

Masterclass Assignment Documentation

Joe Withers*

1 Approach

2 Results

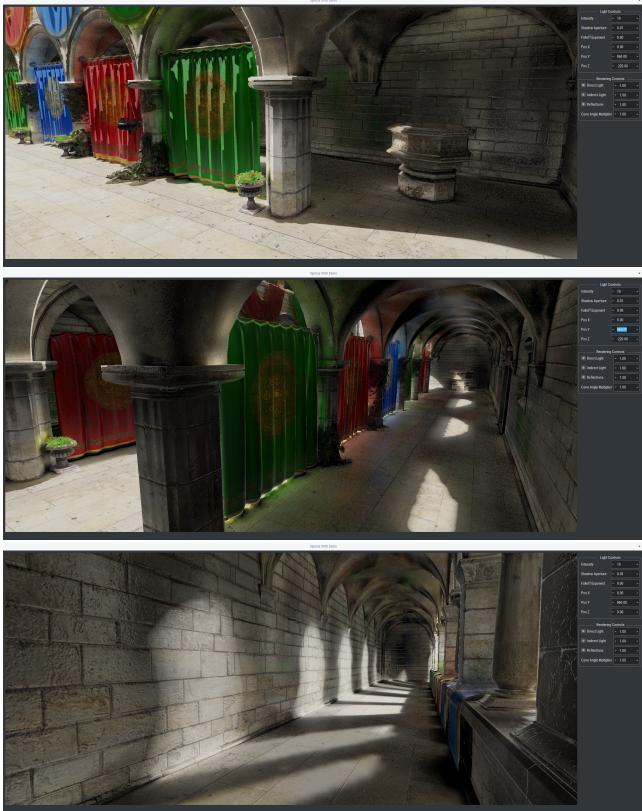


Figure 1: Three screenshots showing the full scene with a voxel resolution of 512^3 .

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.

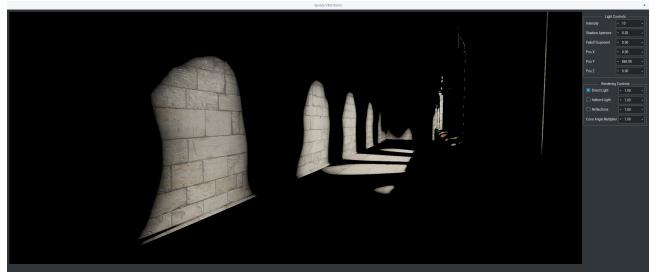


Figure 2: A screenshot showing just the direct lighting pass.



Figure 3: A screenshot showing just the indirect lighting pass.

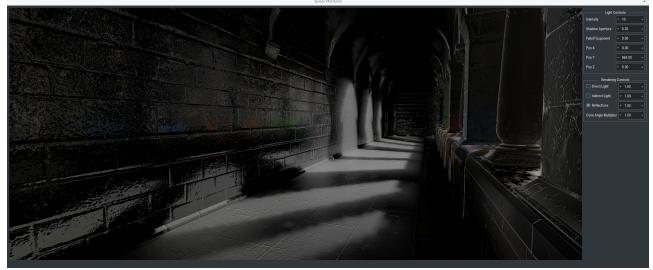


Figure 4: A screenshot showing just the reflection pass.

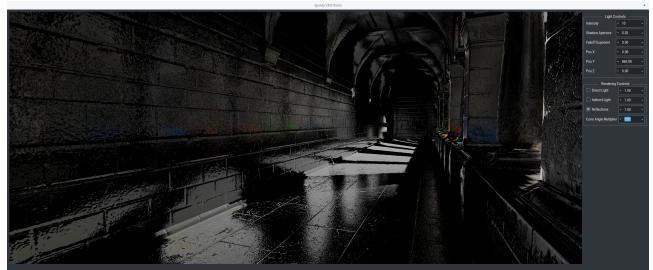


Figure 5: A screenshot showing just the reflection pass, but with the minimum cone angle.

* e-mail:joewithers96@gmail.com

Voxel Resolution	256^3	512^3	768^3
Frames per second	60	60	60
Memory usage	1063MiB	2615MiB	6628MiB
Voxelisation	2ms	2ms	2ms
Light Injection	65ms	378.5ms	12294.5ms
Shading	4.5ms	4.5ms	4.5ms

Figure 6: Averaged performance results when running with a NVIDIA GTX 1080 8GB.

The source material[?] also describes a method for anisotropic mipmapping 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.

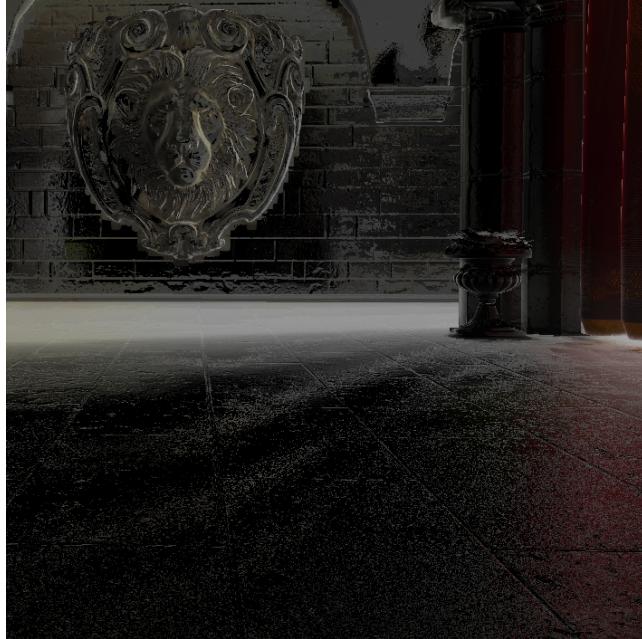


Figure 7: A screenshot showing banding artefacts in the reflection pass.



Figure 8: A screenshot showing noise in the indirect lighting.

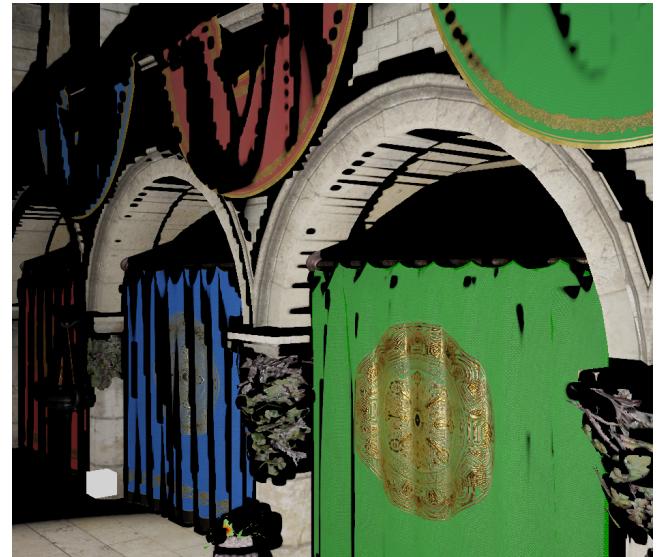


Figure 9: A screenshot showing voxelization artefacts.