

Assignment 2

Advanced Graphics, Augmented Reality, and Virtual Reality

September 2020

1 Learning Objective

In this assignment, we will build a Recursive Ray Tracing Engine in OpenGL. In doing so we aim to get familiar with how ray-tracing works under the hood. We'll also understand how material properties can affect light rays, and how reflections / refractions of light affect our scene.

2 Problem Statement

We will divide the problem statement of building a Ray Tracing engine into 3 main objectives, which you can complete incrementally.

1. **Single Level Ray Tracing** Setup a simple scene in a cubic room (with differently coloured walls), along with a single source of light on the ceiling of your room, and your camera. Create a procedurally generated textured floor with a black-and-white checkerboard pattern. You should have a cuboid, a sphere, and a pyramid in the room. For this objective, these objects are opaque. You may include more 3D figures as you see fit. The objects can be of a single colour.

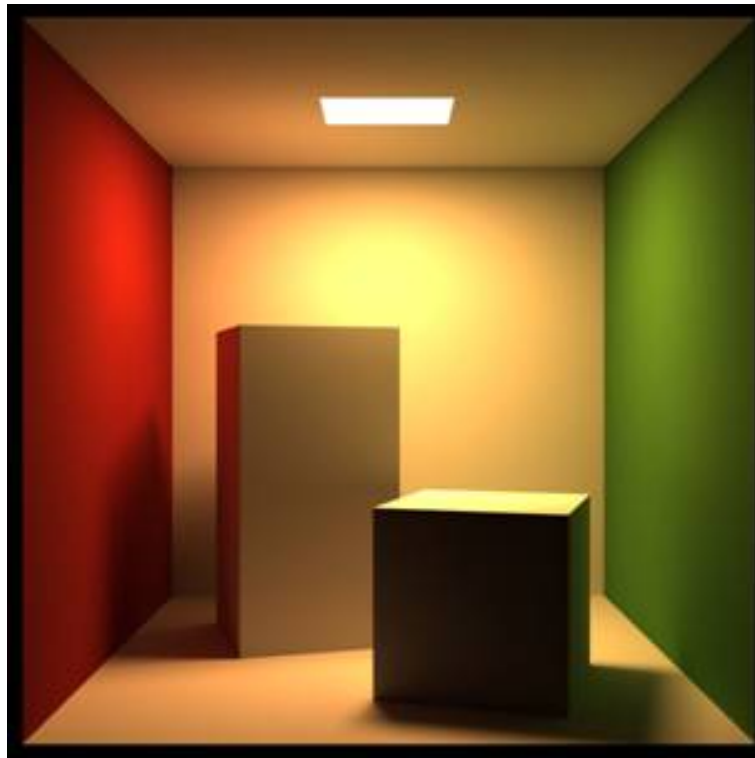


Figure 1: A starting point to give you an idea of objective 1.

2. **Recursive Ray Tracing - Adding Reflections** Expand your previous scene by tracing the path of light over multiple contact points, and accounting for reflections off each object. Maintain the same room, but make the checkerboard textured floor slightly reflective too. Attach different material properties to each object, to have different levels of reflection in the scene, and maintain realistic light illumination. Keep the same objects (a cuboid, a sphere, and a pyramid). The cube should be non-reflective solid colour, the sphere should be a shiny, highly reflective surface (mirror-like), and the pyramid somewhere in between the two.

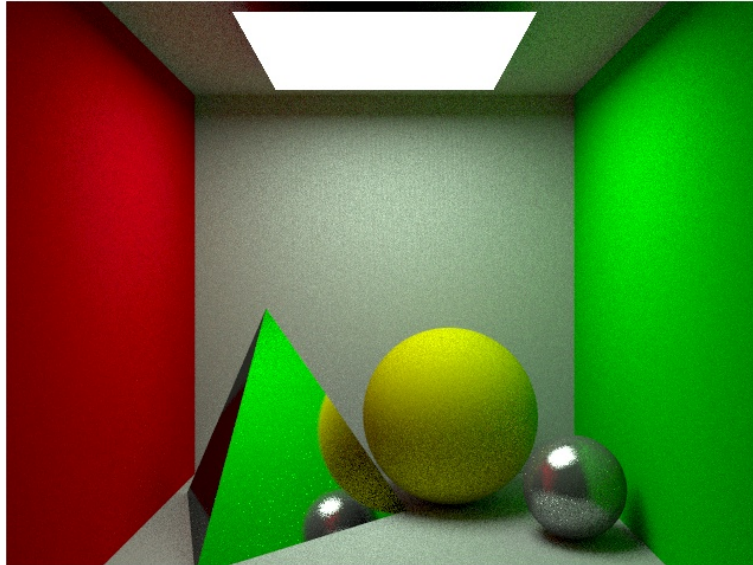


Figure 2: A starting point to give you an idea of objective 2.

3. **Recursive Ray Tracing - Adding Refractions** Expand the previous scene by making objects transparent, and accounting for refracting light rays. Identify whether light effects like Total Internal Reflection are at play here, depending on the materials of the objects. Keep the cuboid a solid, opaque object. Make the sphere a transparent glass ball, and the pyramid as a glass prism. The sphere should be kept in front of the cuboid.

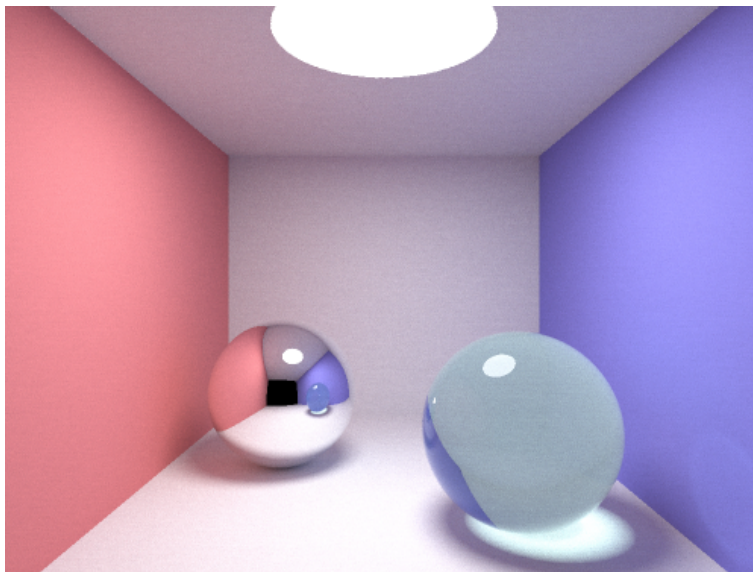


Figure 3: A starting point to give you an idea of objective 3.

3 Additional Information

Keep in mind that the provided images are for reference, and your photos need not be exactly the same, and they need not be as clean or pretty. You may experiment with different/multiple light sources, and pick on a setting that creates a realistic looking image. If you run into noise issues in your images, you can experiment with different sampling rates/denoising algorithms to produce a cleaner image, but image noise will not adversely affect your grade.

4 Marking Scheme

We will grade, along with sir, based on a combination of technical merit, aesthetic appeal, and realism.

- Part1 : 30 points
- Part2 : 30 points
- Part3 : 40 points

Please refrain from copying code from online sources, because the TAs will run plagiarism checks.

5 Submission Guidelines

The assignment can be written in C++ or Python. Your program should generate an image file when run, and you can use any popular image libraries to create them. You should provide a pdf alongside detailing the steps you've taken/algorithms you have used. Please submit all your files in a roll numbered zip on Moodle on or before 11.59 PM September 13, 2020. Keep in mind the late day policy, and use them wisely.

Best of luck!