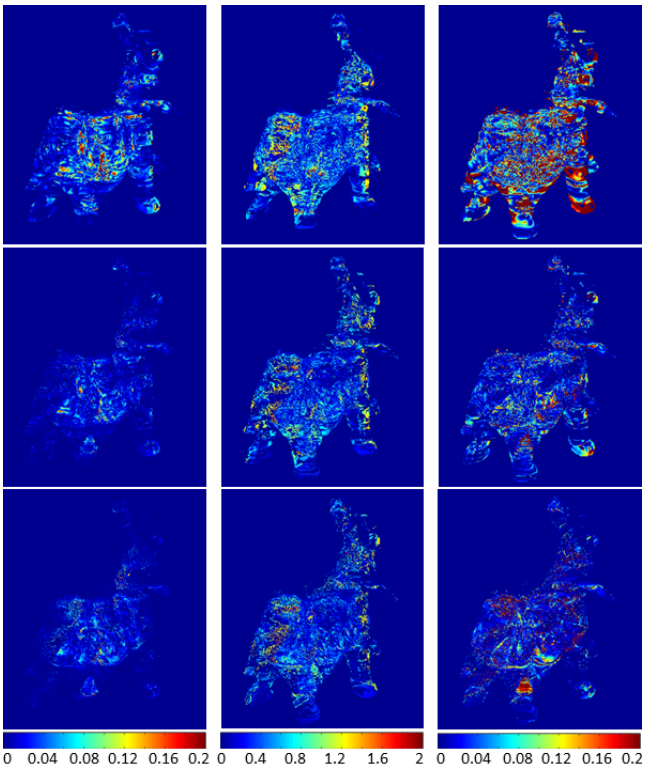


Fig. 7: **Same-speed comparisons.** We run our ray tracer with $\delta = 0.1$, tune all methods to run at the same speed, and compare the accuracy. **Columns (left to right):** Differences in hitpoints, exit directions, path lengths, between **Rows (top to bottom):** n-linear curve with adaptive mesh, $n-$linear curve with octree[17], parabolic curve with non-adaptive mesh, respectively, and our ray tracer. (Hitpoint and distance differences are normalized by the size of the scene, while direction differences are the norm of the difference vector.)

On the other hand, both the octree and the non-adaptive mesh are less compact than the adaptive mesh achieving the same level of accuracy, which

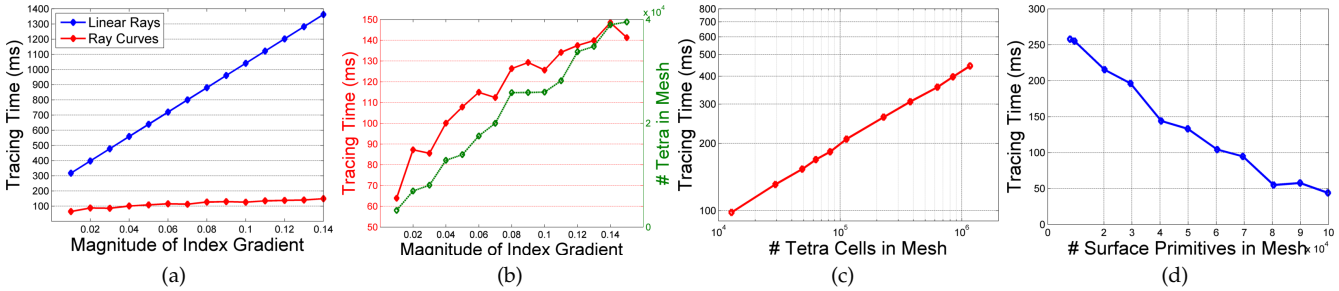


Fig. 8: **Performance and scalability of ray curve tracing**: **(a)** Tracing analytic ray curves vs. numerical integration for **A-LD**, computing propagation paths to the same accuracy. Ray curve tracing scales better with the magnitude of media variations. **(b)** A close-up view of the "Ray Curves" line plot in (a) shows the tracing time (red line) scaling with increasing mesh sizes (green line). The increasing mesh size is a result of the adaptive mesh constructed at a finer resolution for increased media gradient, to keep the approximation error at the same level. **(c,d)** Ray curve tracing scales sub-linearly with tetrahedral cell counts and number of boundary surfaces in the mesh. Note that tracing time decreases with increasing surface primitives because the average propagation distance before a ray bounces off a boundary surface is shortened.

Benchmark	Frame time	Compute Curves	Tetra Intersect (time)	Tetra Intersect (count)	Bisection
Elephant	123	0.0175 (0.01%)	110.08 (88.06%)	51	+108.49
Desert (m)	219	0.0658 (0.03%)	211.39 (96.24%)	179	+247.70
Desert (h)	369	0.1033 (0.03%)	361.59 (97.92%)	254	+443.12
Christmas (m)	259	0.1037 (0.04%)	240.96 (92.89%)	188	+220.98
Christmas (h)	443	0.1948 (0.04%)	427.99 (96.64%)	296	+392.73
Reservoir	233	0.0892 (0.03%)	236.87 (93.79%)	182	+255.03

TABLE 3: **Breakdown of curved-ray traversal time.** Intersection costs dominate the frame time, while ray curve formulation and computation cost is negligible. We also report the average number of tetrahedra that each ray curve traverses. For comparison with [17], we trace n-linear rays for the same scene configurations, and report the additional time that bisection takes in the rightmost column. Our ray formulations avoids the bisection computation due to their analytic surface intersections. All timings are in milliseconds.

lead to higher ray tracing costs. To further investigate the difference between the adaptive mesh we used and the octree used in [17], we analyze the error in media profile approximated by octrees using the same profile **A-LU+F** as Figure 12. According to [17], we merge octree nodes using two thresholds: δ, maximum refractive index difference, and ε, maximum difference in gradients of the refractive index that are allowed within a node. We vary both these thresholds to generate octrees of different resolutions, and we plot the corresponding relative error, computed as in Sec. 6.2 (Fig. 13 in Appendix). Overall the resulting octrees tend to have more nodes when they can achieve the same level of error as tetrahedral meshes. If comparable number of sample points are used, the interpolated profile from the octree by finite difference yields visibly less smooth media and larger errors (Fig. 13).

6.6 Performance of ray curve tracer

We analyze the performance of our ray tracer in greater detail (Figure 8), where we observe significant improvement and better scalability over numerical ray integration. For those results we use RK-4 with the intersection test approximated by piecewise linear segments. All the timings are collected on a single 3.2GHz CPU core. The running time of curved ray traversal scales sub-linearly with the number of tetrahedral cells in the mesh, as shown in Figure 8(b,c). In contrast, the numerical ray integration performance decreases greatly with media variation, and was capped at media gradient of the magnitude 0.15 to keep the running time reasonable.

The performance also scales well with increasing numbers of boundary surfaces (Figure 8(d)), which demonstrates the culling efficiency of the mesh. On the other hand, the numerical method incurs extra costs on surface intersections, even when traversing the same adaptive mesh used by the ray curve tracer. Besides the extra ray steps taken by numerical method, the other source of inefficiency is the fact that when an intersection is detected, the latest integration step needs to be restarted to go only as far as this intersection point, which adds significant amount of computation (see Table 3 under "Tetra Intersect (count)" for the number of restarts needed for numerical ray integration due to surface intersection).

6.7 Applications on outdoor acoustics

Ideally, outdoor acoustic simulations model fully general media based on temperature and wind profiles, as well as complex natural and man-made boundaries.

