Advanced Computer Graphics Final Report

JINGYI LYU, 2022010874, CS23, IIIS

1 INTRODUCTION

For the final project of advanced computer graphics course this semester, my team have chosen project topic `Image Rendering` and implemented a basic physically-based renderer which runs on software pipeline. My teammate is Shengquan Du from CS21 of IIIS.

We have accomplished a bunch of functions and materials, following the instruction of project topic provided. Precisely, we have implemented:

- A path tracing algorithm handling diffusive (Lambertian) and specular material.
- Acceleration structure based on BVH.
- Transmissive material.
- Customized material based on principled BSDF.
- Color texture and normal texture.
- Anti-aliasing.
- Importance sampling and multiple importance sampling.
- Volumetric rendering effects, including homogeneous and inhomogeneous volume rendering, channel independent subsurface scattering, volumetric emission and volumetric alpha shadow.
- Special visual effects: Motion blur and depth of field based on trivial aperture.
- A self-made complicated scene.

Our implementation did not start from scratch, but developed from the `sparkium` framework provided by TAs. For technical issues, we mainly refer to the web version of *Physically Based Rendering: From Theory To Implementation* [5] and the corresponding codebase `pbrt-v3` [2]. Our own codebase can be found at https://github.com/qwaszx-D/sparkium.

Results of those finished tasks is presented in detail below.

2 BASIC FUNCTIONS

A typical path tracing algorithm handling diffusive and specular material was implemented first. We applied it to the popular test scene `Cornell Box`, with result shown in 1 and 2.

Contribution: Jingyi Lyu.

3 ADVANCED MATERIALS

3.1 Transmissive Material

Specular transmissive material was implemented and resulted in 3. The bottom part of block is dim because the floor under block cannot receive direct illumination (since we did not implemented alpha shadow for this case). The "block" actually does not have bottom face, which maybe also a reason.

Contribution: Shengquan Du.

Author's address: Jingyi Lyu, lv-jy22@mails.tsinghua.edu.cn, 2022010874, CS23, IIIS.

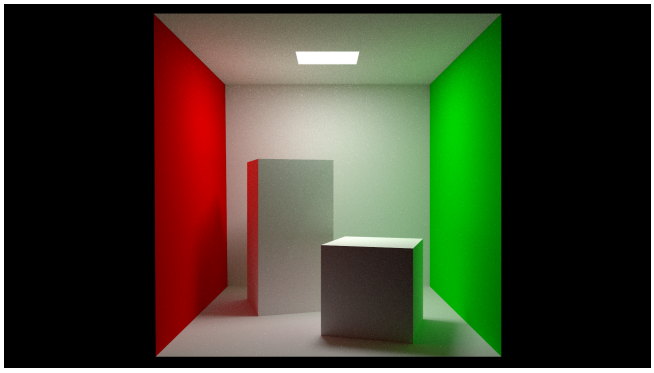

Fig. 1. Cornell Box

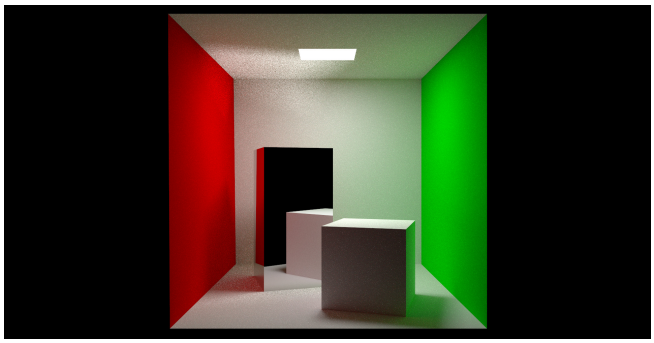

Fig. 2. Cornell Box with specular material

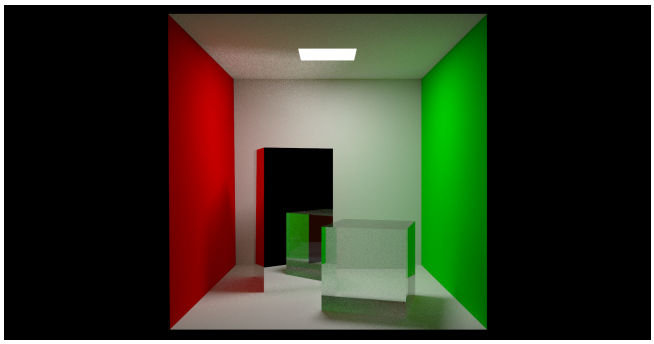

Fig. 3. Cornell Box with transmissive material

3.2 Principled BSDF

Complicated customized material based on Disney Principled BSDF was implemented. A fancy result is shown in 4, where the sea surface is made of principled BSDF with parameter Albedo = [0.064 0.130 0.292], Eta = 1.33, Metallic = 0.8, Roughness = 0.1, Alpha = 0.925, Transmission = 1.

