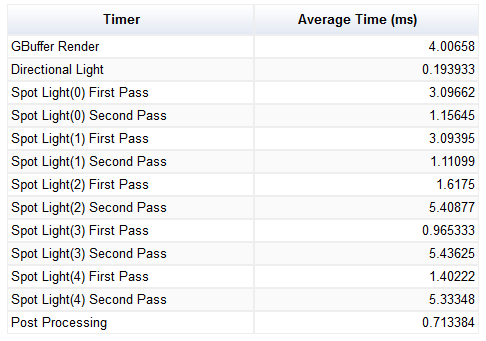
Profiling of Real-Time

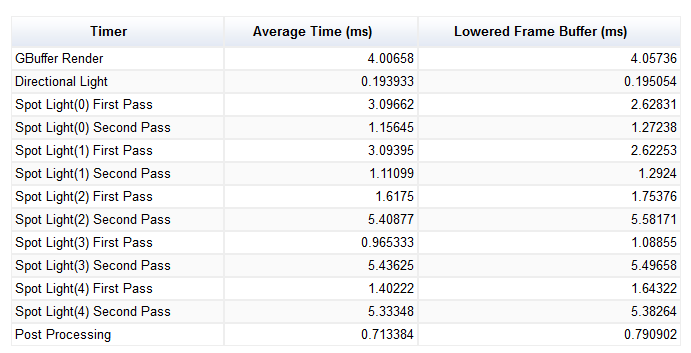
Translucency

Sam Oates

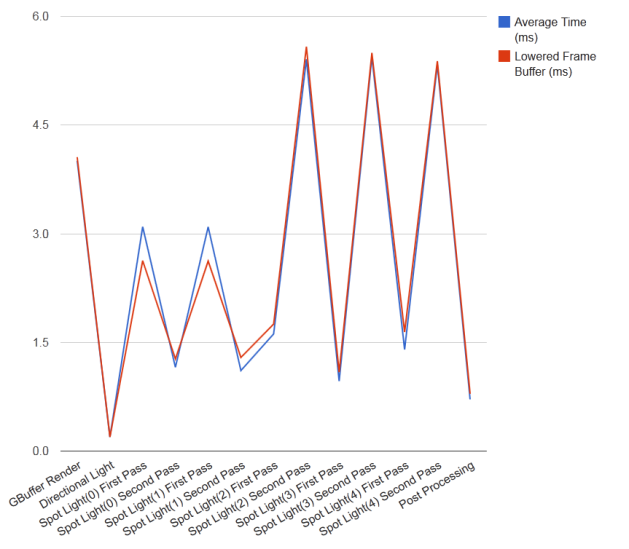
Overview

I began by running performance analysis upon my fully functioning translucent deferred renderer. The program was run in the default window size and the camera animated about a path around the scene. All results where collected and an average of the times were output. The test ran for 60 seconds. The shadow frame buffer is using 4 full RGBA 32 bit floating point textures and the geometry buffer uses RGBA 16 bit floating point textures. The following data was accumulated:

The application averaged at 29.8 frames per second. As can be seen from the table of results, the slowest part of my application is the second pass of the spot lights. The second pass of the spot lights is where the translucency is calculated. However, translucency is only calculated on the spotlights 2, 3 and 4. Meaning the translucency itself is only costing around 2 milliseconds per spot light.

Texture Bandwidth

To test for issues within the texture bandwidth I changed my shadow map frame buffer from having 4 RGBA 32 bit floating point textures to having;

* One single channel 32 bit floating point texture
* Three RGBA 16 bit floating point textures

The results of the lowered texture format can be seen in the above table and visually compared in the adjacent line graph.

There is a marginal improvement using the lower quality textures as expected. However this improvement only results in an average frame rate improvement of two fps, resulting in an average fps of 32. Implying the larger buffer is having a negligible effect on performance.

Rasterization and Frame Buffer Bandwidth