Project: Basic Volume Path Tracing

Zezhong Pan

1 Introduction

In this project, I implemented a basic volumetric path tracer using Nori framework. So far I have implemented simple path tracer and path tracer with multiple importance sampling on Nori in previous assignments. These path tracers only consider surface interactions between a mesh and light, and assume the space is completely vacuum. However in real life, the air where light travels can be full of small particles, such as smoke, fog, and dust. When light enters these participating medium, it will get scattered and absorbed by the particles. This can lead to some different visual effect such as the well-known Tyndall effect. The goal of this project is to extend the path tracer to support this kind of effect.

2 Algorithms

2.1 Radiative Transfer Equation (RTE)

Radiative Transfer Equation is a mathematical model that describes how light interacts with participating media, and is the foundation for volumetric path tracing.

$$(\omega \cdot \nabla)L(x,\omega) = \sigma_s(x)\int_{S^2} f_p(x,\omega_i \to \omega)L(x,\omega_i)d\omega_i - \sigma_t(x)L(x,\omega) + Q(x,\omega)$$

For homogeneous medium, scattering coefficient ω_s and extinct coefficient ω_t are constant everywhere inside. The term $_p$ is the phase function which describes the in-scattering of light from different direction.

To use RTE for Monte Carlo integration, we need to rewrite it as an integral equation. For simplification, let's assume the whole scene is inside the medium, so that we don't need to consider the situation where light crosses the border of media. We also set $Q = 0$, which means the media does not emit light.

$$L(x,\omega) = \int_0^\infty T(r \leftrightarrow x)\sigma_s \int_{S^2} f_p(x,\omega_i \to \omega)L(x,\omega_i)d\omega_i d\tau$$

For homogeneous medium, $T(r \leftrightarrow x) = \exp(-||r - x||\sigma_t)$

Now we have the integral form for Monte Carlo Integration. To do the integration, there are 2 variables we need to sample, free distance τ and direction ω_i.

2.2 Free Distance Sampling

Free distance τ following is drawn from

$$p(\tau) = \lambda(\tau) \exp(-\int_0^\tau \lambda(t)dt)$$

where $\lambda(\tau) = \sigma_t(x - \tau\omega)$ when the sampled distance still lies inside the media.
Since we are using homogeneous media, we can let $\lambda(\tau) = \sigma_t$

$$p(\tau) = \sigma_t \exp(-\sigma_t \tau)$$

Then τ can be drawn by $\tau = -\log \xi / \sigma_t$, where $\xi \sim U(0, 1)$.
Let $L(x, \omega) = \int_0^\infty f(\tau)d\tau$, we have

$$\langle L(x, \omega) \rangle = \frac{f(\tau)}{p(\tau)} = \frac{\sigma_s}{\sigma_t} \int_{S^2} f_p(x, \omega_i \to \omega) L(x, \omega_i) d\omega_i$$

2.3 Direction Sampling

The scattering direction ω_i is sampled based on phase function f_p. In this project, I use the Henyey-Greenstein phase function.

$$f_p(\theta|g) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{1.5}}$$

This function can be sampled with inverse CDF technique:

$$\mu = \frac{1}{2g}(1 + g^2 - (\frac{1 - g^2}{1 - g + 2g\xi_1})^2)$$
$$\phi = 2\pi\xi_2$$

where $\mu = \cos\theta$, and $\xi_1, \xi_2 \sim U(0, 1)$.
Then the new direction can be written as $(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$.
And now we have

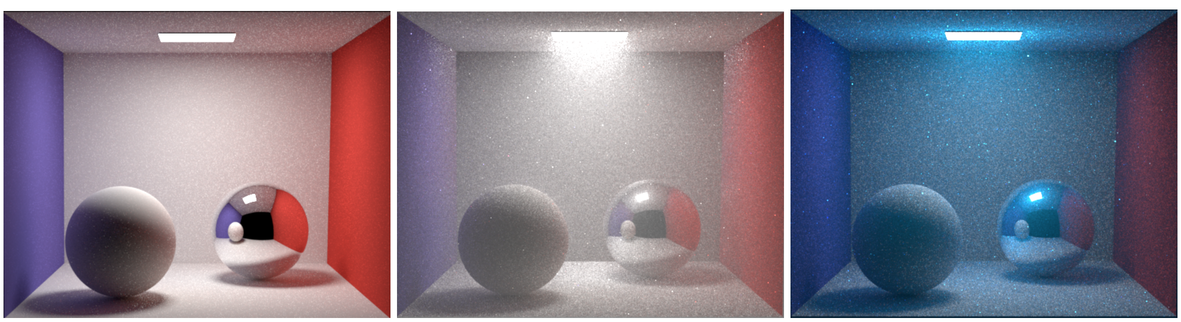
$$\langle L(x, \omega) \rangle = \frac{\sigma_s}{\sigma_t} L(r, \omega_i)$$

2.4 Next Event Estimation

After implementing the simple version of volume path tracer, I noticed that the results are noisy. This is because the scattering light path bounces in random direction and does not guarantee to hit a light source. So I decided to add a simple estimation for direct illumination. Since we made the assumption that the whole environment is inside the medium and so is the light source, we can randomly sample a light source and calculate the radiance with the attenuation function.

```
Radiance(x, w)
    draw t
    if t does not hit a mesh:
        # Medium interaction
        r = x - t*w
        L_d = directIllumination(r,w)
        draw w_i
        L_i = Radiance(r, w_i)
        albedo = sigma_s / sigma_t
        return albedo * (L_i + L_d)
    else:
        # Surface interaction
        return SurfaceRadiance(mesh.o, w)
```

3 Results



We use the CBOX scene to test the path tracer. The first image is rendered without any media using simple path tracer as a reference. The second image is rendered by volume path tracer, setting $g = 0.9$. We can see very obvious scattering effect, especially near the light source. In the third image, I differ the RGB values of the scattering coefficient σ_s to favour blue color. We can see that red color gets absorbed and the whole scene looks like underwater.

4 Conclusion

In this project, I implemented a simple volume path tracer using Nori framework. Right now it only supports cases when whole scene is inside a big volume of homogeneous media without emission. I tried to extend the volume path tracer to handle objects made of homogeneous media, but dealing with boundaries is more complicated than I thought. By the time I wrote this report, I had not managed to produce an integrator that works. Future improvements may also involve extending to heterogeneous media and introducing multiple importance sampling.

5 Reference

- COMPSCI 211C Lecture slides.
- Nori: `https://wjakob.github.io/nori/`
- Jianing Zhang. On Sampling of Scattering Phase Functions. 2019. arXiv:1812.00799v2

6 Appendix

- basicVolumetricPathTracer.cpp: the implementation of the integrator
- volume.h, homoVolume.cpp: sampling free distance and new direction for a volume, for homogeneous volume σ_t, σ_s, g are read as parameters