**Advanced Graphics - Assignment 3: Specialization**

*Dustin Meijer(s5726328) & Luuk van de Wiel(s4088212)*

**TL;DR**

**Specializations chosen**

* CPU Path Tracer
* GPU Whitted-style Ray tracer ( here is a blocking call to glGetUniformLocation in shader.cpp)

**Basic Functionality**

* CPU Path Tracer with RR, MIS, NEE, Refraction, Reflection

**Introduction**

For the course Advanced Graphics at the University Utrecht we focus on studying methods for creating physically accurate images from virtual environments and how to make the process of creating these images run at an interactive frame-rate. This was the last assignment focusing on a specialization towards a focus area. We chose the focus areas of a CPU Path Tracer and a GPU Whitted-style Ray Tracer.

The basic functionality required for a CPU Path Tracer was the following:

“A CPU path tracer with at least basic variance reduction (IS, NEE, RR), basic materials (Lambert, pure specular, dielectric) and triangle mesh support. “- Advanced Graphics Assignment 3 specification by Jacco Bikker

The basic functionality required for a GPU Whitted-Style Ray Tracer was the following:

“A functional GPU ray tracer or path tracer with triangle mesh / BVH support. “- Advanced Graphics Assignment 3 specification by Jacco Bikker

There was also the option to attain additional points by implementing some or all of the following:

* Produce an interesting demo (nice scene or animation)
* Add image postprocessing
* Achieve high performance
* Implement advanced variance reduction
* Implement advanced variance reduction on the GPU.

The rest of this document will discuss what functionality we decided to implement, how it was implemented and what difficulties we faced.

**Implemented Functionality**

In this section we will discuss what functionality was implemented. This is divided in a section about the CPU Path Tracer and a section about the GPU Whitted-Style Ray Tracer.

**CPU Path Tracer**

Our first focus was to create the most basic Path Tracer imaginable so that we could obtain a ‘ground truth’ image to compare our more advanced implementations with. This path tracer be found in the BasicSample function in Renderer.cpp.

To compare outputs of different path tracer implementations, we use the total energy count of the scene. This sum of energies from each pixel should eventually be (almost) the same if the path tracers converge. The second tactic we used was to compare to implementations side-by-side by splitting the image in two halves, this was useful for quick identification if something was awry.

Sample

DiffuseReflection

CosineWeightedDiffuseReflection

DirectSampleLights

SampleMIS

BVH Traversal Issues

**GPU Whitted-style Ray Tracer**

Our goal was to eventually creating a GPU based path tracer, sadly this was not meant to be. There were a lot of engineering troubles to overcome with merging our project from assignment 2 and the OpenCL template project that was provided to us by Jacco Bikker. After some help from Jacco, the two projects were successfully merged. Development started on the Whitted Style Ray Tracer, with trying to get data to the GPU and finally the creation of a test kernel where pixels would light up if anything was hit. In the end, the GPU ray tracer does not run because when registering the shader, the call to glGetUniformLocation in shader.cpp does not return. On the internet, no information could be found about how to fix this. There is a test kernel implemented in program.vl and two passthrough shaders were added to the project as well.

**Scene(s)**

*TestScene* - For testing the path tracer. We created a scene that contains a single sphere, a red wall to the left, a green wall to the right and white back-wall, ceiling and floor. There is two triangle light sources in the scene arranged as a rectangle in the ceiling against the back-wall.

**Work Division**

CPU Path Tracer

Basic Sample - Luuk

Next Event Estimation - Luuk

Refraction - Luuk

Reflection - Luuk

Multi Importance Sampling - Luuk & Dustin

Russian Roulette - Luuk

GPU Whitted-style Ray Tracer

Merge of AG-Ray Tracer and OCL-lab-template - Dustin

Test Kernel - Dustin

Report - Dustin & Luuk