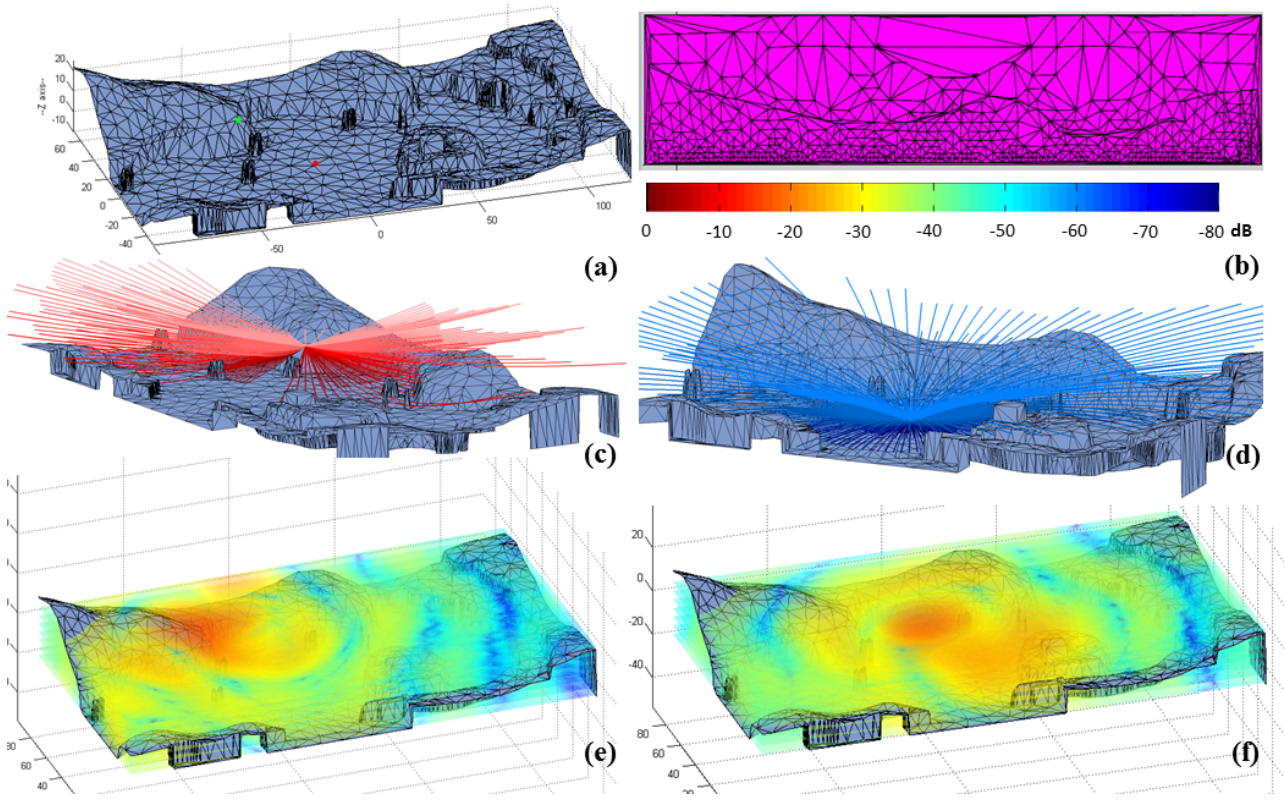


Fig. 9: **Outdoor acoustic applications. (a)** Wire-frame rendering of the Reservoir scene, the green and red dots are the two sound sources. **(b)** Side view of the adaptive mesh, notice the vertical gradation of cells as well as the embedded terrain surfaces. **(c,d)** Illustrative upward- and downward-refractive ray trace for the two sources, respectively. **(e,f)** The 3D pressure field computed based on the ray traced paths, here visualized in stacks of slices. We compute 10k ray paths at 4 fps, and the 3D pressure field is generated at 1 fps.

Existing methods either ignore the non-linear media, simplify the media by reducing the dimensions in its variations (e.g. assuming it is stratified or has only 2D variations), or requires off-line computations. By accelerating the ray models with analytic ray curves and adaptive mesh, we achieve interactivity with a 3D varying media profile and complex boundaries.

In Fig. 14 in Appendix, we highlight our method applied to different atmospheric profiles and the outdoor benchmarks Christmas and Desert. The ray plots in Fig. 14(a, d) are generated with stratified profiles (**A-LU, A-LD**), and the propagation paths do not deviate from a 2D plane. In contrast, Fig. 14(b, c, e, f) illustrate the ray paths with 3D fluctuations modeled in the profiles (**A-LU+F**, **A-LD+F**), as well as interacting with 3D geometry. For the Reservoir scene, which has terrains that shape the propagated sound field in a more significant way, we show the mesh constructed, the ray curves traced, and the pressure field computed from the propagation paths in Figure 9. Our interactive ray curve tracer for computing propagation paths enables fast generation of those acoustic field results.

7 Limitations and Future Work

There are several limitations to our approach. The first is that the adaptive unstructured mesh is precomputed. Therefore, our current implementation is limited to static environments. In dynamic scenes, our approach is limited to the cases where the media property changes do not invalidate the topology of the mesh. Secondly, the efficiency of tracing analytic ray curves depends on the existence of spatial coherence in media. Conceivably there will be a point when a chaotic media has little coherence that the valid range of analytic ray curves reduces to the same with linear ray steps. However, most natural media used for visual and acoustic simulation tends to be fairly coherent and varies smoothly; in these cases tracing analytic ray curves works quite well.

As future work, we would like to parallelize this approach on a multi-core CPUs or many-core GPUs. Each path of our analytic ray curves propagates independently from each other, thererfore our algorithm is as amenable to parallelism as linear ray tracing. We would also like to explore dynamically adapting the media mesh as in [45], [46], which can be useful for simulation of fluctuating or turbulent media, and of dynamic scenes. Another avenue for future work is to combine our method of simulating refractive propagation with complementary methods that simulate scattering and absorption in participating media.

8 Conclusions

We addressed the challenge of simulating sound and light propagation in large outdoor scenes with general varying media and complex media boundaries. We developed an efficient ray-tracing based algorithm that eliminates the need of making simplifying assumptions about the media variations or the scenes.

In particular, we traced analytic ray curves that overcome the step size limitation of linear rays, computed closed-form intersections of the ray curves with the scene objects, and constructed adaptive media mesh for efficient representation of the underlying general media profiles. The mesh is also able to conform to the media/objects boundaries, so that surface interactions can be computed seamlessly with media traversal, and the terrain or obstacle-following temperature and wind profiles, commonly found in real-world measurements[58], [59], can be modeled.

We highlight the propagation results on outdoor benchmarks with realistic atmospheric profiles and complex obstacles, running at near interactive rates on a single CPU core. Our algorithm enables fast sound simulation in large outdoor scenes that were not feasible with previous methods.

Acknowledgment

This research was supported by ARO Contracts W911NF-10-1-0506, W911NF-12-1-0430, W911NF-13-C-0037, and the National Science Foundation award 1320644.

