Assignment 1 Report
Computer Graphics

Student Name: Ziyue Qiu
Date: May 2 2025

1 Project Overview

This project implements a CPU-based path tracer in C++, featuring:

1. **Diffuse and mirror surfaces**
2. **Direct lighting and shadows**
3. **Indirect lighting for point light sources**
4. **Antialiasing**
5. **Ray mesh intersection**

2 Compilation and Utilization

All source code lives in `main.cpp` and is managed with CMake:

```
mkdir build && cd build
cmake .. && make
./CG-main    # runs the tracer
```

By default, the executable looks for `model/cat.obj` and outputs `image.png` at 512×512.

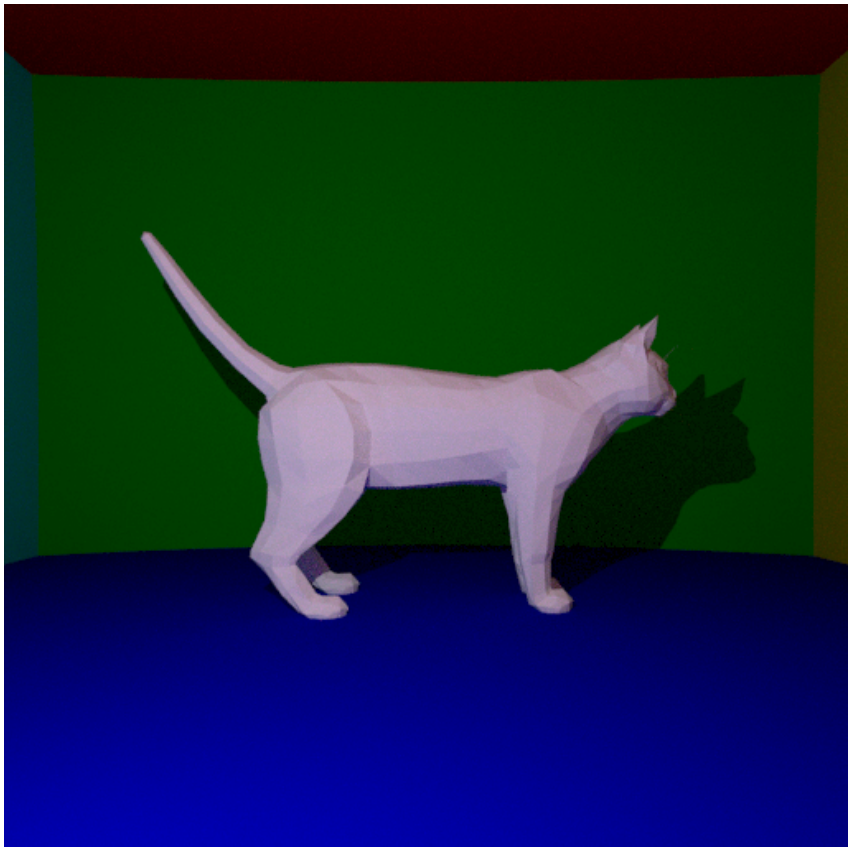
3 Testing Environment

- **CPU:** Intel(R) Xeon(R) W-1270P CPU @ 3.80GHz
- **Compiler:** g++ 11.4.0 with `-O3`
- **CMake:** 3.22.1

All timings below were measured under this configuration.

4 Final Results

Here is the final rendered image of the cat model:



Resolution	512×512
Samples per pixel	64
Max ray bounces	5
Field of view	60°
Light intensity	1×10^7
Cat albedo	$(1.0, 1.0, 1.0)$
Rendering time:	4.55698 seconds

5 Specific Implementation

5.1 Class Organization

- `Vector`: 3D math (operators, dot/cross, normalization).
- `Ray`: origin + direction.
- `Geometry`: abstract base for `Sphere` and `TriangleMesh`.
- `Sphere`: quadratic intersection, normals, mirror & transparent flags.
- `TriangleMesh`: OBJ loader, BVH builder (`compute_bvh`), Möller–Trumbore intersection.
- `Scene`: holds objects & light, dispatches intersections and shading.

5.2 Shading & Lighting

- **Direct:** shadow-ray test, point light falloff.
- **Indirect:** cosine-weighted hemisphere sampling.
- **Mirror & Transparent:** recursive reflection/refraction based on Fresnel.

5.3 Acceleration

- **BVH:** built by median-split on triangle centroids, leaf size = 8.
- **Traversal:** iterative stack, prune by AABB tests.

End of Report