Project Proposal for PFP, 2019 Fall

Jiakang Guo <jg4006>

1. Introduction

The proposed project is going to be a parallel ray tracer. Since I've taken the computer graphics course and have experience writing ray tracers in Java, plus there's already some haskell codes one can easily find[1], I plan to introduce some improvements in this particular project, in terms of render quality and speed. The project would be done alone.

The formal statement of the problem is, given several files describing all the meshes (.obj, for instance), materials and lights in a static scene, as well as the camera pose & parameters, and write the rendered image of certain dimension to the disk.

Some necessary and optional improvements I could think of at this point:
Necessary:

1. Parallel implementation. This shouldn't be hard since most related algorithms are parallel in nature.
2. Support large scene with 10k+ triangles: use tree structure to speed up the computation of ray-triangle intersection. One method is BVHs described in [2].

Optional:

3. Monte carlo ray tracing, as taught in COMS 4160 computer graphics course, by Prof. Changxi Zheng.

2. Development Plans

Each of the following stage should take 3~5 days:

1. Specify input formats, and generate a naive image purely based on material color, ignoring light.
2. Implement and test the date structure in [2].
3. Basic parallel ray tracer based upon 1 and 2.
4. (optional) Modify 3 into a monte carlo ray tracer.
5. Build up a complex scene as the final demo.

3. References

[1] Htrace, Haskell RayTracer http://www.nobugs.org/developer/htrace/index.html
[2] Notes on efficient ray tracing, Solomon Boulos, University of Utah