Volume Path Tracer

Hang Ruan*
SJTU ACM 2021
Shanghai, China
zzrh01@sjtu.edu.cn

Yijing Guo*
SJTU ACM 2021
Shanghai, China
guoyijin@sjtu.edu.cn

Yuhao Wang*
SJTU ACM 2021
Shanghai, China
jerx2y@sjtu.edu.cn

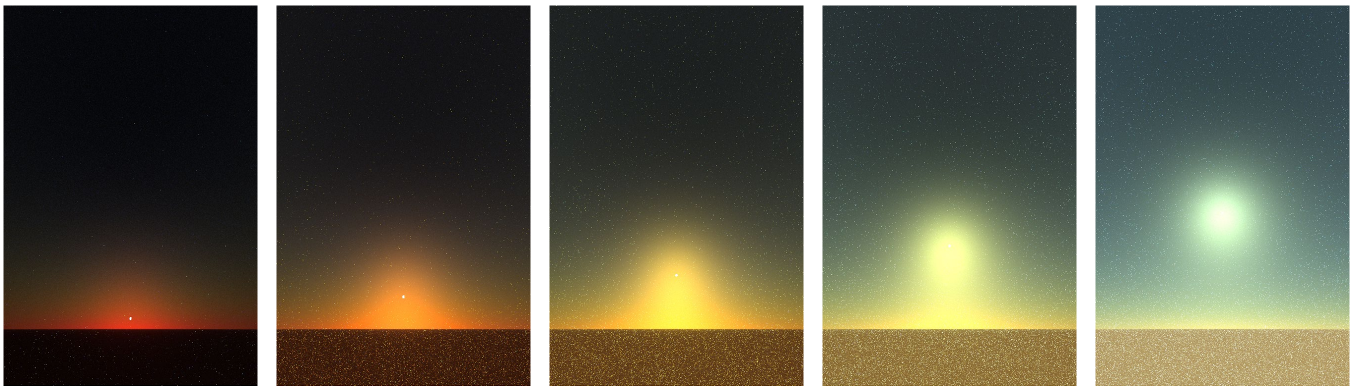


Figure 1: Sunrise with the atmosphere.

ABSTRACT

In this project, we will build a volumetric path tracer that can handle scattering and absorption inside participating media inside *lajolla*. We will split the development into 5 steps and build 5 volumetric path tracers. Each has more features than the previous ones.After complementing the renderer, we introduce the atmospheric medium to simulate the natural phenomenon.Then we render some scene files for the simulation of sunrise and generate a video for it.

ACM Reference Format:
Hang Ruan, Yijing Guo, and Yuhao Wang. 2024. Volume Path Tracer. In *Proceedings of ACM Conference (Computer Graphics).* ACM, New York, NY, USA, 5 pages. https://doi.org/

1 INTRODUCTION

This project explores the development of a volumetric path tracer designed to simulate light transport in participating media with variable density, such as smoke and fog. By extending traditional path tracing techniques, which typically focus on surface interactions, this volumetric path tracer incorporates the complexities of light scattering, absorption, and emission within the media itself. Our approach involves constructing a series of incrementally complex tracers, each capable of handling more sophisticated interactions than its predecessor.

The core of the paper is dedicated to the sequential development of the tracer, starting from a simple model that handles only absorption and systematically advancing through more intricate models that incorporate single and multiple scattering within both homogeneous and heterogeneous media. Each development section is accompanied by theoretical explanations, pseudocode, and results from rendered scenes.

In order to simulate natural atmosphere, we further introduce the atmospheric medium. It is a simplified model of the real atmosphere. Then we obtain a sequence of scene files for sunrise and render them to form an animation.

2 RELATED WORKS

Volumetric path tracing is a significant area in rendering technologies, primarily used to simulate light interaction within participating media like smoke, clouds, or fog. This method extends the traditional path tracing approach by considering the volume the light travels through, not just its interaction with surfaces.