References

[1] USGPC, "Us standard atmosphere," *United States Government Printing Office, Washington, DC*, 1976.
[2] A. S. Monin and A. M. Obukhov, "Basic regularity in turbulent mixing in the surface layer of the atmosphere," *Trudy Geofiz Inst, Akad Nauk SSSR*, no. 24, p. 151, 1954.
[3] J. A. Businger, J. C. Wyngaard, Y. Izumi, and E. F. Bradley, "Flux-profile relationships in the atmospheric surface layer," *Journal of the Atmospheric Sciences*, vol. 28, no. 2, pp. 181–189, 1971.
[4] H. A. Panofsky and J. A. Dutton, *Atmospheric Turbulence: Models and Methods for Engineering Applications*. New York: Wiley-Interscience, 1984.
[5] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, 2nd ed. Springer, 2011.
[6] E. M. Salomons, *Computational Atmospheric Acoustics*. Springer, 2001.
[7] K. Attenborough, K. M. Li, and K. Horoshenkov, *Predicting Outdoor Sound*. CRC Press, 2006.
[8] J. Kang, *Urban Sound Environment*. CRC Press, 2006.
[9] M. Berger, T. Trout, and N. Levit, "Ray tracing mirages," vol. 10, no. 3, pp. 36–41, 1990.
[10] F. K. Musgrave, "A note on ray tracing mirages," vol. 10, no. 6, pp. 10–12, 1990.
[11] Y. Zhao, Y. Han, Z. Fan, F. Qiu, Y. chuan Kuo, A. Kaufman, and K. Mueller, "Visual simulation of heat shimmering and mirage," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 1, pp. 179–189, 2007.
[12] J. J. Degnan, "Millimeter accuracy satellite laser ranging: a review," *Contributions of space geodesy to geodynamics: technology*, pp. 133–162, 1993.
[13] A. H. Dodson, "Refraction and propagation delays in space geodesy," *International Journal of Remote Sensing*, vol. 7, no. 4, pp. 515–524, 1986.
[14] C. S. Gardner, "Correction of laser tracking data for the effects of horizontal refractivity gradients," *Applied optics*, vol. 16, no. 9, pp. 2427–2432, 1977.
[15] V. Badescu, *Modeling Solar Radiation at the Earth Surface*. Springer, 2008.
[16] A. S. Glassner, Ed., *An introduction to ray tracing*. London: Academic Press, 1989.
[17] C. Cao, Z. Ren, B. Guo, and K. Zhou, "Interactive rendering of non-constant, refractive media using the ray equations of gradient-index optics," in *Proceedings of the Eurographics conference on Rendering*, 2010, pp. 1375–1382.
[18] U.S. Office of Naval Research, "Ocean acoustics library: Ray models/software," http://oalib.hlsresearch.com/Rays/index.html, 2014, [Online; accessed 15-August-2014].
[19] J. Stam, E. Languénou, and P. Syntim, "Ray tracing in non-constant media," in *Proceednigs of the Eurographics Workshop on Rendering*, 1996, pp. 225–234.
[20] D. Bernabei, A. Hakke Patil, F. Banterle, M. Di Benedetto, F. Ganovelli, S. Pattanaik, and R. Scopigno, "A parallel architecture for interactive rendering of scattering and refraction effects," *IEEE Computer Graphics and Applications*, vol. 32, no. 2, pp. 34–43, 2012.
[21] D. Gutierrez, F. J. Serón, A. Muñoz, and O. Anson, "Simulation of atmospheric phenomena," *Computers & Graphics*, vol. 30, no. 6, pp. 994–1010, 2006.
[22] T. R. Satoh, "Symplectic ray tracing: A new frontier in non-linear ray tracing," in *Procceedings of WSCG*, 2003.
[23] E. Gröller, "Nonlinear ray tracing: visualizing strange worlds," *The Visual Computer*, vol. 11, no. 5, pp. 263–274, 1995.
[24] D. Weiskopf, "Four-dimensional non-linear ray tracing as a visualization tool for gravitational physics," in *Proceedings of IEEE Visualization*, 2000, pp. 445–448.
[25] D. Gutierrez, A. M. noz, O. Anson, and F. J. Serón, "Non-linear volume photon mapping," in *Rendering Techniques*, 2005, pp. 291–300.
[26] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H.-P. Seidel, "Eikonal rendering: Efficient light transport in refractive objects," *ACM Transactions on Graphics*, p. 59, 2007.
[27] X. Sun, K. Zhou, E. Stollnitz, J. Shi, and B. Guo, "Interactive relighting of dynamic refractive objects," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 35:1–35:9, 2008.
[28] D. Weiskopf, T. Schafhitzel, and T. Ertl, "Gpu-based nonlinear ray tracing," *Computer Graphics Forum*, vol. 23, no. 3, pp. 625–634, 2004.
[29] A. Muñoz, D. Gutierrez, and F. J. Serón, "Optimization techniques for curved path computing," *The Visual Computer*, vol. 23, no. 7, pp. 493–502, 2007.
[30] V. Havran, "Heuristic ray shooting algorithms," Ph.D. dissertation, Faculty of Electrical Engineering, Czech Technical University, 2000.
[31] A. Lagae and P. Dutré, "Accelerating ray tracing using constrained tetrahedralizations," *Computer Graphics Forum*, vol. 27, no. 4, pp. 1303–1312, 2008.
[32] E. Cerezo, F. Pérez, X. Pueyo, F. J. Serón, and F. X. Sillion, "A survey on participating media rendering techniques," *The Visual Computer*, vol. 21, no. 5, pp. 303–328, 2005.
[33] D. Gutierrez, H. W. Jensen, W. Jarosz, and C. Donner, "Scattering," in *ACM SIGGRAPH Asia Courses*, 2009, pp. 15:1–15:620.
[34] Y. Yue, K. Iwasaki, B.-Y. Chen, Y. Dobashi, and T. Nishita, "Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media," in *Proceedings of ACM SIGGRAPH Asia*, 2010, pp. 177:1–177:8.
[35] K. Museth, "VDB: High-resolution sparse volumes with dynamic topology," *ACM Transactions on Graphics*, vol. 32, no. 3, p. 27, 2013.
[36] L. Szirmay-Kalos, B. Tóth, and M. Magdics, "Free path sampling in high resolution inhomogeneous participating media." *Computer Graphics Forum*, vol. 30, no. 1, pp. 85–97, 2011.
[37] Q. Fang, "Mesh-based monte carlo method using fast ray-tracing in plücker coordinates," *Biomedical Optics Express*, vol. 1, no. 1, pp. 165–175, 2010.
[38] G. Marmitt and P. Slusallek, "Fast ray traversal of tetrahedral and hexahedral meshes for direct volume rendering," in *Proceedings of the Joint Eurographics / IEEE VGTC conference on Visualization*, 2006, pp. 235–242.

[39] I. Wald, H. Friedrich, and A. Knoll, "Interactive isosurface ray tracing of time-varying tetrahedral volumes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1727–1734, 2007.

[40] P. Muigg, M. Hadwiger, H. Doleisch, and M. E. Gröller, "Interactive volume visualization of general polyhedral grids," *IEEE Transaction on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2115–2124, 2011.

[41] F. M. Miranda and W. C. Filho, "Volume rendering of unstructured hexahedral meshes," *The Visual Computer*, vol. 28, no. 10, pp. 1005–1014, 2012.

[42] M. Bußler, T. Rick, A. Kelle-Emden, B. Hentschel, and T. Kuhlen, "Interactive particle tracing in time-varying tetrahedral grids," in *Proceedings of the Eurographics conference on Parallel Graphics and Visualization*, 2011, pp. 71–80.

[43] I. Wald, H. Friedrich, G. Marmitt, and H.-P. Seidel, "Faster isosurface ray tracing using implicit kd-trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, 2005.

[44] A. Rúger and D. Hale, "Meshing for velocity modeling and ray tracing in complex velocity fields," *Geophysics*, vol. 71, no. 1, pp. U1–U11, 2006.

[45] P. Cignoni, C. Montani, E. Puppo, and R. Scopigno, "Multiresolution representation and visualization of volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 352–369, 1997.

[46] P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno, "Multiresolution modeling and visualization of volume data based on simplicial complexes," in *Proceedings of the Symposium on Volume Visualization*, 1994, pp. 19–26.

[47] D. P. Bacon, N. N. Ahmad, Z. Boybeyi, T. J. Dunn, M. S. Hall, P. C. S. Lee, R. A. Sarma, M. D. Turner, K. T. Waight, S. H. Young, and J. W. Zack, "A dynamically adapting weather and dispersion model: The operational multiscale environment model with grid adaptivity (omega)," *Monthly Weather Review*, vol. 128, no. 7, pp. 2044–2076, 2000.

[48] T. E. Dowling, "Earth general circulation models," *Comparative Climatology of Terrestrial Planets*, vol. 1, pp. 193–211, 2013.

[49] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7th ed. Cambridge University Press, 1999.

[50] A. Ghatak, *Optics*, 3rd ed. Tata McGraw-Hill, 2005.

[51] Y. A. Kravtsov and Y. I. Orlov, *Geometrical Optics of Inhomogeneous Media*. Springer, 1990.

[52] Y. Qiao, "The solution of the ray-path differential equation in the axial gradient-index media," *Acta Optica Sinica*, vol. 4, no. 1, pp. 89–92, 1984.

[53] W. B. Kerr, F. Pellacini, and J. D. Denning, "Bendylights: artistic control of direct illumination by curving light rays," in *Proceedings of the Eurographics conference on Rendering*, 2010, pp. 1451–1459.

[54] F. Grave, T. Müller, C. Dachsbacher, and G. Wunner., "The Gödel Engine - An interactive approach to visualization in general relativity," *Computer Graphics Forum*, vol. 28, no. 3, pp. 807–814, 2009.

[55] B. Roberts, "Horizontal-gradient acoustical ray-trace program trimain," U.S. Naval Research Laboratory, Washington, DC, Tech. Rep. Rep. 7827, 1974.

[56] W. H. Watson and R. W. McGirr, "Raywave ii: A propagation loss model for the analysis of com- plex ocean environments," U.S. Naval Ocean Systems Center, San Diego, CA, Tech. Rep. Rep. TN-1516, 1975.

[57] E. Khular, K. Thyagarajan, and A. K. Ghatak, "A note on mirage formation," *American Journal of Physics*, vol. 45, no. 1, pp. 90–92, 1977.

[58] A. L'Espérance, J. Nicolas, D. K. Wilson, D. W. Thomson, Y. Gabillet, and G. Daigle, "Sound propagation in the atmospheric surface layer: Comparison of experiment with FFP predictions," *Applied Acoustics*, vol. 40, no. 4, pp. 325–346, 1993.

[59] J. S. Lamancusa and P. A. Daroux, "Ray tracing in a moving medium with two-dimensional sound-speed variation and application to sound propagation over terrain discontinuities," *The Journal of the Acoustical Society of America*, vol. 93, no. 4, pp. 1716–1726, 1993.

[60] P. S. Jackson and J. C. R. Hunt, "Turbulent wind flow over a low hill," *Quarterly Journal of the Royal Meteorological Society*, vol. 101, no. 430, pp. 929–955, 1975.

[61] T. R. Oke, *Boundary Layer Climates*, 2nd ed. London: Routledge, 1988.

[62] M. Ament, C. Bergmann, and D. Weiskopf, "Refractive radiative transfer equation," *ACM Transactions on Graphics*, vol. 33, no. 2, pp. 17:1–17:22, 2014.

[63] H. Si and K. Gärtner, "Meshing piecewise linear complexes by constrained delaunay tetrahedralizations," in *Proceedings of the International Meshing Roundtable*. Springer, 2005, pp. 147–163.

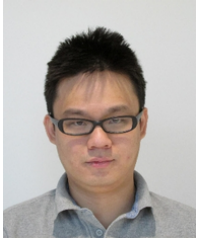
[64] J. Bronson, J. A. Levine, and R. Whitaker, "Lattice cleaving: A multimaterial tetrahedral meshing algorithm with guarantees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 2, pp. 223–237, 2014.

