

# Checkpoint: GPU Ray Tracer with Optimized Parallel BVHs

## Work Done So Far

We've finished parallelizing C++ implementation of ray tracing on [smallpt](#) to work on GPU and finished debugging. We've went through Karras's [paper](#) on Bounding Volume Hierarchy (BVH) tree construction on GPU and ready to start implementing at this point.

## Comparing Current Work & Proposal

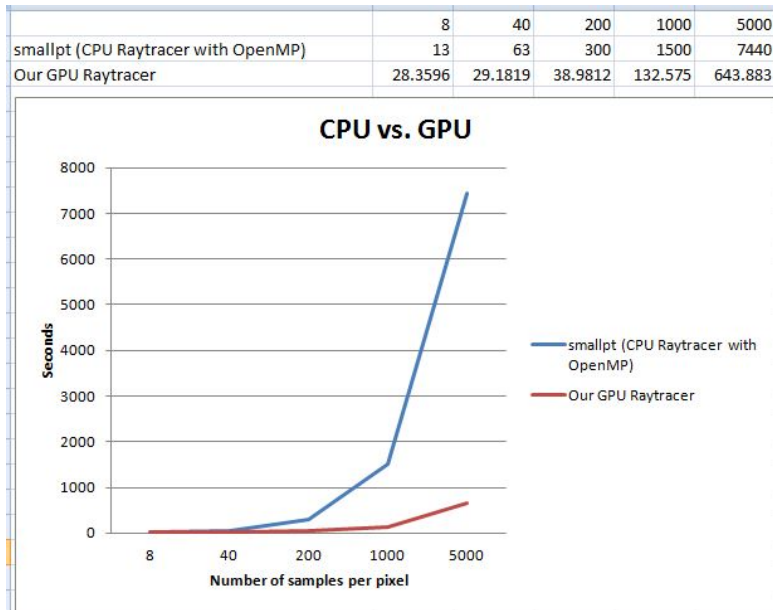
We are planning to finish Karras's both papers, one on parallel BVH construction and one on BVH optimization eventually, and we think we can keep up with the schedule. Looking for a proper C++ ray tracing library that can be ported to CUDA cost us a considerable amount of time, and since that's already done, we are relatively comfortable with adding code/data structures to GPU side now.

## What to Show at Parallelism Competition

We are planning to show several graphs on runtime comparisons of ray tracer implementation on CPU, on GPU, on GPU with BVH, on GPU with optimized BVH.

## Preliminary Results

We have a graph of CPU ray tracer vs. GPU ray tracer performance at this point:



## Issues

As mentioned earlier, we spent vast amount of time trying to find a ray tracing library that we can port to CUDA. We've tried CMU 15-462's ray tracing project starter code, Don Cross's [ray tracer](#), and finally [smallpt](#). The previous two both failed because they use too many external libraries, and we weren't able to find the CUDA alternatives for them. Also, compilation using CMake also took us a lot of time.