One seminal work in this field is by Kajiya (1986)[3], who introduced the rendering equation that forms the foundation of light transport models used in computer graphics, including volumetric path tracing. Further, Jensen and Christensen (1998) expanded on this by integrating volumetric path tracing into practical rendering systems[2], showcasing its application in scenes with fog and smoke. Recent advancements have focused on optimizing volumetric path tracing for more efficient computations and better integration with real-time rendering engines. Novák et al. (2019) introduced an adaptive method[4] that significantly reduces the

*All authors contributed equally to this research.

Computer Graphics, June 2024, Shanghai Jiao Tong University, Shanghai, China

computational overhead by focusing computational resources on visually important features of the volume.

For handling heterogeneous media, researchers have explored multiple scattering and absorption models. Wenzel Jakob's implementation, referenced in our project, utilizes spectral varying densities to achieve realistic rendering of heterogeneous volumes. This approach aligns with techniques discussed by Georgiev et al. (2019), where the emphasis is on accurately simulating light transport within media that vary in composition and properties.

In our project, we adopt these foundational theories and contemporary advancements to build a volumetric path tracer capable of simulating both homogeneous and heterogeneous media. We iteratively enhance our renderer's capabilities, beginning with simple absorption and extending to complex scattering interactions, following a modular approach that facilitates understanding and integration of each component.

After completing the design of the renderer, when simulating real-world media, we base our work on Nishita et al.'s paper "Display of the Earth Taking into Account Atmospheric Scattering"[5] and obtained a simplified atmospheric model.

3 TECHNICAL APPROACH

3.1 Radiative Transfer Equation

Volumetric path tracing is an extension of the traditional path tracing technique used to render scenes involving not only solid objects but also participating media like smoke, fog, or clouds. These media are characterized by their interaction with light, including absorption, scattering, and emission of light within the volume itself. The complex interplay of light within these media necessitates a robust mathematical model to accurately simulate the phenomena.

The core of our volumetric path tracing method is based on solving the Radiative Transfer Equation (RTE), which describes the propagation of radiance through a medium with variable absorption and scattering properties. The RTE is expressed as:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}L(\mathbf{p}(t),\omega) = &-(\sigma_a(\mathbf{p}(t))+\sigma_s(\mathbf{p}(t)))L(\mathbf{p}(t),\omega)\\ &+L_e(\mathbf{p}(t),\omega)\\ &+\sigma_s(\mathbf{p}(t))\int_{S^2}\rho(\mathbf{p}(t),\omega,\omega')L(\mathbf{p}(t),\omega')\mathrm{d}\omega',\end{aligned} \tag{1}$$

where:

- $L(\mathbf{p}(t),\omega)$ is the radiance at position $\mathbf{p}(t)$ and direction ω,
- σ_a and σ_s are the absorption and scattering coefficients, respectively,
- L_e represents the emission of the medium,
- ρ denotes the phase function which describes the angular distribution of scattering.

3.2 Next Event Estimation

Next Event Estimation (NEE) is a critical technique in volumetric path tracing designed to efficiently handle direct lighting effects, especially in complex scenes involving participating media. NEE targets the enhancement of realism by ensuring that light sources are sampled explicitly, alongside the traditional random sampling methods associated with the phase function of the media.

In practice, NEE works by calculating the direct contribution from light sources at each point of scattering within the medium. For every scattering event, the algorithm decides whether to sample the light sources directly, based on their visibility and influence on the scene. If a light source is sampled, a shadow ray is cast from the scattering point to the light source to determine if the path is unobstructed:

$$L_s(\mathbf{p},\omega) = \frac{L_e(\mathbf{l})V(\mathbf{p},\mathbf{l})G(\mathbf{p},\mathbf{l})}{p_L(\mathbf{l})}, \tag{2}$$

where L_e is the emitted radiance from the light source, V is the visibility function (1 if visible, 0 otherwise), G is the geometric term that accounts for the distance and angle between the scattering point and the light source, and p_L is the probability of sampling the light source. This approach is particularly beneficial in media where light interactions are predominantly diffuse and indirect, making the capture of direct light essential for accurate rendering.

3.3 Multiple Importance Sampling

Multiple Importance Sampling (MIS) addresses the challenge of efficiently combining different sampling strategies to minimize variance in the Monte Carlo integration used in volumetric path tracing. MIS is essential when dealing with participating media, where the complexity of light interactions can lead to significant noise if not handled correctly.