[65] C. D. Correa, R. Hero, and K.-L. Ma, "A comparison of gradient estimation methods for volume rendering on unstructured meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 3, pp. 305–319, 2011.

[66] D. J. Mavriplis, "Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes," in *Proceedings of the AIAA Computational Fluid Dynamics conference*, no. 3986, 2003.

[67] R. M. Jones, J. P. Riley, and T. M. Georges, *HARPA: A Versatile Three-dimensional Hamiltonian Ray-tracing Program for Acoustic Waves in the Atmosphere above Irregular Terrain*. U.S. Department of Commerce, National Oceanic and Atmospheric Administration, Environmental Research Laboratories, 1986.

Qi Mo received Bachelors degree in Computer Science (2004) from Nanjing University and Masters degree (2006) from the University of Iowa. She is currently a Ph.D. student in the Computer Science Department at the University of North Carolina at Chapel Hill. She has worked as a research intern at NVIDIA. Her research interests include interactive sound propagation and ray tracing.



Hengchin Yeh received the Bachelor's degree in Physics (2002), Master's degree in Computer Science and Information Engineering (2004) from National Taiwan University, and Ph.D. degree in Computer Science (2014) from University of North Carolina at Chapel Hill. He has previously been a research intern at Google and Disney Research and is now working full-time at Google. His research interests are physically-based sound simulation for graphics applications, including sound synthesis and sound propagation.

Dinesh Manocha is currently a Distinguished Professor of Computer Science at the University of North Carolina at Chapel Hill. He has published more than 370 papers in the leading conferences and journals in computer graphics, robotics, computational geometry, databases, multimedia, high performance computing, and symbolic computation and received 12 best paper award. Manocha has served as program committee member or program chair of more than 100 leading conferences and as a member of the editorial board or guest editor of twelve leading journals. He is a Fellow of ACM, IEEE, and AAAS and received Distinguished Alumni Award from Indian Institute of Technology, Delhi.

APPENDIX A
DERIVATION OF ANALYTIC RAY CURVES

Here we provide the derivation of analytic ray curves based on a locally constant gradient of the propagation speed c and of the squared refractive index n. The analytic solutions in various forms have been derived in different context including geometric optics [49], [50], [51] and computational acoustics [5], [6].

A.1 c-linear profile

When the propagation speed c has a local gradient ∇c, we take the direction of ∇c as the z-axis, and the local media profile can be written as:

$$c(z) = c_0 + \alpha z. \tag{15}$$

From Equation (1a) we have

$$\frac{d}{ds}\left(\frac{1}{c}\frac{dx}{ds}\right) = 0, \frac{d}{ds}\left(\frac{1}{c}\frac{dy}{ds}\right) = 0, \frac{d}{ds}\left(\frac{1}{c}\frac{dz}{ds}\right) = -\frac{1}{c^2}\frac{\partial c}{\partial z}. \tag{16}$$

We use the following symbols

$$\xi_0 = \frac{1}{c}\frac{dx}{ds}, \quad \eta_0 = \frac{1}{c}\frac{dy}{ds}, \quad \zeta(s) = \frac{1}{c}\frac{dz}{ds}, \tag{17}$$

and we can see that ξ_0 and η_0 are constant along the ray trajectory according to Equation (16). As a result,

$$\xi_0^2 + \eta_0^2 + \zeta^2 = \frac{1}{c^2}\left(\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2\right) = \frac{1}{c^2}. \tag{18}$$

If we rotate the x-y plane around the z axis until η_0 becomes 0 and put the origin of the coordinate system at the ray origin, the ray becomes a plane curve lying in the plane formed by the z-axis and the initial ray direction at the origin (Figure 1 in the paper), which we call the *ray plane*. The other axis of the *ray plane* is called axis r, and that $\xi_0' = \frac{1}{c}\frac{dr}{ds} = \frac{cos\theta_0}{c_0}$, where θ_0 is the angle between initial ray direction and the r axis. In the ray plane, integrating $\frac{dr}{ds}$ along the ray gives

$$r(s_t) = \int_{s_0}^{s_t} \frac{dr}{ds} ds = \xi_0' \int_{s_0}^{s_t} c ds = \xi_0' \int_{z_0}^{z} \frac{dz}{\zeta}. \tag{19}$$

We solve ζ from Equation (18) and plug it into Equation (19), which gives us a circular curve:

$$r(z) = \frac{\sqrt{1 - \xi_0'^2 c_0^2} - \sqrt{1 - \xi_0'^2 (c_0 + \alpha z)^2}}{\xi_0' \alpha}. \tag{20}$$

A.2 n^2-linear profile

When the squared refractive index n^2 has a local gradient ∇n^2, we denote the gradient direction direction as the z-axis, so that:

$$n^2(z) = n_0^2 + \alpha z. \tag{21}$$

From Equation (1b), and using a derivation analogous to Equation (16), (17), and (18), we obtain:

$$\frac{d}{ds}\left(n\frac{dx}{ds}\right) = 0, \frac{d}{ds}\left(n\frac{dy}{ds}\right) = 0, \frac{d}{ds}\left(n\frac{dz}{ds}\right) = \frac{\partial n}{\partial z}, \tag{22}$$

$$\xi_0 = n\frac{dx}{ds}, \quad \eta_0 = n\frac{dy}{ds}, \quad \zeta(s) = n\frac{dz}{ds}, \tag{23}$$

$$\xi_0^2 + \eta_0^2 + \zeta^2 = n^2\left(\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2 + \left(\frac{dz}{ds}\right)^2\right) = n^2. \tag{24}$$

We perform a similar rotation to the ray plane with axis r and z, and denote $\xi_0' = n\frac{dr}{ds} = n_0 cos\theta_0$. As in Equation (19), we obtain:

$$r(s_t) = \int_{s_0}^{s_t} \frac{dr}{ds} ds = \xi_0' \int_{s_0}^{s_t} \frac{ds}{n} = \xi_0' \int_{z_0}^{z} \frac{dz}{\zeta}. \tag{25}$$

We solve ζ from Equation (24) and plug it into Equation (25) to derive the ray trajectory:

$$r(z) = \frac{2\xi_0'}{\alpha}\left(\sqrt{-\xi_0'^2 + n_0^2 + \alpha z} - \sqrt{-\xi_0'^2 + n_0^2}\right), \tag{26}$$

which is a parabolic curve.

APPENDIX B
ANALYTIC SOLUTIONS OF n-LINEAR AND MIRAGE PROFILES

In Sec. 3.1 we defined three media profiles: n-linear, inferior mirage, and superior mirage, that have known analytic solutions for trajectories, and those solutions are used in Sec. 6.3, 6.4 as ground truth to validate the ray tracing results of both our analytic ray curve tracer and the numerical ray integration methods we are compating to. Here we give those analytic solutions for reference, more details and plots can be found in prior work [17], [52], [57].

B.1 n-linear profile

With n-linear profile, $n(z) = n_0 + \alpha z$, $\alpha = \|\nabla n\|$, n_0 is c at ray origin. Given the ray origin and initial direction $\mathbf{d}$, the trajectory is a plane curve that lies in the *ray plane*, formed by z-axis which coincides with the direction of ∇n and $\mathbf{d}$. We denote the axis within the *ray plane* that is perpendicular to z-axis as the r-axis, and θ_0 is the angle between $\mathbf{d}$ and r axis. The analytic trajectory in r-z coordinates can be derived to be:

$$r(z) = \frac{n_0 cos\theta_0}{\alpha} \ln(2cz + b + 2\alpha\sqrt{A + bz + cz^2}|_0^z, \tag{27}$$

where $A = {n_0}^2 \sin_{\theta_0}{}^2, b = 2\alpha n_0, c = \alpha^2$, when the angle between $\mathbf{d}$ and the z-axis is less than $\pi/2$.

When the angle between $\mathbf{d}$ and the z-axis is equal to or greater than $\pi/2$, the analytic trajectory can be divided into two segments at the point $f = (x_f, z_f)$, where

$$z_f = \frac{n_0 \cos\theta_0 - n_0}{\alpha}, x_f = -r(z_f). \tag{28}$$

The two segments are: $x = -r(z), 0 \le x \le x_f$ and $x = 2x_f + r(z), x > x_f$.

B.2 Inferior mirage

For the inferior mirage profile (V-IM) given in Sec. 3.1, and in the coordinate system where z-axis represents the height and r-axis represents the range, and the origin of the coordinate system is taken to be at the same range with the ray origin but with a height on the ground. z_0 is the height of the ray origin in this coordinate, $theta_0$ is the angle between ray initial

direction $\mathbf{d}$ and r-axis, and the analytic trajectory has been derived to be:

$$r(z) = \frac{n(z_0)\cos\theta_0}{\beta\sqrt{B}} \ln \frac{\sqrt{B} + \sqrt{B - Cexp(-\beta z)}}{\sqrt{B} - \sqrt{B - Cexp(-\beta z)}}|_{z_0}^{z}, \tag{29}$$

where

$$B = \mu_0^2 + \mu_1^2 - (n(z_0)\cos\theta_0)^2, C = \mu_1^2, \tag{30}$$

when the angle between $\mathbf{d}$ and z-axis is less than $\pi/2$.

