Parallel Path Tracer using CUDA

Jim Liu
Owen Wang

URL

SUMMARY

We are going to 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. But there exist some problems. First, the path tracer is slow. And because the path tracer shoots rays from every pixel, in theory, this pattern can be paralleled.

Second, the final output image is still a little bit noisy. One solution for that 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, which is also a place where we can parallel and improve.

THE CHALLENGE

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

The second challenge is improving the memory access. In lecture 8, we talked about the inefficient memory access problem of ray tracing and how to use ray packet (SIMD implementation). And CUDA implementation will also face the same problem.

And we also have a second part which is to smooth the image by solving PDE. Since the linear system we are going to solve is very big, we will design around the blocking of the matrix and memory access of it. Also we plan to research different algorithms and find out the best one to parallel.

RESOURCES

RTX 2080 from GHC machines
Starter code from Computer Graphics: 15462
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