MIS utilizes a weighted average of different estimators to optimize the sampling process. Each sampling strategy—whether sampling the phase function or direct light sampling—is considered based on its effectiveness under specific conditions. The weights for each sampling strategy are calculated using the balance heuristic, which is one of the most common methods for determining the weights in MIS:

$$w_i = \frac{n_i p_i^2}{\sum_j n_j p_j^2}, \tag{3}$$

where n_i and n_j are the numbers of samples, and p_i and p_j are the probabilities for the i-th and j-th sampling strategies, respectively. The final radiance estimation at any point in the medium is then a weighted sum of the estimates from each strategy:

$$L(\mathbf{p},\omega) = \frac{\sum_i w_i L_i(\mathbf{p},\omega)}{\sum_i w_i}, \tag{4}$$

This formulation ensures that each sampling method contributes optimally, based on its probability of providing a useful estimation towards the final image. By employing MIS, volumetric path tracing algorithms can effectively balance the contributions from different light paths, reducing variance and improving the convergence rate of the rendering process.

3.4 Medium Design

In our project, considering the convenience of modeling and the authenticity of the scenario, we have designed several participating media. The specific content is shown as follows.

Homogeneous and Heterogeneous The two media are basic design for our project. Homogeneous medium has fixed σ_a and σ_s

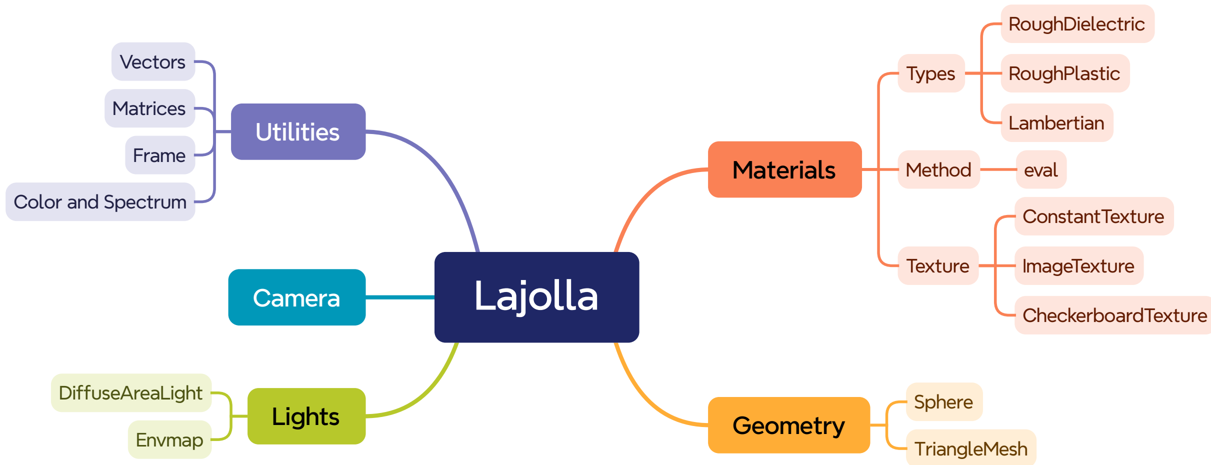


Figure 2: Framework of *Lajolla*

and heterogeneous medium has a density function for the coefficients.

Atmospheric In order to simulate real-life scenarios, we consider introducing a medium model to simulate the Earth's atmosphere. Inspired by Nishita et al's paper[5], we design a relatively simple atmosphere medium model. Our model is based on the reference of this github.

The atmosphere is natural gas enveloping the Earth, which can be conceptualized as a sphere of participating media. It indeed consists of various layers with different densities that decreases exponentially. We simplify the model and assume it has only one layer. Then the atmosphere can be regarded as a kind of specific heterogeneous medium with the density decreasing exponentially with altitude. The density function is as follows:

$$d(h) = d(0) * e^{-\frac{h}{H}}$$

where h is the altitude and H is called the scale height, denoting the thickness of atmosphere if it is even on the surface of the Earth.

