Ray Tracing Renderer

Mei Yixuan

January 8, 2022

1 Introduction

This document describes the design of a simple ray tracing renderer, along with some sample scenes rendered using it. I finished it as course project for Advanced Computer Graphics, given by Prof. Shimming Hu in Tsinghua University.

Complete code for this renderer can be found in https://github.com/AntonyMei/RayTracingRender, along with all external libraries and resources. This code is designed for this course project only and implies no warranty, use at your own risk. All code follow Apache License 2.0, except those external libraries and resources.

2 Results

This sections shows some of the demo scenes rendered using this renderer.

2.1 Hollow Glass Ball

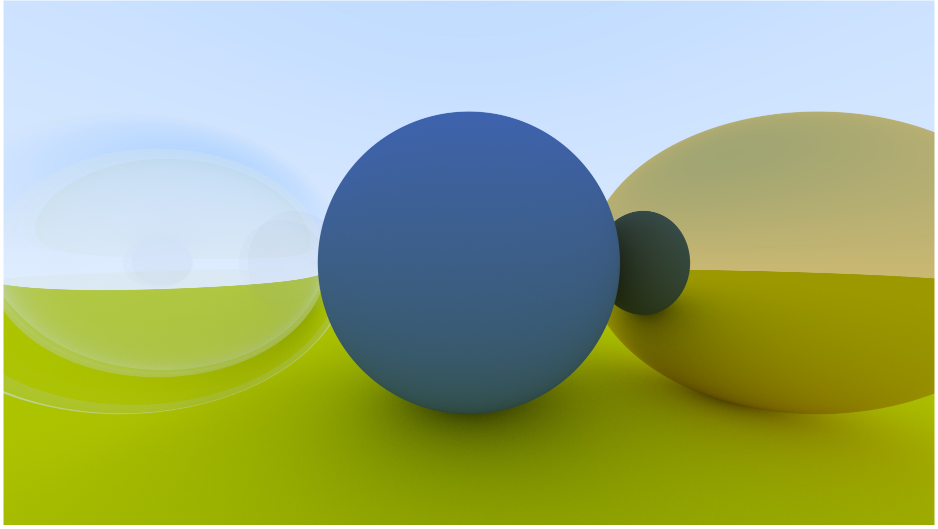

Figure 1: Hollow Glass Ball

This scene shows the usage of Lambertian (pure diffuse), Metallic (pure reflection) and Dielectric (refraction with full reflection) material

3 Usage

4 Renderer Structure

The basic structure of this renderer is as follows:

4.1 Basic Renderer:

Integrator: Monte Carlo Path Tracing, Photon Mapping.
Accelerator: BVH (AABB with SAH), Kd-Tree.
Hardware Acceleration: OpenMP on Windows, Multi-processing on Linux.

4.2 Objects:

Hittable Objects: Triangle, Triangular Mesh, Box, Sphere.
Complex Models: .obj Model with .mtl Material Description.
Skybox: Constant Skybox, Directional Skybox, Realistic Skybox, Multi-layer Skybox.
Transforms: Rotation, Translation.

4.3 Materials & Textures

Materials: Lambertian, Metal, Dielectric, PBR material, Isotropic, Diffuse Light.
Textures: Color Texture, Checker Texture, Perlin Noise Texture, Marble Texture, Image Texture, Bump Texture.

4.4 Visual Effects

Camera Effects: Off-focus Blur, Motion Blur.
Volumetric Rendering: Participating Media.
Other Visual Effects: Gamma Correction, Caustic.

5 Code Explanation

6 References

Code References:
[**1**] Ray Tracing Mini-books, by Peter Shirley (https://raytracing.github.io)
[**2**] Physically Based Rendering: From Theory to Implementation, by Matt Pharr, Wenzel Jakob, and Greg Humphreys (https://github.com/mmp/pbrt-v3)
[**3**] Dezeming Family (https://dezeming.top/)
External Libraries:
[**1**] Tinyobjloader (https://github.com/tinyobjloader/tinyobjloader)
[**2**] stb_image (https://github.com/nothings/stb)
Model Resources:
[**1**] Morgan McGuire, Computer Graphics Archive, July 2017 (https://casual-effects.com/data)