CSE306 Project - Raytracer

Philippe Guyard

Code structure

In this project, I implemented all mandatory labs as well as some optional features. I used Cmake as my main build tool, and VsCode as my editor. I do not know if Visual Studio parses CMakeLists files, but if not, the project can be compiled using the command

```
$ cmake . && make
```

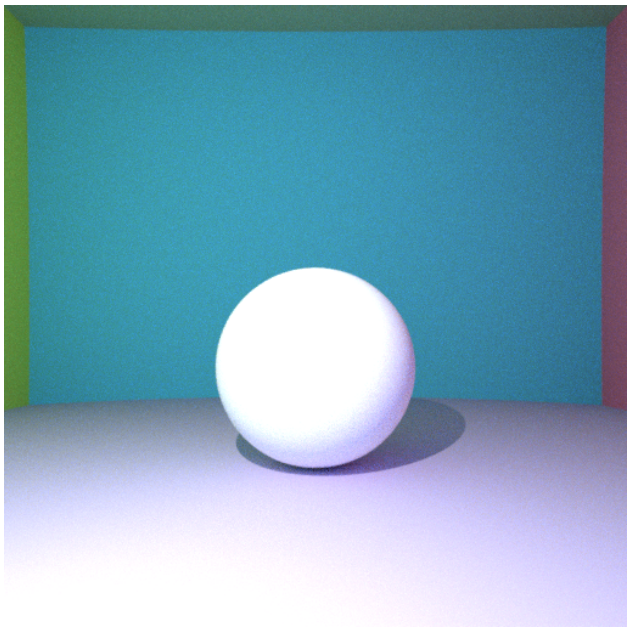
in the main directory. Here is a description for all the project files:

- `main.cpp`: this file contains
  - A simple Image class that works as a wrapper around the given library
  - The actual scene setup (defining the objects, fov, light source, and camera)
  - An implementation for anti-aliasing
  - Part of the indirect lighting implementation. Only averaging on many rays is done in this file, the rest is in `raytracer/scene.h`

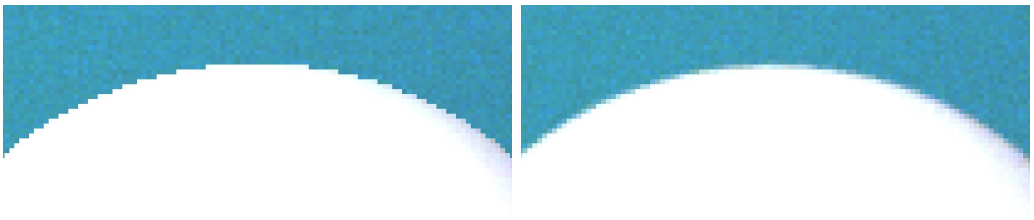
  a simple image wrapper, as well as the scene setup
- `raytracer`: this directory the raytracer in itself. Notably:
  - `raytracer/scene.h`: has an implementation of the main `getColor` method, including an implementation of mirror and transparent materials. Indirect lighting is also added there
  - `raytracer/bvh.hpp`: has a (somewhat general) implementation of the BVH structure. Note that it uses allocators instead of vector indices as integers (as suggested in the lecture notes).
  - `raytracer/sphere.cpp`: has an implementation of the ray sphere intersection algorithm

  - `raytracer/triangle.hpp`: ray triangle intersection algorithm
- `mesh_reader.hpp`: this file is the one provided for reading .obj files
- `stb_*`: these are the header files provided to us for saving images

Pictures



(a) Full image



(b) No anti-aliasing (c) Anti-aliasing

Figure 1: A demo of anti-aliasing.
Rendering time (with -O3) $\approx$ 11 seconds. Parallelization is enabled with dynamic scheduling.
Indirect lighting enabled with 64 rays per pixel.