The atmosphere consists of air molecules such as oxygen, nitrogen and argon, together with water vapor and droplets. These two important components undergo special types of scattering. Air molecules undergo Rayleigh scattering, which is the main reason why the sky appears blue; the water vapor and droplets are responsible for Mie scttering, which results in the while halo we see around light sources, such as the sun. In our atmospheric model, we mix water and air molecules in certain proportions(here we choose 50

Rayleigh scattering. . This is responsible for air molecules. The scattering coefficient σ_s of Rayleigh is:

$$\sigma_s(h) = \frac{8\pi^3(n^2-1)^2}{3N\lambda^4} * \rho(h)$$

where $n = 1.00029$ is the index of refraction of air, $N = 2.504 \times 10^{25}$ is the molecular number density of air and λ is the wavelength of light. $H = 8.5km$ for air molecules.

The phase function for Rayleigh P_r is as follows:

$$P_r(\theta) = \frac{3}{4} * (1 + \cos^2(\theta))$$

where θ means the angle between the incomming ray and the direction of light.

Mie scattering. . This is responsible for water.

The scattering coefficient σ_s of Mie is:

$$\sigma_s(s) = \sigma_s(0) * e^{-\frac{h}{H}}$$

where $H = 1.2km$ for water.

In addition, to fit the measurement, we use $\sigma_s/\sigma_e = 0.9$ for water vapor's absorption of light.

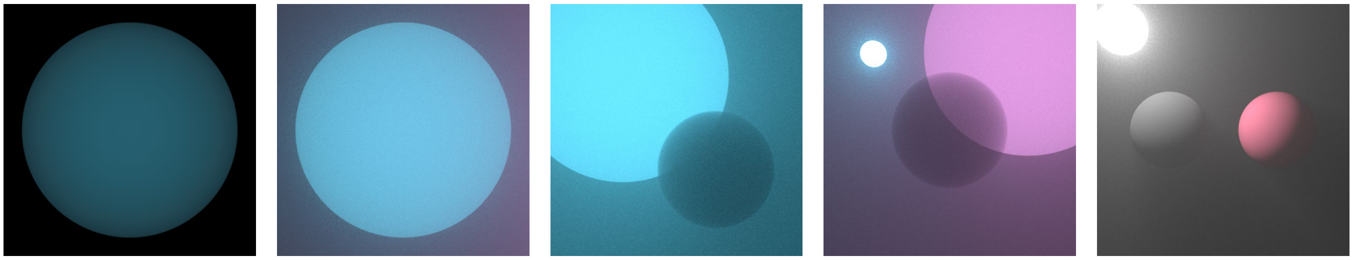
The actual Mie scattering is difficult to simulate and we use the Cornette-Shanks phase function to approximate Mie as some papers[1]. The phase function P_m is as follows:

$$P_m(g,\theta) = \frac{3}{2} * \frac{(1-g^2)(1+\cos^2(\theta))}{(2+g^2)(1+g^2-2g\cos(\theta))^{\frac{3}{2}}}$$

where g is the term to determine forward or backward scattering. We choose $g = 0.76$ as this paper[1].

3.5 Animation

The scene file used for animation generation is from this github repo. We render all the scene files in order and produce .exr images. Finally we use ffmpeg to put the image sequence together and form a video.



Step 1 Step 2 Step 3 Step 4 Step 5

Figure 3: Output of each step

4 RESULTS

We used the *lajolla* rendering framework for our project, which uniquely employs `std::variant` to manage polymorphic behaviors, differing significantly from traditional rendering systems. This design choice not only improves type safety but also optimizes performance by aligning with modern C++ practices. Operations within *lajolla* are encapsulated in operator structs, implemented as functors by overloading the parenthesis operator. This modular approach not only enhances the framework's flexibility and maintainability but is also better for the framework's transfer to the GPU framework. Figure 2 illustrates the basic framework of *lajolla*.

We split the development into 5 steps and build 5 volumetric path tracers. Each has more features than the previous ones.