Contribution: Shengquan Du.

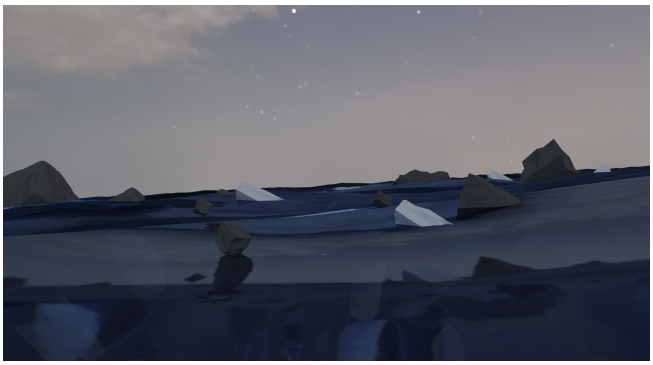

Fig. 4. Cornell Box with transmissive material

3.3 Volumetric Rendering

We have implemented volumetric rendering effects based on volumetric light transport and bidirectional subsurface scattering reflection distribution function (BSSRDF) respectively, whitch has covered most requirements of this course project.

Volumetric light transport is similar to normal path tracing, but light has probability interacting with participating medium which proportional to attenuation coefficient of medium. Upon interaction, an incident direction is sampled according to phase function, which defines the angular distribution of scattered radiation at a point. Based on travelling distance and attenuation strength, the proportion of reduce in light (as well as shadow ray, when estimating direct illumination) strength can be computed.

BSSRDF is a fast approximation to true volumetric scattering. It is usually presented in form of $S(p_o, w_o, p_i, w_i)$, which is extened from *BSDF* by adding degree of freedom about incident position p_i. The property of BSSRDF materials are determined by parameters including albedo, mean free path length and refractive index.

One can refer to Chapter 11 and Chapter 15 of [5] for detail.
Contribution: Jingyi Lyu.

3.3.1 Volumetric Light And Fog (Volumetric Light Transport). As typical volumetric rendering effect, we have implemented homogeneous volumetric light and volumetric fog, shown in 5 and 6.

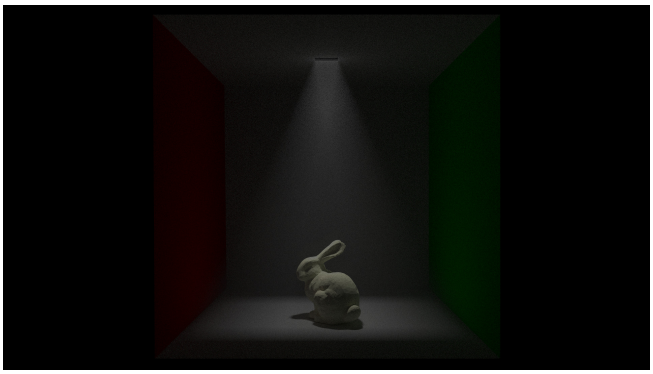

Fig. 5. Volumetric light in dim space

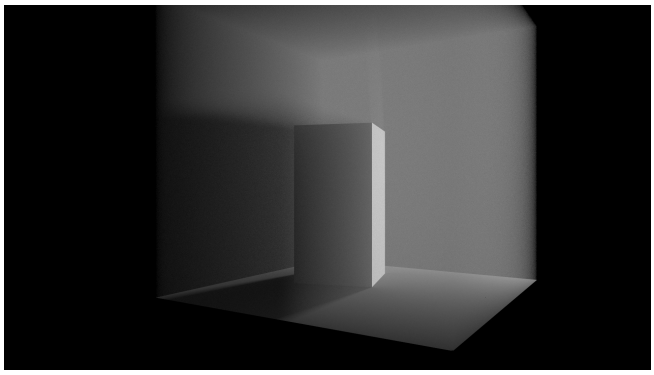

Fig. 6. Heavy fog illuminated by strong light source

3.3.2 Inhomogeneous Volumetric Medium (Volumetric Light Transport). We provided support for inhomogeneous medium whose density varies throught the space. Density grid is required to express those medium, which is stored in external file and read at each loading. To sample medium with varying density correctly, we used delta tracking. The result is shown in 7.

