Parallel Path Tracer using CUDA

Jim Liu
Owen Wang

URL

https://jim3060.github.io/PPwebsite/index.html

SUMMARY

We will implement an optimized path tracer leveraging GPU computing resources.

BACKGROUND

Path tracing is a computer graphics Monte Carlo method of rendering images of three-dimensional scenes such that the global illumination is faithful to reality. We shoot rays from every pixel on the camera and accumulate their illuminance along their path.

In 15-462, we implemented a single-threaded path tracer. However a few problems are present.

Firstly, the path tracer is slow as the path tracer shoots rays from every pixel. We expect to be able to parallelize this algorithm.

Secondly, the final output image currently contains non-trivial amounts of noise. One solution (KM15) is to use the gradient information to smooth the image which involves solving a Poisson equation. This requires solving a sparse linear system which has the size of (imagesize)^2, and is another area open to parallelization and improvement.

THE CHALLENGE

The first challenge is converting the single threaded CPU program into a CUDA implementation. We will be required to find a way to stream all the BVH structure and material information into CUDA.

The second challenge is improving memory access efficiency. In lecture 8, we talked about the inefficient memory access problem of ray tracing and one solution using ray packets (SIMD implementation). This problem will still be present in a CUDA implementation.

A further challenge is smoothing the image by solving PDE. Since the linear system we are going to solve is very large, we must design an algorithm taking advantage of matrix blocking for

efficient memory access. We plan to research different algorithms and to find the one with greatest parallel performance.

RESOURCES

- RTX 2080 from GHC machines
- Starter code from: https://github.com/RayTracing/raytracing.github.io/tree/master/src/TheRestOfYourLife (We use this instead of code from 15462 because it is clearer and support texture)
- Kettunen, Markus, et al. "Gradient-domain path tracing." ACM Transactions on Graphics (TOG) 34.4 (2015): 1-13.

GOALS AND DELIVERABLES

- A functioning and parallel path tracer.
- An analysis report around the performance along the implementation. For example, performance comparison between using ray packaging or not; speedup comparison between single threaded version and parallel version and the ideal; performance comparison between different PDE solvers.

PLATFORM CHOICE

CUDA, GPU, RTX 2080

SCHEDULE

Week	Items
Week 10	Proposal and coding environment setup
Week 11	Implement CUDA path tracer without gradient info
Week 12	Record and analysis basic path tracer’s performance
Week 13 (Mid)	Implement the gradient domain method
Week 14	Parallel the poisson solver
Week 15	Analysis the solver and path tracer’s whole performance
Week 16	Finalize and write report