4.1 Single monochromatic absorption

In this experiment, we implement a renderer capable of rendering a scene that includes a single, homogeneous, and purely light-absorbing volume medium. In this experiment, the scattering coefficient σ_s is set to 0 , indicating no light scattering occurs.

Under such conditions, the volumetric rendering equation can be simplified to

$$\frac{d}{dt}L(\vec{p}(t),\omega) = -\sigma_a L(\vec{p}(t),\omega).$$

Through derivation, we obtain the formula:

$$L(\vec{p}(t),\omega) = L_0 \exp\left(-\sigma_a t\right)$$

Let's see our basic volume path tracing Algorithm:

- `Step 1`: Sampling the primary ray, starting from the camera.
- `Step 2`: Detect the intersection of this ray with objects in the scene and calculate the transmittance and pdf of the ray.
- `Step 3`: Determine whether the ray touches an object's surface. And when the light has already scattered before it touches an object's surface, handle the scattering (which does not occur in task 1, but will occur in subsequent tasks).
- `Step 4`: If the ray `r` touches the object's surface and the surface is the light source.

  return $transmittance * Le$.

The experimental results are illustrated through rendered figur 3 step 1 that exhibit the visual outcomes of our rendering technique, effectively conveying the level of detail and realism attained.

4.2 Only a single scattering event

In Experiment 2, we assume that light undergoes only a single scattering event within the medium, with all other conditions remaining the same as in Experiment 1.

When it is assumed that light undergoes only a single scattering event within the medium, a simplified equation can be used to approximate the calculation of scattered light.

$$L_{\text{scatter1}}(\mathbf{p},\omega) = \int_{\mathcal{M}} \rho\left(\mathbf{p}(t),\omega,\omega'\right) L_e\left(\mathbf{p}',\omega'\right) \exp\left(-\sigma_t \left\|\mathbf{p}(t)-\mathbf{p}'\right\|\right) \cdot \frac{\left|\omega'\cdot\mathbf{n}_{\mathbf{p}'}\right|}{\left\|\mathbf{p}(t)-\mathbf{p}'\right\|^2} \text{visible}\left(\mathbf{p}(t),\mathbf{p}'\right) \mathrm{d}\mathbf{p}'$$

The radiative transfer equation is then

$$\frac{\mathrm{d}}{\mathrm{d}t}L_2(\mathbf{p}(t),\omega) = -\sigma_t L_2(\mathbf{p}(t),\omega) + \sigma_s L_{\text{scatter1}}\left(\mathbf{p},\omega\right).$$

We employ importance sampling on the transmittance function. The probability density function (PDF) for sampling time t is derived from the transmittance:

$$p(t) = \sigma_t \exp\left(-\sigma_t t\right),$$

where σ_t is the extinction coefficient. By integrating the transmittance, we obtain the cumulative distribution function (CDF) and its inverse:

$$T_{\exp}^{-1}(t) = -\exp\left(-\sigma_t \cdot t\right) + 1.$$

The inverse CDF allows us to sample t from a uniform distribution u as follows:

$$t = \frac{\log(1-u)}{-\sigma_t}.$$

This method ensures that the sampling process is aligned with the physical properties of light scattering within the medium.

The experimental results are illustrated through rendered figure 3 step 2 that exhibit the visual outcomes of our rendering technique, effectively conveying the level of detail and realism attained.

4.3 Multiple scattering between multiple volumes

In Experiment 3, we consider multiple scattering between multiple volumes, with all other conditions remaining the same as in Experiment 2.

Under this assumption, our volumetric integral becomes:

$$L_3(\mathbf{p}(0), \omega) = \int_0^{t_{\text{hit}}} \exp\left(-\sigma_t t\right) \sigma_s L_{\text{scatter}}(\mathbf{p}, \omega) dt + \exp\left(-\sigma_t t_{\text{hit}}\right) L_e(\mathbf{p}(t_{\text{hit}}))$$

When multiple scattering events can occur, we must also account for variations in the medium and integrate the Russian Roulette algorithm to determine when to terminate the scattering process.