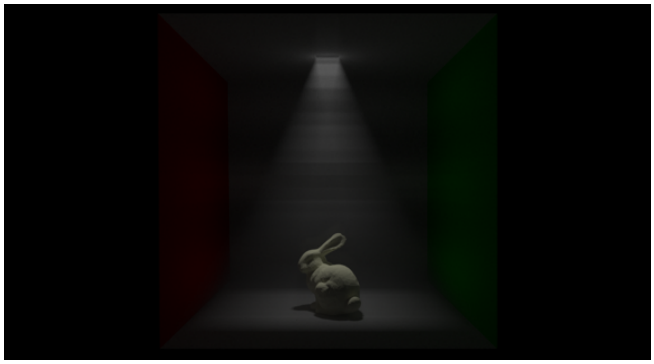

Fig. 7. Volumetric light in inhomogeneous medium

3.3.3 Volumetric Emission (Volumetric Light Transport). Volumetric emission effect is shown in 8. We did not provide importance sampling support for this.

3.3.4 Volumetric Alpha Shadow (Volumetric Light Transport). Attenuation of shadow ray elicits volumetric alpha shadow. Based on different estimation mechanism of attenuation in homogeneous and inhomogeneous case, our alpha shadow also supports homogeneous and inhomogeneous case, which are shown in 9 and 10. In the latter scene, the fog on left side is slightly denser than right side, so as the shadow. Some patterned texture appears on the fog due to pseudo random feature or floating point error during grid generation and scattering sampling.

3.3.5 Channel Independent Subsurface Scattering (BSSRDF). We have applied BSSRDF material to the `Cornell Box` scene, resulting in the blue translucent box in 11.

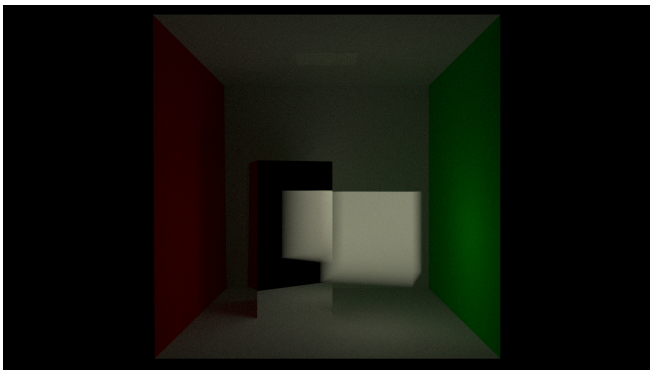

Fig. 8. Volumetric emission

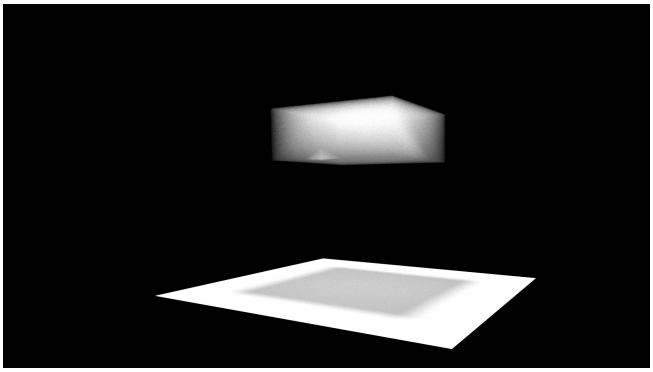

Fig. 9. Homogeneous volumetric alpha shadow

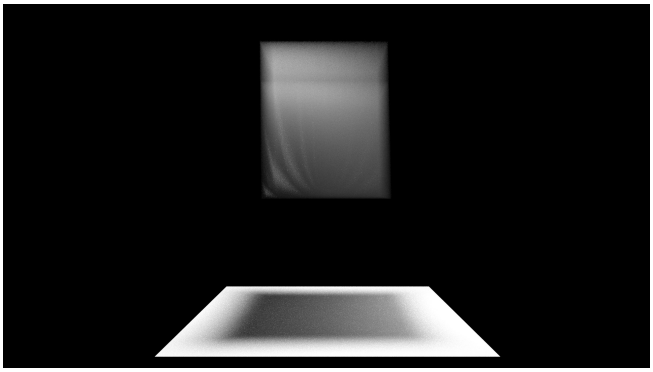

Fig. 10. Inhomogeneous volumetric alpha shadow

4 FUNCTIONAL FEATURES

4.1 Texture Mapping

We have implemented texture mapping and normal texture. 12 shows result of trivial color texture, while the fluctuation of sea surface (especially the reflection phenomenon) in 4 was acheived by normal texture. The corresponding raw texture is shown in 13, which is generated by Perlin noise algorithm [6].

