Assignment 3:
Basic Ray Tracing

NAME: GAOYE
STUDENT NUMBER: 2022533005
EMAIL: GAOYE2022@SHANGHAITECH.EDU.CN

1 INTRODUCTION

The report is about my CG assignment3: Basic Ray Tracing. The following are the functions I managed to achieve.

I can compile the source code and configure the language server environment.
I implemented ray-triangle intersection functionality.
I implemented ray-AABB intersection functionality.
I implemented the BVH construction.
I implemented the IntersectionTestIntegrator and PerfectRefraction material for basic ray tracing validation, handing refractive and solid surface interactions.
I implemented a direct lighting function with diffuse BRDF and shadow testing.
I implemented anti-aliasing via multi-ray sampling per pixel within a sub-pixel aperture.

2 IMPLEMENTATION DETAILS

2.1 Language server etc.

I chose Visual Studio 2022 as the compiler, and vscode as the the workspace. Considering that Visual Studio can not generate compile_command.json for clangd, I also used Ninja to generate compile_command.json.

2.2 Ray-triangle intersection

The ray-triangle intersection is implemented using the Möller-Trumbore algorithm, which efficiently computes the intersection point using barycentric coordinates.

2.3 Ray-AABB intersection

The AABB intersection uses the slab method, which computes entry and exit points along each axis and finds their overlap.

2.4 BVH

The BVH construction implements a spatial partitioning structure using median splitting heuristic for balanced tree construction.

2.5 Integrator and Perfect Refraction material

The refraction part implements total internal reflection handling with fallback to reflection.

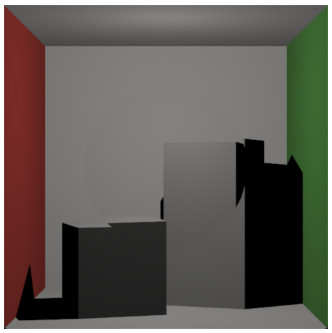

Fig. 1. Final result

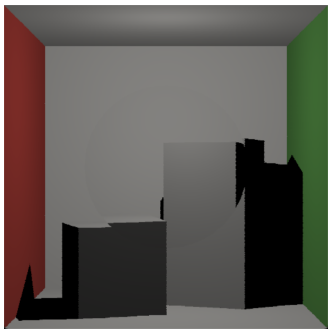

Fig. 2. Low spp

2.6 Direct light

The direct light implements accurate shadow testing with distance-based occlusion checking, and a simple point light model with uniform intensity distribution.

2.7 Anti-alising

This part is partially implemented by the framework, what really matters is commitSample function, performing proper sample accumulation.

3 RESULTS

The image shown in 1 shows my final render result, and the image 2 show the scene with low spp(spp = 2).