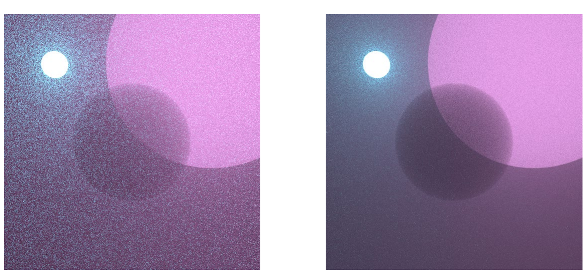
The experimental results are illustrated through rendered figure 3 step 3 that exhibit the visual outcomes of our rendering technique, effectively conveying the level of detail and realism attained.

4.4 Adding NEE and MIS to the path tracer

In this experiment period, we need to add the NEE and MIS into the path tracer. The introduction of these two method is listed in the methodology part. In volume path tracing methodologies, the computational approach typically involves separately quantifying the direct radiance emanating from light sources and the indirect radiance originating from other surfaces. Thus, the equation can be expressed as follows:

Total Radiance = Direct Radiance (from light sources) + Indirect Radiance

If we sample the rays from light sources independently and assess them using a probability density function, better outcomes can be achieved with a lower number of samples.



Rendered without NEE Rendered with NEE

Figure 4: Output of with and without NEE, 256 sample pixels.

Therefore, we conducted an ablation study: comparing the path tracer models before and after the integration of NEE. The rendered images, with a sampling count of 256, are shown below. It can be observed that the model without NEE integration exhibits significantly more noticeable noise in the rendered images.

4.5 Multiple chromatic heterogeneous volumes

When dealing with multiple chromatic and heterogeneous volumes, here is a super clever idea called null-scattering to solve the two problems at once. It is a special model of light transport that adjusts the density of the medium to maintain radiance equilibrium while allowing the use of analytical methods to simplify the computation of light transmission within the medium.

Here is a trick that we convert the heterogeneous medium into a homogeneous medium by inserting fake particles that do not scatter lights. We can then derive a solution by applying the analytical solution from the homogeneous medium. Now we introduce σ_n to denote the density of the fake particles and add $-\sigma_n L + \sigma_n L$ to the radiative transfer equation:

$$\begin{aligned}\frac{d}{dt}L(\mathbf{p}(t), \omega) = &-\left(\sigma_t(p(t)) + \sigma_n(\mathbf{p}(t))L(\mathbf{p}(t), \omega)\right) \\ &+ \sigma_n(\mathbf{p}(t))L(\mathbf{p}(t), \omega) \\ &+ \sigma_s(\mathbf{p}(t)) \int_{S^2} \rho(\omega, \omega')L(p(t), \omega')d\omega'\end{aligned}$$

Now if we can make sure that $\sigma_t + \sigma_n$ is a constant for all t, we successfully convert the radiative transfer equation back to a homogeneous medium.

A common choice for the constant is the upper bound of $(\mathbf{p}(t))$ so that $(\mathbf{p}(t)) \geq 0$. Then we can apply Monte Carlo sampling to it.

As for multiple chromatic media, we need to deal with RGB colors. This can be solved by sampling a channel (R, G and B) equiprobably whenever we need to sample distance.

This part is the basis of our atmospheric medium and animation.

4.6 Animation with atmospheric medium

Finally we render a sequence of scene files and generate a video. This can be found at our github.

REFERENCES

[1] Eric Bruneton and Fabrice Neyret. 2008. Precomputed atmospheric scattering. In *Computer graphics forum*. 1079–1086.

[2] Henrik Wann Jensen and Per H Christensen. 1998. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 311–320.

[3] James T Kajiya. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150.

[4] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural importance sampling. *ACM Transactions on Graphics (ToG)* 38, 5 (2019), 1–19.

[5] Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, and Eihachiro Nakamae. 1993. Display of the earth taking into account atmospheric scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 175–182.

A TEAM MEMBER CONTRIBUTIONS

All authors contributed equally to this project.

Hang Ruan(33%), Contributions: Volume Path Tracer:task1,task2,task3

Yuhao Wang(33%), Contributions: Volume Path Tracer:task4,task5

Yijing Guo(33%), Contributions: simulation of sunrise and generate a video