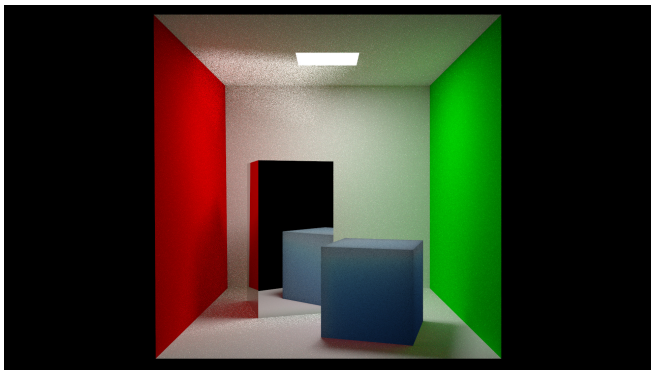

Fig. 11. BSSRDF material

Contribution: Shengquan Du; Noise generation done by Jingyi Lyu.

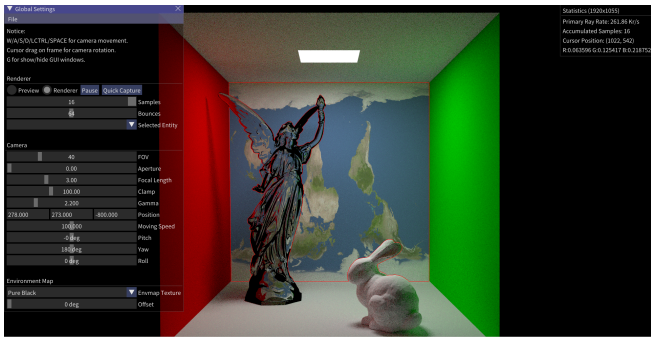

Fig. 12. Trivial color texture

4.2 Acceleration Structure

We have implemented acceleration of intersection computation with BVH. Scene in 12 has thousands of triangles while the primary ray rate rendering it is still about $\frac{1}{3}$ of rendering 1.
Contribution: Shengquan Du.

4.3 Importance Sampling

Importance sampling and multiple importance sampling are applied to enhance path tracing convergence. The following figure 14 contains a typical setting for multiple importance sampling: Planes of different roughness reflecting light sources with different extent.
Contribution: Jingyi Lyu.

4.4 Anti-Aliasing

Basic anti-aliasing by casting multiple rays through one pixel had been implemented by `sparkium` framework, we just applied it throughout our rendering process.
Contribution: None.