When the angle between $\mathbf{d}$ and z-axis is equal to or greater than $\pi/2$, the trajectory is divided into two segments by the point (x_f, z_f), where

$$z_f = -\frac{1}{\beta} \ln (1 - \frac{n^2(z_0)cos^2\theta_0 - \mu_0^2}{\mu_1^2}), x_f = r(z_f). \tag{31}$$

B.3 Superior mirage

For the superior mirage profile (V-IM) given in Sec. 3.1, the coordinate system is similarly defined where z-axis represents the height, r-axis represents the range, and the origin is at the same range with the ray origin but with a height on the ground. z_0 is the height of the ray origin in this coordinate, $theta_0$ is the angle between ray initial direction $\mathbf{d}$ and r-axis, and the analytic trajectory has been derived to be:

$$r_1(z) = \frac{n(z_0)\cos\theta_0}{\beta\sqrt{D}} \ln \frac{\sqrt{D + Cexp(-\beta z)} + \sqrt{D}}{\sqrt{D + Cexp(-\beta z)} - \sqrt{D}}|_{z_0}^{z}, \tag{32}$$

for $cos\theta_0 \leq \sqrt{\frac{\mu_0^2}{\mu_0^2 + \mu_1^2 exp(-\beta z_0)}}$, and

$$r_2(z) = \frac{-2n(z_0)cos\theta_0}{\beta\sqrt{-D}} \arctan(\sqrt{-\frac{C}{D}exp(-\beta z) - 1})|_{z_0}^{z}, \tag{33}$$

for $cos\theta_0 > \sqrt{\frac{\mu_0^2}{\mu_0^2 + \mu_1^2 exp(-\beta z_0)}}$, where

$$D = \mu_0^2 - (n(z_0)\cos\theta_0)^2, C = \mu_1^2, \tag{34}$$

when the angle between $\mathbf{d}$ and z-axis is less than $\pi/2$.

When the angle between $\mathbf{d}$ and z-axis is equal to or greater than $\pi/2$, the trajectory is divided into two segments by the point (x_f, z_f), where

$$z_f = -\frac{1}{\beta} \ln (\frac{n^2(z_0)cos^2\theta_0 - \mu_0^2}{\mu_1^2}), x_f = r_2(z_f). \tag{35}$$

APPENDIX C
CONSIDERATIONS OF EMBEDDED MESH

The design choice of whether to embed the boundary surfaces or not, as discussed in Section 5.2, depends on whether the resolution of surface tessellations matches the resolution of media variations. Here we show this connection with the geometric representations used for acoustic benchmarks, Christmas and Desert scenes. We tessellate the boundary surfaces in these benchmarks to different resolutions, using the same set of media samples, and construct a different constrained tetrahedral mesh for each resolution.

As shown in Fig. 16(a), there is a particular range of resolution for each scene at which the surface tessellation and the adaptive media mesh resolution match each other; other tessellation levels produce lower-quality mesh with more cells. This effect is even more apparent (Fig. 16(b)) when we build optimized tetrahedral mesh with a quality threshold measured in the average aspect ratio.

Even when we link boundary surfaces with the media cells they overlap with, rather than embedding them in the mesh, we can see from 16(c) and (d) that a mismatch between surface tessellation and media variation still leads to slower traversal. Even though the mesh is not affected by the surface tessellation in this scenario, the number of surface primitives that overlap each media cell increases with finer surface tessellation, which slows down the traversal.

While constrained mesh construction is more expensive than unconstrained mesh construction, the linking of surfaces also results in significant cost in terms of pre-processing, as shown in Figure 16(e). Given a complex media profile with boundaries tessellated at a compatible resolution, the lower traversal time for mesh with embedded boundary surfaces may be worth the extra construction cost.

APPENDIX D
GRADIENT ESTIMATION SOLUTIONS

With linear least square, the estimated gradient from solving Equation (13) is:

$$\nabla m(\mathbf{x}_0) = \sum_{k=1}^{4} \mathbf{p}_k (m(\mathbf{x}_k) - m(\mathbf{x}_0)) \tag{36}$$

The coefficients, $\mathbf{p}_k$ are:

$$\mathbf{p}_k = \begin{bmatrix} \alpha_{k,1} - \frac{r_{12}}{r_{11}}\alpha_{k,2} + \beta\alpha_{k,3} \\ \alpha_{k,2} - \frac{r_{23}}{r_{22}}\alpha_{k,3} \\ \alpha_{k,3} \end{bmatrix} \tag{37}$$

where

$$\alpha_{k,1} = \frac{\Delta x_k}{r_{11}^2} \tag{38a}$$

$$\alpha_{k,2} = \frac{1}{r_{22}^2}(\Delta y_k - \frac{r_{12}}{r_{11}}\Delta x_k) \tag{38b}$$

$$\alpha_{k,3} = \frac{1}{r_{33}^2}(\Delta z_k - \frac{r_{23}}{r_{22}}\Delta y_k + \beta\Delta x_k) \tag{38c}$$

$$\beta = \frac{r_{12}r_{23} - r_{13}r_{22}}{r_{11}r_{22}} \tag{38d}$$

and

$$r_{11} = \sqrt{\sum_{k=1}^{4} w_k(\Delta x_k)^2} \tag{39a}$$

$$r_{12} = \frac{1}{r_{11}} \sum_{k=1}^{4} w_k \Delta x_k \Delta y_k \tag{39b}$$

$$r_1 3 = \frac{1}{r_{11}} \sum_{k=1}^{4} w_k \Delta x_k \Delta z_k \tag{39c}$$

$$r_{22} = \sqrt{\sum_{k=1}^{4} w_k(\Delta y_k)^2 - r_{12}^2} \tag{39d}$$

$$r_{23} = \frac{1}{r_{22}}\left(\sum_{k=1}^{4} w_k \Delta y_k \Delta z_k - \frac{r_{12}}{r_{11}} \sum_{k=1}^{4} w_k \Delta x_k \Delta z_k\right) \quad (39e)$$

$$r_{33} = \sqrt{\sum_{k=1}^{4} w_k (\Delta z_k)^2 - (r_{13}^2 - r_{23}^2)} \quad (39f)$$

where $\Delta(.) = (.)_k - (.)_0$, and x_k, y_k, z_k are the Cartesian coordinates of $\mathbf{x}_k$.

In contrast, with Green-Gauss gradient estimation as used in [17], given a tetrahedral cell with media properties m defined on its vertices $\{m_k, k = 1, ..., 4\}$, the gradient within that cell is given by:

$$\nabla m = \sum_{k=1}^{4} \frac{A_k m_k}{T} N_k, \quad (40)$$

where T is the volume of the tetrahedral cell, and A_k, N_k are the area and the normal of the face opposite to vertex k, respectively.

This Barycentric interpolation leads to C^0-continuity of the media property, m, across shared faces, edges, and vertices of neighboring cells. However, there can be discontinuity in the media gradient between neighboring cells. It has been mentioned as future work in [17] that continuity in gradient could potentially remove certain visual artifacts, and this improvement is even more important for acoustic applications than visual ones.

APPENDIX E
COMPARISON OF MESHES GENERATED FROM LOCAL GRADIENTS OF n, c, n^2

For any general media profile, whether given in the propagation speed c or in the refractive index n, we could transform the input profile into equivalent profiles of n, c, or n^2 based on the relation $n = c_0/c$. The media gradient in the form of ∇n, ∇c, or ∇n^2 can be computed respectively, and a different adaptive mesh can be constructed using Algorithm 2 for each of the gradient measures, to be traversed by the n-linear, c-linear (circular), and n^2-linear (parabolic) rays.

In this Appendix we analyze the approximation errors associated with each of the three kinds of meshes, for the profiles A-LU+F and A-DU+F, in Figure 17 and 18, respectively. Overall the approximations of the underlying media are at the same accuracy level across different kinds of meshes with comparable size (number of cells). One of the meshes may be better at approximating specific media profiles, but the differences are small. We therefore recommend selecting among the three meshes on a per scene basis, but since the difference is small, c-linear and n^2-linear profiles may be better choices due to their more efficient boundary intersections.