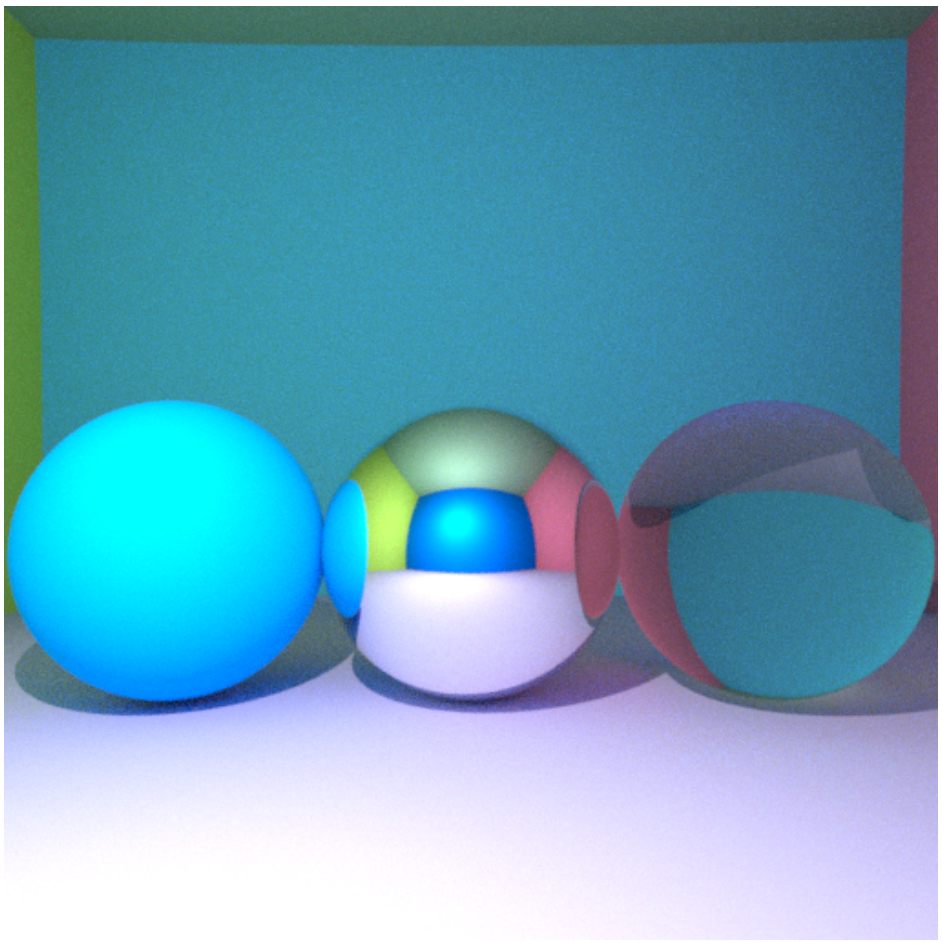


Figure 2: Three spheres each of different material type: diffuse, mirror, and transparent.
Rendering time (with -O3) $\approx$ 11.82 seconds. Parallelization is enabled with dynamic scheduling.
Anti-aliasing enabled, light intensity = 2E10, FOV = 60 degrees, gamma correction enabled, max bounces per ray = 5.
Indirect lighting enabled with 64 rays per pixel.

Figure 3: A cat.
Rendering time with BVH $\approx$ 13.15 seconds.
Parameters same as in figure 2, soft normals enabled

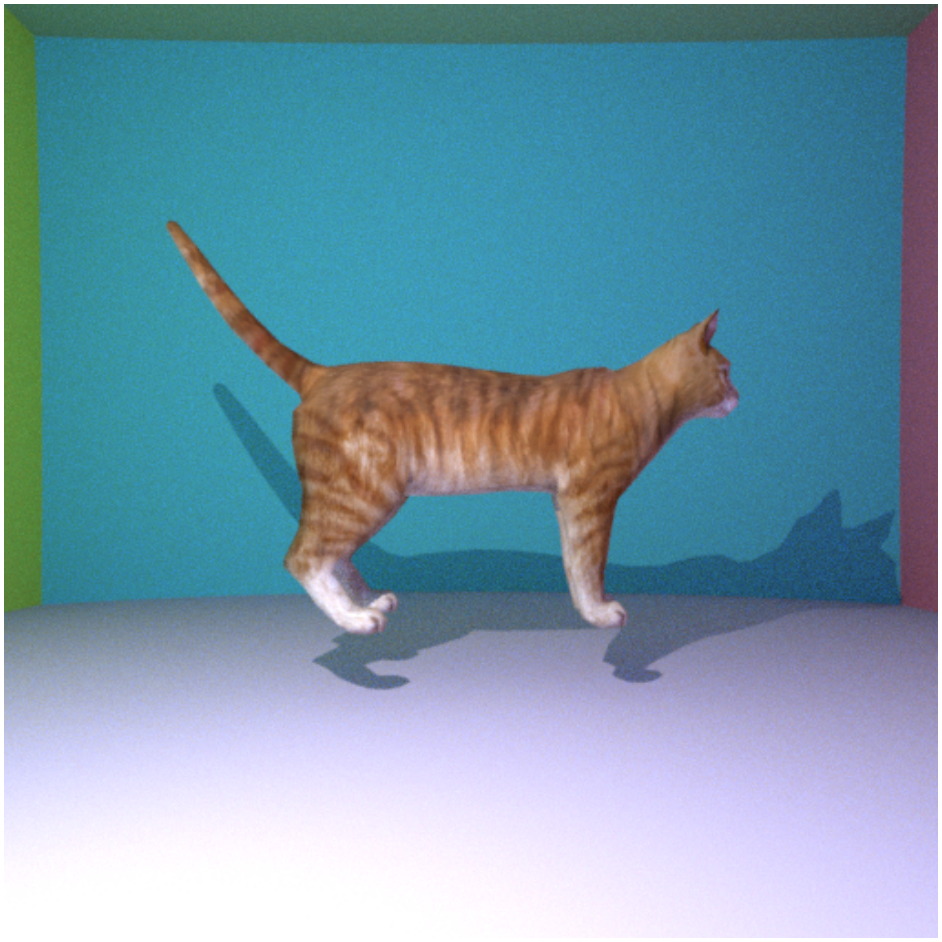


Figure 4: A cat, but this time with textures.
Rendering time (with -O3) $\approx$ 16.5 seconds
Parameters same as in figure 3.