# Computer Graphics Assignment 1

Tu Nguyen

May 15, 2024

## 1 Overview

In this project, I implement a raytracer in C++, which prolongs for 4 labs, each lab has its own objective:

- Lab 1: Raytracing with diffuse and mirror surfaces to create the balls (sphere objects)
- Lab 2: Add shadow and indirect lighting
- Lab 3: Raytracing with mesh objects to create the cat image
- Lab 4: Add bounding box to reduce rendering time (due to lack of time, I was not able to implement the BVH structure)

## 2 Code Structure

All the code are in the file `main.cpp`. Constant values used in the file:

- "num ray max": number of rays per pixel
- "num reflex max": control the recurrence of the raytracing and facilitate indirect lighting (1 for direct lighting)
- `W, H`: Width and height of the image, normally be 512, sometime smaller to reduce rendering time

Main classes used in the file:

- Vector: facilitate the operation of vectors consisting of 3 doubles, representing the position of the vector in the 3-D environment
- Ray: represent a ray, which consists of a root and a unit direction vector
- Geometry: a "mother" class for the mesh (class TriangleMesh) and the spheres (class Sphere), which has a function "check intersect" to check if a ray intersect with the object or not
- Scene: initiate all the objects (the cat or the spheres) and used to render the image

# 3 Tool

I use my own laptop to render the images (Windows 11)

- Chip: Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz
- Cores: 4
- Memory: 7.8 GB
- Base speed: 2.3 GHz

# 4 Result images

## 4.1 Diffuse sphere

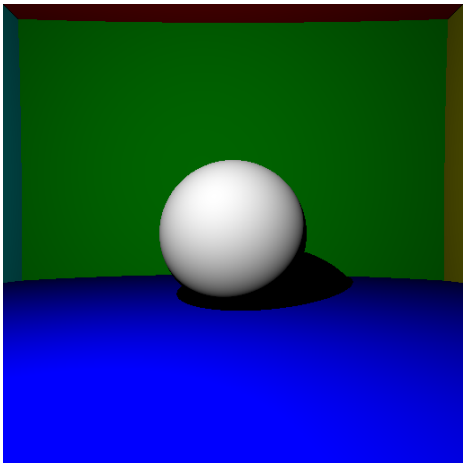

Figure 1: Direct lighting including shadow of 1 sphere - 2 rays per pixel, (rendering in 1 second, using openmp)

## 4.2 Indirect Lighting

## 4.3 Mirror sphere

## 4.4 Ray Mesh Intersection direct lighting

## 4.5 Ray Mesh Intersection indirect lighting

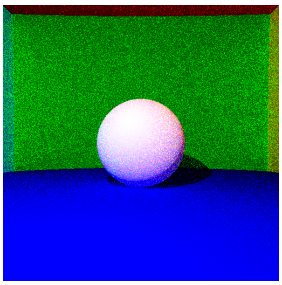

Figure 2: Indirect lighting including shadow of 1 sphere with - 2 rays per pixel, max recurrent: 20(rendering in 2 seconds, using openmp)

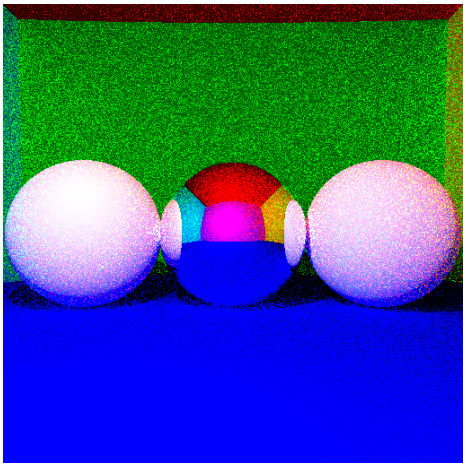

Figure 3: Indirect lighting including shadow of 1 mirror sphere and 2 diffuse spheres - 2 rays per pixel, (rendering in 1 second, using openmp)

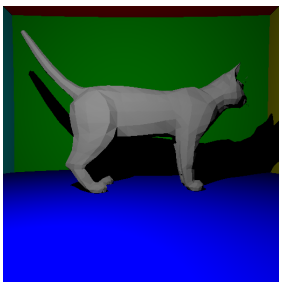

Figure 4: Ray mesh intersection direct lighting (rendering in 15 seconds without bounding box, 7 seconds with bounding box)

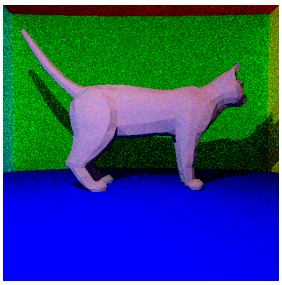

Figure 5: Ray mesh intersection direct lighting (rendering in 207 seconds without bounding box, 43 seconds with bounding box)