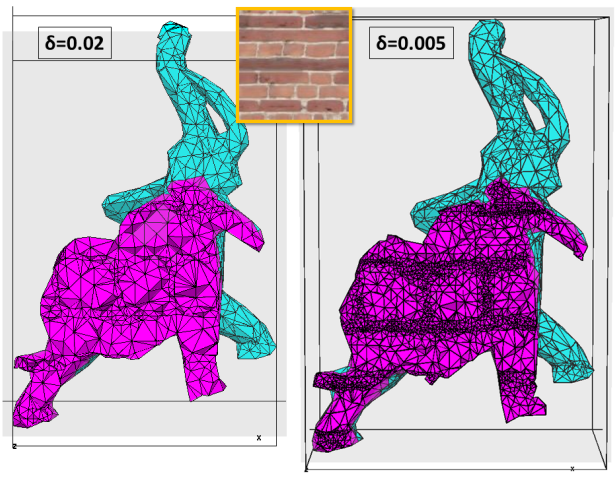


Fig. 10: **Adaptive meshes** generated for the elephant model[17], the interior index-of-refraction following the luminance of the brick texture. Meshes built with two different δ values are shown. The mesh cell sizes can be seen to adapt to the underlying media gradient.

(a) ray curves, 7.3 fps (b) ray stepping (size 1.0), 6.92 fps (c) ray stepping (size 0.05), 0.22 fps

(d) photograph of superior mirage (e) diff. between (a) and (b) (f) diff. between (a) and (c)

(g) ray curves, 8.9 fps (h) ray stepping (size 1.0), 8.2 fps (i) ray stepping (size 0.05), 0.35 fps

(j) photograph of inferior mirage (k) diff. between (g) and (h) (l) diff. between (g) and (i)

Fig. 11: **Same-quality/same-speed comparisons** between ray curve tracing and piece-wise linear ray tracing, on visual benchmarks of superior mirages **V-SM (a-f)** and inferior mirages **V-IM (g-l)**. The atmospheric media is modeled with an adaptive mesh of 28,313 tetrahedral cell, covering a physical volume of $50\text{m}\times50\text{m}\times400\text{m}$. 512×512 rays are traced from the viewer position for each image. **(a,g)** Ray curve tracing results, **(d,j)** Photographs of similar phenomena, **(b,h)** Same-speed comparison, the size of ray steps is chosen to match the performance of ray curve tracing, **(e,k)** Difference images, **(c,i)** Same-quality comparison, the size of ray steps is chosen to match the rendering quality of ray curve tracing, **(f,l)** Difference images. The ray curve tracer is more efficient than piece-wise linear rays when rendering at comparable quality. With same speed comparison the artifacts from piece-wise linear rays are most visible in areas where trajectories are curved the most. All frame rates are measured with single CPU thread.

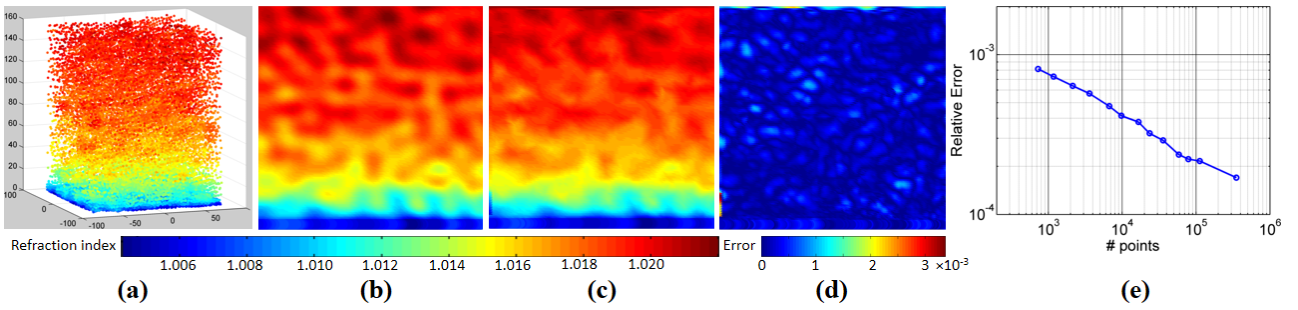


Fig. 12: **Approximation error of adaptive meshes.** Approximating the stratified-plus-fluctuation (**A-LU+F**) atmospheric profile using re-sampled points S containing $100\times$ fewer points than the input profile. **(a)** The positions of S color-coded by the index of refraction. **(b,c)** The original and approximated index of refraction n_G, $\tilde{n}_G$ on a slice, respectively. **(d)** Absolute error, $|n_G - \tilde{n}_G|$. **(e)** Relative error $E_{rel} = \|n_G - \tilde{n}_G\|/\|n_G\|$ versus the number of resampled points in S. The original grid has 2.09×10^5 $(128 \times 128 \times 128)$points.

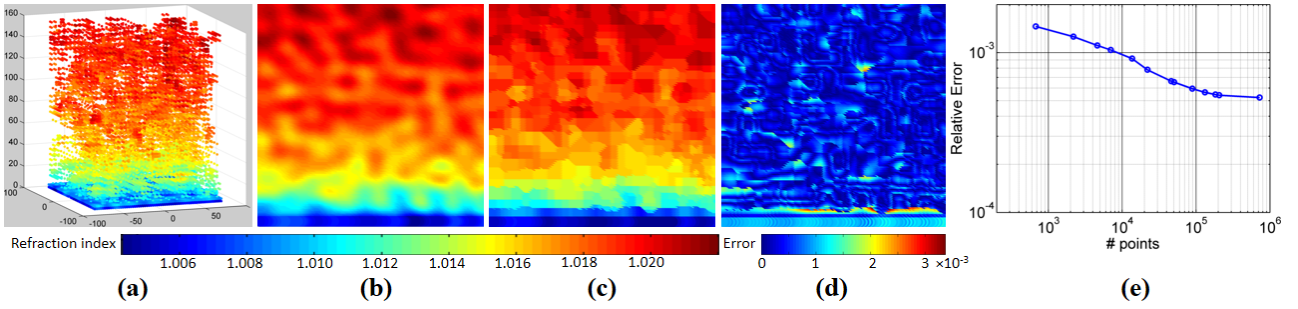
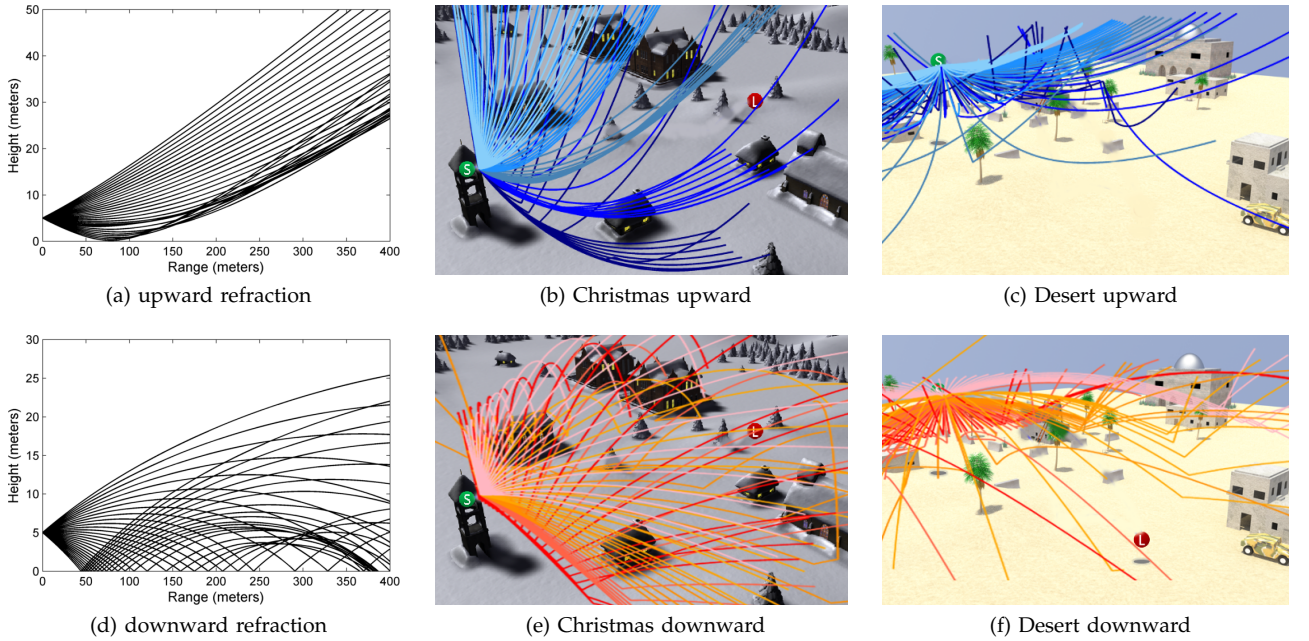


