Jeremiah Darais
Advanced Shader Programming
Volume Renderer Writeup

The goal of this project was to create a ray-casting volume renderer using shader programs. At first, I set out to implement this with Cg, since I had already had some experience with GLSL, and learning Cg seemed like a good idea. After trying to implement the ray-caster with Cg and running into too many problems that I didn't know how to solve, I went back to GLSL. One of the problems I encountered was interpolation error. If colors and texture coordinates are interpolated in screen space, the result is inaccurate if the triangle isn't facing the viewer directly. OpenGL allows you to set a rendering hint that will tell it to do the slower but more accurate interpolation that accounts for perspective, however my program still seemed to be interpolating in screen-space. This of course did not work for my purposes, especially since I was rendering the back and front faces of a color cube to get the world-space in and out positions of a ray being cast in the fragment program. The interpolation error caused my volume to be really wavy where the bounding-cube faces were near perpendicular to the viewer. I also had some trouble getting textures to the fragment program, and I wasn't finding much documentation on the Cg OpenGL API. When I switched back to GLSL, I had a functional volume renderer quite quickly because it was merely a matter of porting my Cg program over to GLSL. It was nice that after switching, OpenGL was doing correct interpolation of colors.

For the ray-casting algorithm, I used a back-to-front method, since that translates into simple alpha-blending, which was familiar and easy, even though that doesn't give any chance for early ray termination. I could get decent results by doing 500 samples per ray, meaning in texture space, distance between samples was 0.002.

To do lighting, I made 6 more texture lookups to get the dx,dy, and dz of the local region which I could use to get a normal and compute simple diffuse shading.

One rendering artifact I was unable to get rid of was the appearance of stripes along the volume. I experimented with the renderer quite a bit to find out what was causing it. I have some reason to suspect that they were caused by the volume data, since the stripes seemed attached to the volume, and not the screen.