15418 Milestone Report

Parallel Packet Tracing in Scotty3D

Kenechukwu Echezona

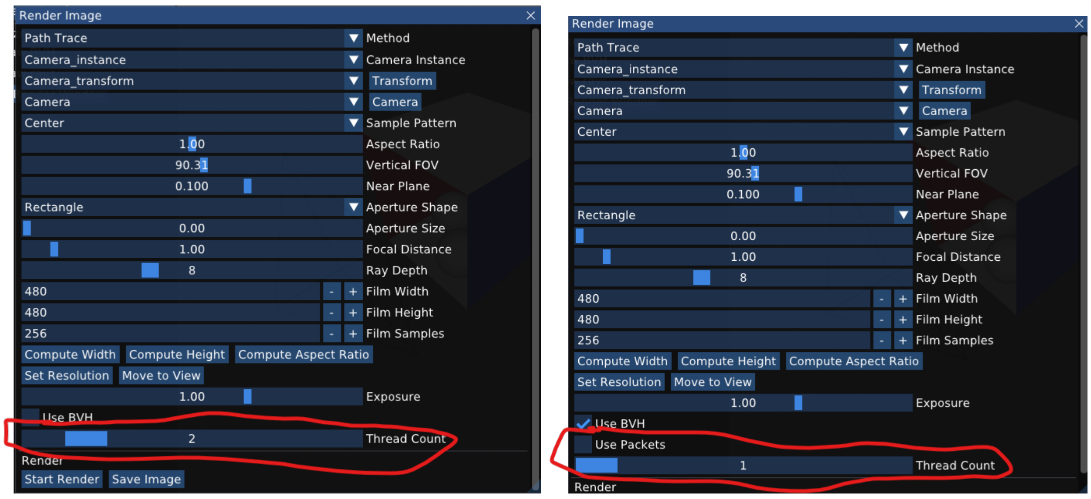
Updated Project Page

https://keche1234.github.io/15418-Final-Project-CPU-Ray-Tracing-via-Packet-Based-BVH-Traversal/

Due to my partner dropping the course, I'll need to adjust the scope of the project to complete it on time.

What's Been Done

We've developed a Ray Packet data structure, and studied the Scotty3D base code enough to understand where we might best utilize OpenMP and ISPC (the former to parallelize across the Ray Packets, and ISPC for BVH navigation within a Ray Packet, respectively). We also figured out how to edit the application itself to support our Parallel Packet Tracer as an option.



Render Image

Method: Path Trace
Camera Instance: Camera_instance
Transform: Camera_transform
Camera: Camera
Sample Pattern: Center
Aspect Ratio: 1.00
Vertical FOV: 90.31
Near Plane: 0.100
Aperture Shape: Rectangle
Aperture Size: 0.00
Focal Distance: 1.00
Ray Depth: 8
Film Width: 480
Film Height: 480
Film Samples: 256
Compute Width | Compute Height | Compute Aspect Ratio
Set Resolution | Move to View
Exposure: 1.00
□ Use BVH
Thread Count: 2
Render
Start Render | Save Image

Render Image

Method: Path Trace
Camera Instance: Camera_instance
Transform: Camera_transform
Camera: Camera
Sample Pattern: Center
Aspect Ratio: 1.00
Vertical FOV: 90.31
Near Plane: 0.100
Aperture Shape: Rectangle
Aperture Size: 0.00
Focal Distance: 1.00
Ray Depth: 8
Film Width: 480
Film Height: 480
Film Samples: 256
Compute Width | Compute Height | Compute Aspect Ratio
Set Resolution | Move to View
Exposure: 1.00
☑ Use BVH
□ Use Packets
Thread Count: 1
Render

Figure 1: New additions to the render window, edited in via `widgets.cpp`. Ray packets are specifically used for BVH navigation, so the packet option only appears when the "Use BVH" option is selected.

Where I Am

I have a code skeleton of the appropriate areas for parallelization, which methods to overload, and which to create new versions of to support parallelization. We had tested our Ray Packet structure via a sequential structure and got near identical results.

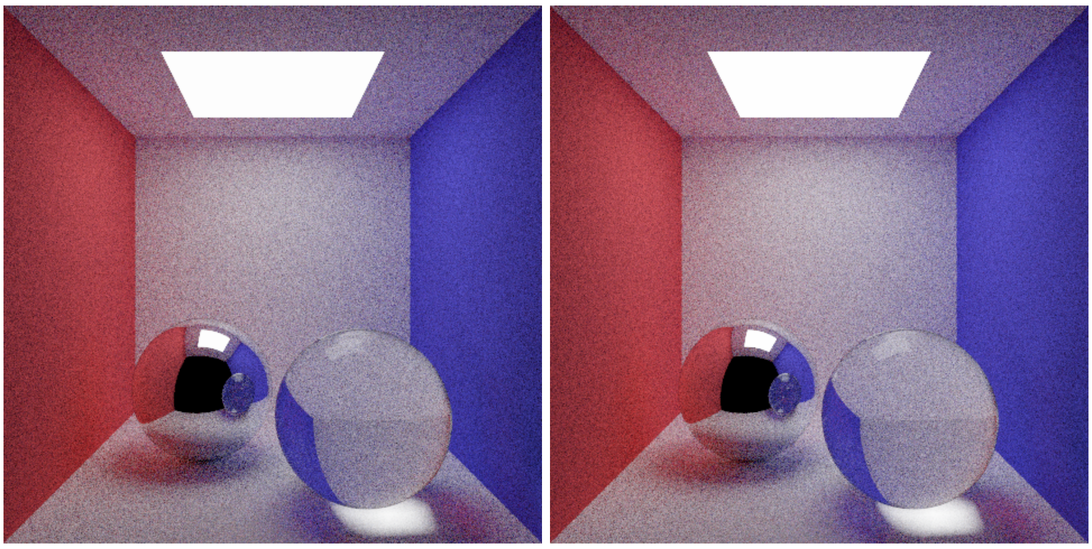


Figure 2: Left: Rendering `A3-cbox-spheres.js3d` using the sequential Ray implementation. Right: Rendering the same file using the sequential RayPacket structure.

Final Submission

In addition to the GitHub link for the Scotty3D code itself (which must only be shared with the graders due to this being based off of a submission from another class), the final submission will consist of (1) a visual, qualitative comparison of rendered sample scenes, and (2) speedup graphs comparing the sequential, OpenMP packetless, ISPC Ray Packets, and OpenMP + ISPC Ray Packet implementations, and (3) a discussion of the result

Issues

I would like to run Scotty3D on one of the Gates Clusters, but despite deleting every file in my private repository from prior semesters and even some files from this semester, the files from this class and another class I'm taking (where I also need the GHC Clusters) are taking up over 90% of my storage. As a result, all tests and comparisons for Scotty3D will need to be done on my computer, an Intel-based HP ENVY x360 with 10 Cores and a 64-bit Instruction Set (which I believe means my computer can support 8-lane SPMD on 8-bit values, or 4-lane SPMD on 16-bit values). If the `OldFiles` repository clears up, then I'll be able to run the Maekfile.js on the Scotty3D repository and see if I can get the window to open through the GHC Clusters.