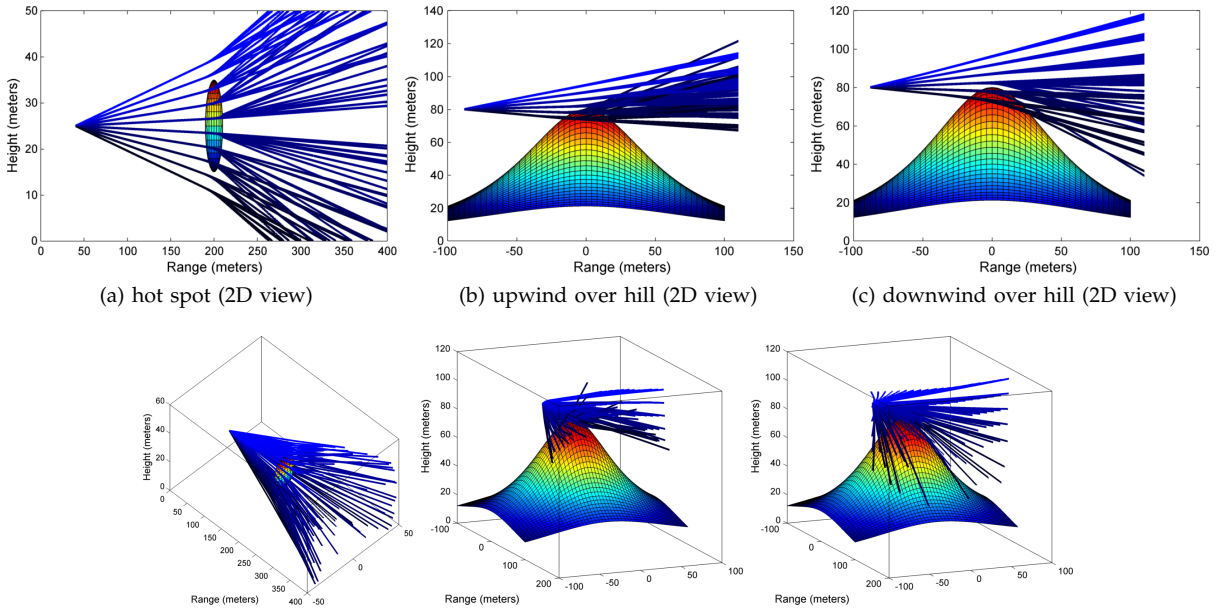
Fig. 13: **Compare to approximation error using octree.** We approximate the **A-LU+F** profile using octree, in comparison to our method of adaptive mesh analyzed in Figure 12. We build an octree given the same input media profile on a regular grid of $128\times128\times128$ points, using the same method as [17]. For the particular octree in (a-d) we use the threshold for differences in indices of refraction $\delta = 0.003$ and the threshold for differences in index gradients $\varepsilon = 0.0003$, to get similar numbers of samples (26,923) as in the re-sampled points S. **(a)** The positions of centers of octree cells, color-coded by the refractive index. **(b,c)** The original and approximated index of refraction n_G, $\tilde{n}_G$ on a slice. **(d)** Absolute error, $|n_G - \tilde{n}_G|$. **(e)** Relative error $E_{rel} = \|n_G - \tilde{n}_G\|/\|n_G\|$ versus the number of octree cells. The original grid has 2.09×10^5 points.



(a) upward refraction (b) Christmas upward (c) Desert upward

(d) downward refraction (e) Christmas downward (f) Desert downward

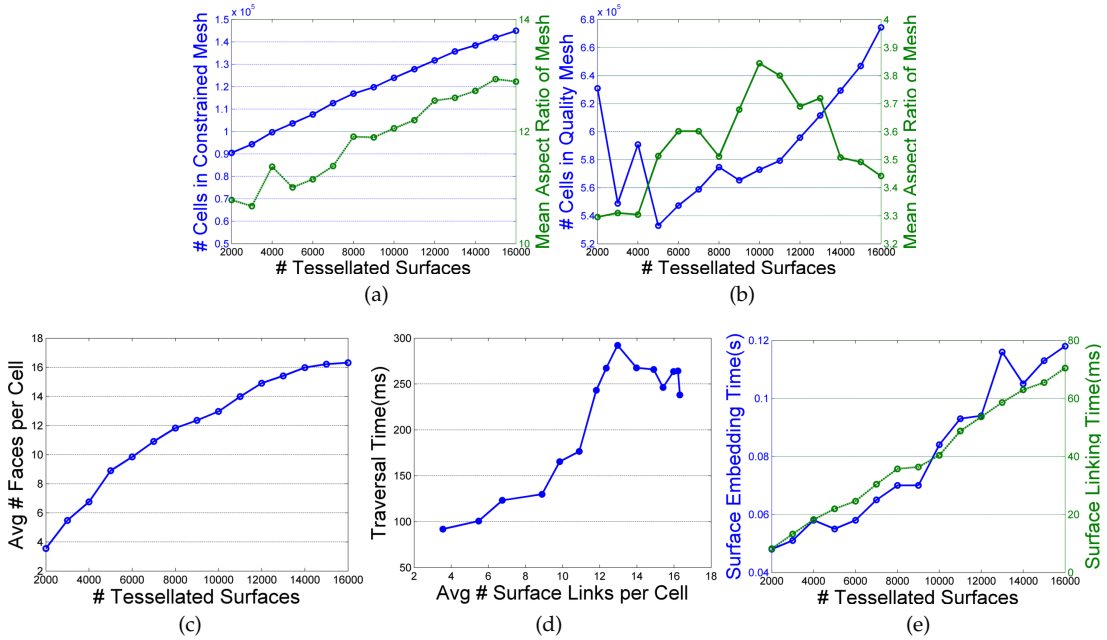
Fig. 14: **Acoustic propagation in the atmosphere. (a,d)** 2-D views with stratified profiles **A-LU**, **A-LD**. **(b,c,e,f)** Curved ray trajectories for Christmas and Desert benchmarks. Both upward **(A-LU+F)** and downward refractive **(A-LD+F)** atmosphere with random fluctuations are simulated. We trace 10K rays for up to 3 surface reflections at 4.5 fps for Desert(m) and 3.8 fps for Christmas(m). Here we show a representative set of ray paths for each scene and condition. The detailed performance results are listed in Table 3.



(a) hot spot (2D view) (b) upwind over hill (2D view) (c) downwind over hill (2D view)

(d) hot spot (3D view) (e) upwind over hill (3D view) (f) downwind over hill (3D view)

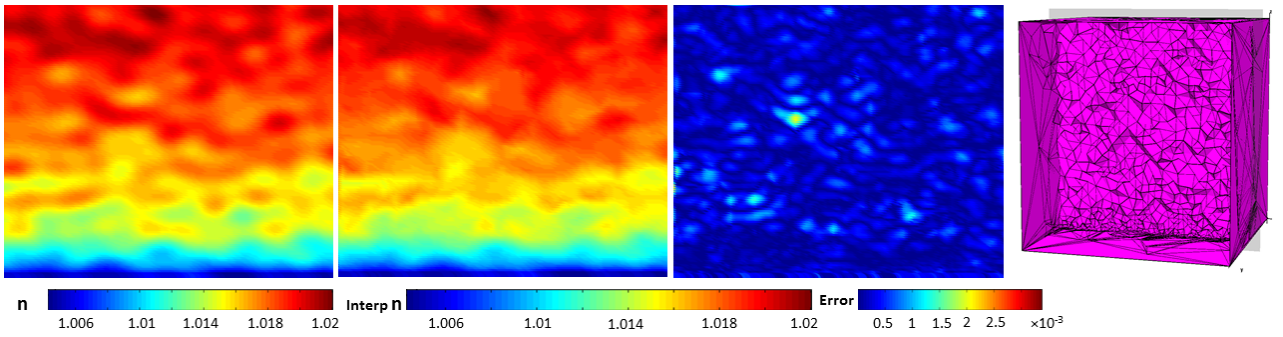
Fig. 15: **Acoustic propagation.** Illustrative ray paths are shown here for the acoustic profiles (defined in Sec. 3.2), including **(a,d)** Hot spot (**A-HS**) (the sphere shows the location and influence region of a heat source), **(b,e)** Up-wind propagation (**A-UW**) and **(c,f)** Downwind propagation over a hill (**A-DW**). The acoustic propagation trajectories deviate significantly from linear paths, and we show the out-of-plane propagation for A-HS, A-UW, and A-DW each from two different views. Our curved ray tracer computes the resulting 3D paths accurately at 10× the speed of linear ray stepping.



