

Ray tracing

Yan Li (yan_li@brown.edu)

Brown University

Introduction

Ray tracing is originally a project of our graphics course developed on Qt's API. The requirement for this project was to implement the ray tracing algorithm. However, I added a lot of features just for fun. These features includes: using KdTree data structure to speed up, using GPU to parallelize the computation to make the rendering real-time, super sampling and better refraction.

In the project, I both kept the implementation of GPU and CPU ray tracing in order to make a comparison for debugging reasons. For GPU part, I used OpenCL's API to distribute the computation. The design for the parallelization is pretty same because the computation of each pixel in the final image is totally independent with other pixel.

In CPU side, the normal processing time will be several seconds using the original algorithm. With the help of KdTree structure that is a way of doing space subdivision and GPU's parallelization, the speed could reach real-time.

Demo

<https://www.youtube.com/watch?v=2qHus4QMfvk&list=UUUVul0Yfzw0oitbOE5LM31bg>

Code info

5000 – 6000 lines of C++ code

Files

I have put duplicates of the source files that are written by myself in “./my_work”. Feel free to take a look. Please notice that raytraceGPU.cl is GPU code. It's actually pretty similar with C code.

Execution

To run the code you must have OpenCL's, Qt's library and a good graphics card. I would suggest not building the project because it really takes some work.

Thank you for reading my code.