ACG Project Midterm Report

Zhicheng Tang
tangzc24@mails.tsinghua.edu.cn

Abstract

I choose the **Rendering** topic.

1 Basic Information

In this project, I aim to build upon the provided ShortMarch rendering framework by implementing a physically based rendering system with support for advanced graphics functionalities. I have already established core ray tracing methods, including handling of transmissive materials, point and area light sources, and basic anti-aliasing. Moving forward, I plan to extend the renderer with color texture and normal mapping, skybox environment lighting, and volumetric rendering with mipmap support for efficient texture sampling. My goal is to create a fully functional, extensible rendering system capable of synthesizing visually complex and physically plausible scenes.

2 Technical Points Detail

The project has implemented several key rendering techniques within the framework:

- **Basic:** I have implemented a basic ray tracing pipeline with **Russian Roulette** termination. The system supports materials with both diffuse and specular components, and can render objects of various geometries loaded from mesh files. (See Figure 1)
- **Acceleration Structure:** The spatial acceleration structure is provided and utilized from the underlying ShortMarch framework.
- **Light Sources:** The renderer supports both point lights and area lights (Figures 2 and 3). Area lights produce **alpha shadows** due to their spatial extent. For computation light intensity from area lights, **importance sampling** is used by sampling several points from the area and calculate their direct lightings.
- **Transmissive material:** The implementation supports ray tracing for transmissive materials. For each intersection, the algorithm calculates the probability of reflection and refraction based on material properties, then stochastically samples the corresponding ray path. (See Figure 5)
- **Anti-aliasing:** To reduce aliasing artifacts, a basic supersampling approach is implemented. Instead of simply choosing the center of each pixel, I randomly sample a point within the pixel area during each rendering phase. This effectively performs Monte Carlo integration to smooth out edges.

Below are some other methods that I plan to implement during the next step.

- **Texture Mapping:** I will first implement texture support. This includes color textures for albedo mapping, normal maps for surfaces, and a skybox for environment lighting. Additionally, I plan to implement an adaptive mipmap algorithm to efficiently handle texture filtering.
- **Material Model:** A Principled BSDF material model will be developed to achieve realistic appearances for various materials.
- **Volumetric Rendering:** I will then focus on volumetric effects. This includes homogeneous volume rendering for effects like fog or smoke, and may extend to support key subsurface scattering phenomena for simulating translucent materials such as skin, wax, or marble.

Conference'17, Washington, DC, USA
2025. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 Detailed Schedule

- **Day 1 and 2:** Implemented and debugged the ray tracing algorithm for specular and diffuse objects based on the provided framework by completing the ClosestHit function.
- **Day 3:** Extended the lighting system from a single hard-coded point light to support both point lights and area lights. Additionally, implemented the Russian Roulette algorithm.
- **Day 4:** Added support for transmissive materials and implemented an anti-aliasing method via supersampling to improve image quality.

4 Rendering Results

Here are several figures for the current results of my project:

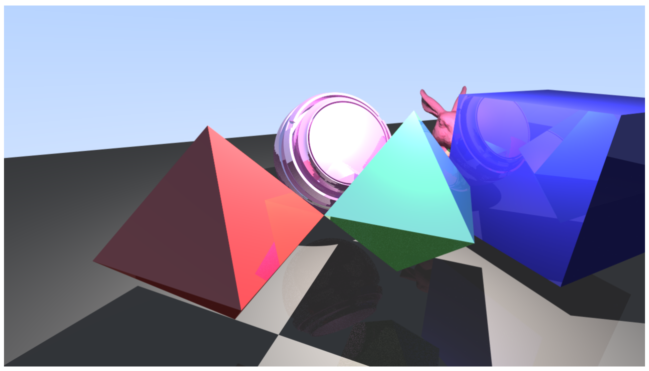

Figure 1: Basic materials, light and shadow

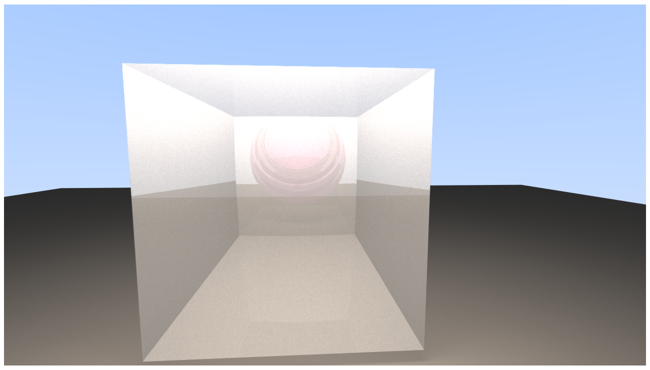

Figure 5: Transmissive material

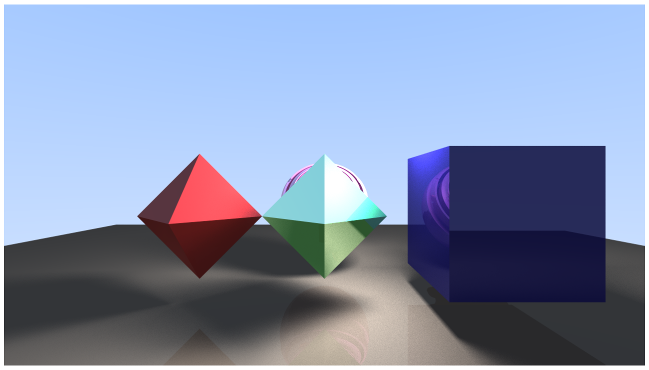

Figure 3: Area Light

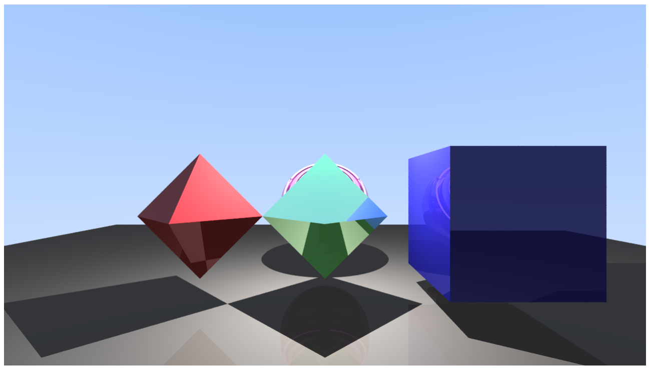

Figure 2: Point light

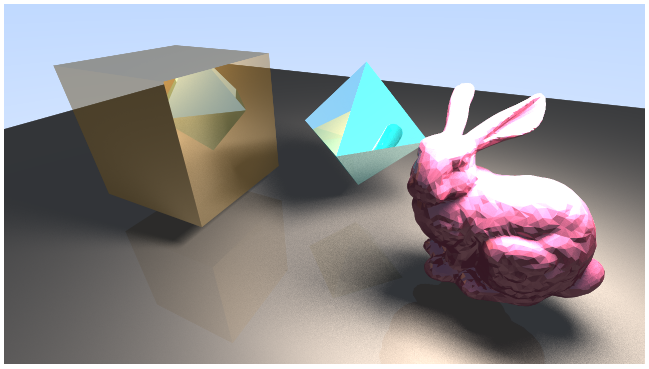

Figure 4: Metaillic reflection

5 Acknowledgement

The author thanks the teacher assistants for providing the Short-March framework.