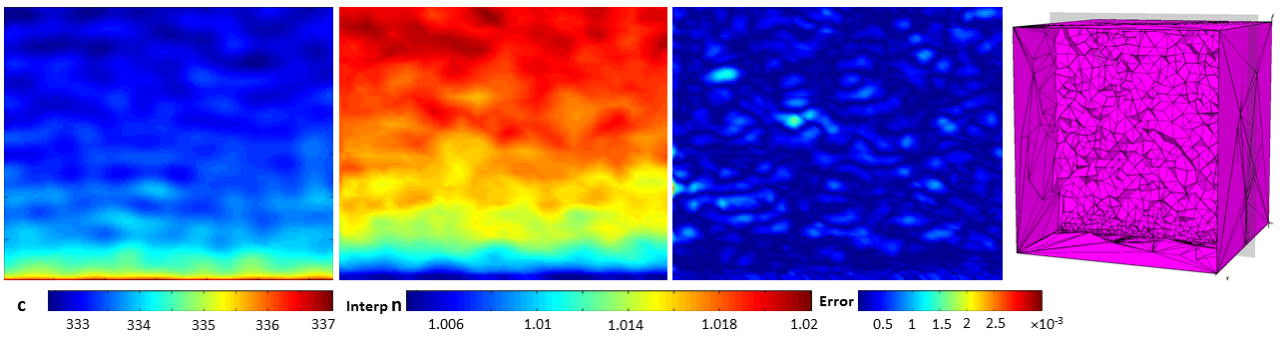
(a) (b)

(c) (d) (e)

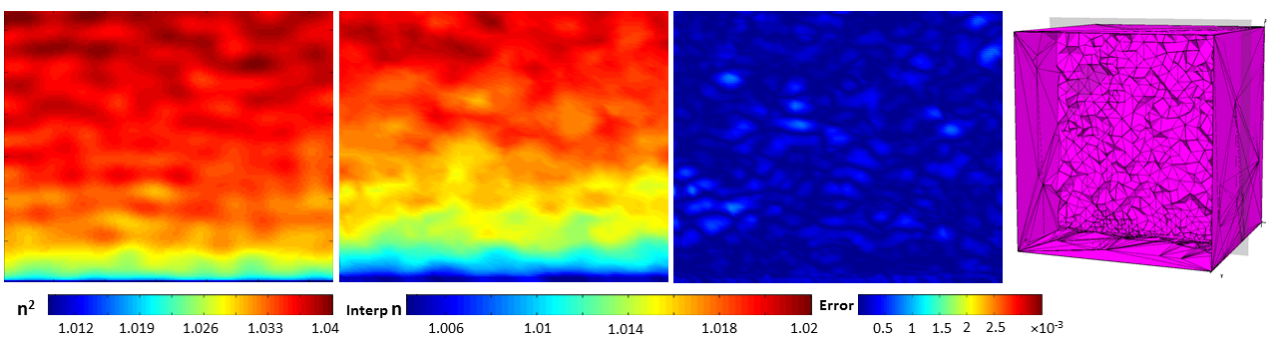
Fig. 16: **Comparison between embedding and linking boundary surfaces.** Tessellation of surfaces: **(a)** impacts the sizes and quality of the constrained mesh, the mesh quality reaches a high point (low mean aspect ratio) for tessellation that matches the surrounding media sample density. **(b)** impacts the sizes of quality meshes, which are constrained meshes that are optimized to achieve a quality threshold. The size of quality mesh is most compact when the tessellation matches the surrounding media sample density. **(c)** impacts the number of surfaces overlapping with each tetrahedral cell, which need to be linked. **(d)** Average number of surface links in turn impacts the traversal performance. **(e)** Tessellation of surfaces impacts the construction time of both embedding and linking.



(a)

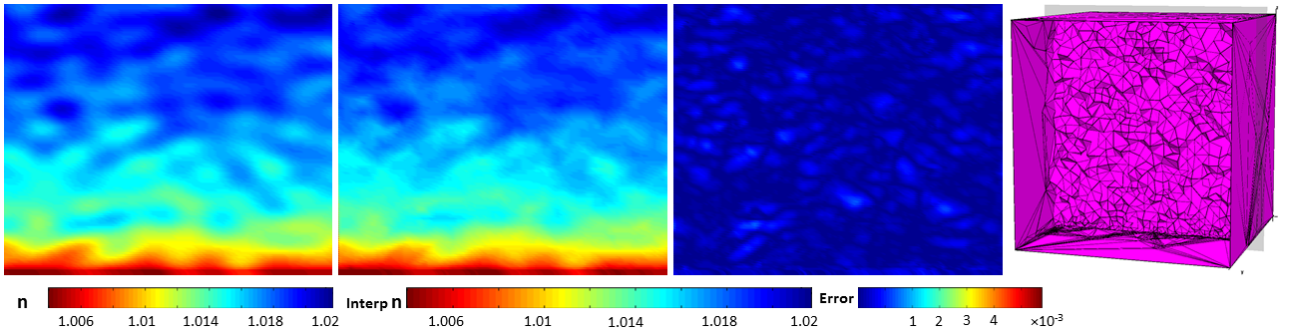


(b)

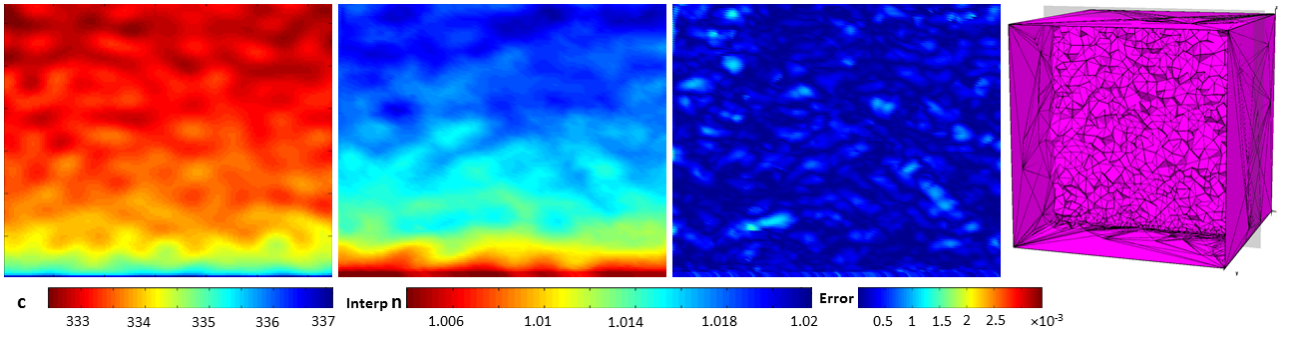


(c)

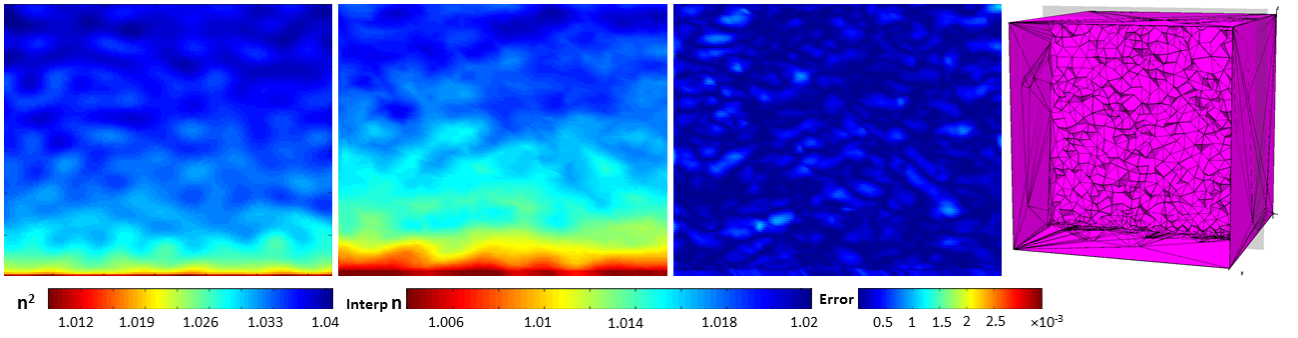
Fig. 17: **Comparisons of 3 analytic ray profiles: upward refractive atmosphere.** With the **A-LU+F** profile (defined in Sec. 3.2), we compute the same media profile in terms of c (sound speed), n (acoustic refractive index, with reference $c_0 = 340m/s$), and n^2, visualized in the leftmost column of **a,b,c**, respectively. The adaptive meshes constructed according to Algorithm 2 are shown in the rightmost column of **a,b,c**, with the control parameters $\delta = 0.001, 0.35, 0.023$, respectively. The control parameters are selected to achieve similar level of approximation error (measured in n and visualized in the second column from right) in the interpolated profiles over the three meshes. The resulting meshes have cell counts of $153867, 138965, 119670$ respectively, which are roughly on the same level, with the n^2-linear profile producing slightly more compact mesh than the other profiles.



(a)



(b)



(c)

Fig. 18: **Comparisons of 3 analytic ray profiles: downward refractive atmosphere.** With the profile **A-LD+F** defined in Sec. 3.2, we repeat the experiment in Figure 17. The equivalent media profiles in terms of n, c, and n^2 are shown in leftmost column in **a,b,c**, respectively. The adaptive meshes shown in the rightmost column of **a,b,c** are constructed with the control parameter $\delta = 0.001, 0.3, 0.002$ respectively, achieving similar level of approximation error. The resulting meshes have cell counts of $133735, 177958, 130759$ respectively. With this downward refracting profile, the mesh sizes are still on the same level, with n^2-linear profile producing slightly more compact mesh than the other profiles.