Masterclass Assignment Documentation

Joe Withers*

1 Approach

2 Results

3 Future Improvements

Unfortunately I wasn't able to implement a sparse data structure as described in the source material[?] before the assignment deadline. Having a sparse data structure to store the emissive voxel texture would save a significant amount of GPU memory consumption, allowing the resolution for voxelization to be much higher; currently dense 3D textures it is capable of hitting the 2 gigabyte memory limit of my GPU.

The speed at which lighting rays are marched during the light injection pass would also be improved, as it would be able to step further through the voxel texture whilst ray marching if the higher levels in the octree are known to be empty.

I did start working on a sparse voxel octree as described in OpenGL Insights[?] by using an OpenGL Atomic Counter for calculating the number of fragments needed for the voxel fragment list, and I believe I understand the technique well enough should I wish to implement it in the future.

The source material[?] also describes a method for anisotropic mip-mapping of the 3D texture, storing a version of the texture for each directional axis, which improves visual accuracy when cone tracing. Whilst I could have implemented this with the dense 3D textures I was using, I opted to use OpenGL's built in mip-mapping to save time.

* e-mail:joewithers96@gmail.com