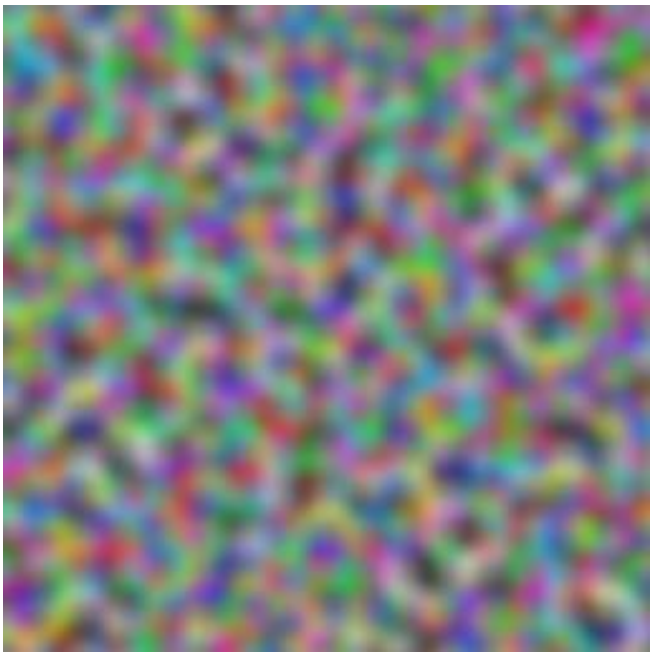

Fig. 13. Raw perlin noise for normal texture

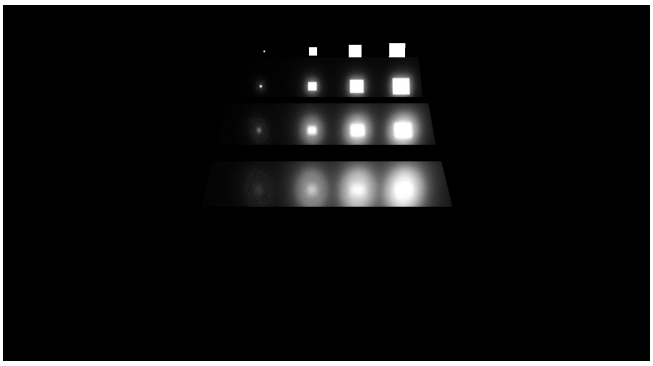

Fig. 14. Application of multiple importance sampling

5 SPECIAL VISUIAL EFFECTS

5.1 Depth Of Field

Trivial aperture is also inherit function of camera in `sparkium`. We applied it and obtained the following result in 15.
Contribution: None.

5.2 Motion Blur

We have implemented motion blur effect and obtained the rotating Earth in 16 as result.
Contribution: Shengquan Du.

6 OUR OWN SCENE

We collected 3D models from [4], [1], [3] and used Blender to manipulate and convert these models to meet the requirement of our framework and scene design. Then we applied various materials to those models, including principled BSDF (the character, sea and the throne), normal texture mapping (the see surface), volumetric fog (the beam and its scattering effect) and color texture (of the sky). The final result shown in 17 is an imitation of a CG appeared in Genshin Impact. 1920*1080 pixels and 256 samples per pixel are computed, costing 6 hours.
Contribution: Shengquan Du, Jingyi Lyu.

Fig. 15. Depth of field based on trivial aperture

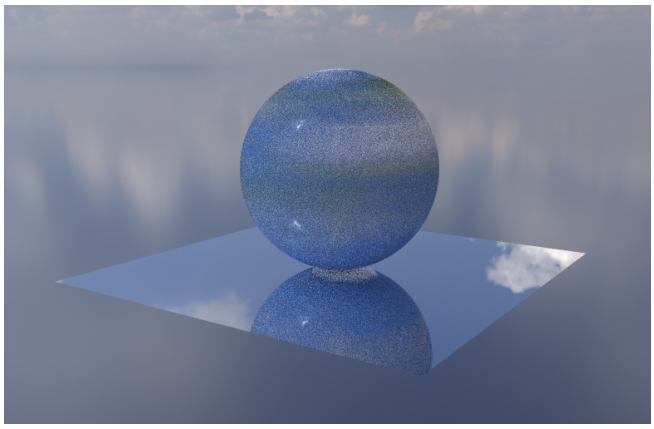

Fig. 16. Motion blur: Earth rotating fast

7 DISCUSSION

We have noticed noisy light spot occurring sparsely during rendering process, which burdened convergence performance a lot since their strength are very large. We inferred that this could be caused by sampling incident direction near the top or edge of hemisphere. In this case, the sampling probability density could be very small, so the resulting incident illumination will be scaled tremendously. To tackle with this, we clamped all the probability density on denominator to $[p_l, +\infty)$, while all strength returned each recuresion to $[0, l_u]$, where p_l, l_u are two parameters to adjust manually.

Fig. 17. Our final scene: Crisis of Fontaine prophecy

Apart from that, though the current render result is satisfying, we observed that behaviour of BSSRDF material is unstable sometimes, and its direct illumination may have some bugs. Since sampling and evaluation of BSSRDF are very complicated, we still need time to check them in the future.

REFERENCES

[1] Christiana. 2023. [Genshin Impact] Prophecy of Fontaine. https://www.aplaybox.com/details/model/9dRqWivE3QNT.
[2] Pharr et al. 2023. pbrt-v3. https://github.com/mmp/pbrt-v3.
[3] miHoYo and Guanhai. 2023. [Genshin Impact] Furina. https://www.aplaybox.com/details/model/iBOW5aAVDaoH.
[4] Muyuhe. 2023. The prophecy of Fontaine and the water god. https://www.aplaybox.com/details/model/JS242L59Zx6Q.
[5] Pharr, Jakob, and Humphreys. 2021. Physically Based Rendering: From Theory To Implimentation. https://www.pbr-book.org/3ed-2018/contents.
[6] Wikipedia. 2023. Perlin noise. https://en.wikipedia.org/wiki/Perlin_noise.