UNIVERSITÄT STUTTGART
Institut für Visualisierung und Interaktive Systeme (VIS)

Thomas Ertl, Steffen Frey *Stuttgart, 2019-06-14*

Scientific Visualization (Assignment 9)

Exercise 9. 1 [3 Points] Octree Structure for Isosurfacing

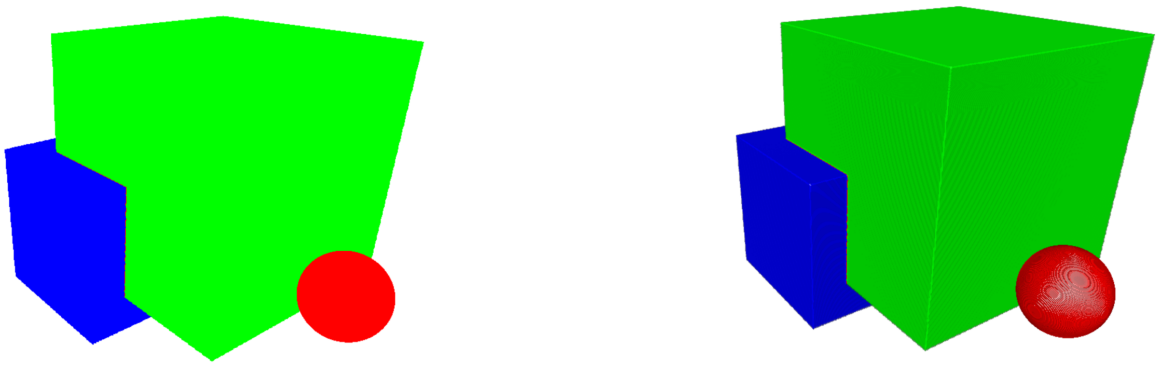
Consider a complete octree, i.e., each leaf node is of a voxel size, for isosurface extraction. The dataset to be visualized is a volume with a regular grid of $500 * 500 * 100$, and each voxel is a 32 bit floating point scalar. Calculate the memory consumption of the complete octree for this dataset in bytes. Each octree node records the minimum and maximum of the data under its spatial coverage.

Exercise 9. 2 [6 Points] Volumetric Illumination

Prerequisites Obtain the skeleton shader (`volumetricLighting2.glsl`) from ILIAS and familiarize yourself with ShaderToy (`https://www.shadertoy.com/new/`). ShaderToy only needs a WebGL-compatible browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. To get started, just copy the skeleton shader into the source text box on the right and press the play button on the lower left of the source box.

ShaderToy essentially gives you a pre-configured fragment shader you can work with. Opposed to standard fragment shaders from OpenGL, for example, some additional input parameters like the current time, mouse coordinates, or sound processing capabilities are available.

Tasks Volumetric Illumination is a good way to improve the visual quality of volume renderings. In this task, you will program your own simple volume lighting, including the computation of the needed surface normals.



(a) (b)

Figure 1: Volume raycasting without illumination (a) and with Phong illumination with central differences gradient computation (b).

The program skeleton provides a fully functional volume raycaster, raycasting a volume that encloses two cubes and a sphere. You should see the result as shown in Figure 1(a) when running the skeleton code in ShaderToy. Your tasks will be:

1. Compute the gradients using intermediate differences.
2. Compute the gradients using central differences.
3. Shade the scene using the Blinn-Phong shading model.

All positions where code has to be added are marked with a `TODO` and `Insert code here`. Note that you will have to use parameters defined at the beginning of the code. By uncommenting / commenting the respective `#define` lines you can switch between the usage of central and intermediate differences. The result using central differences is shown in Figure 1.

Hint: `lightDir` is the direction away from the light, not towards it.

Hint: You can play with the alpha values in the transfer function to get transparent objects.

Submission Deadline: 2019-06-21, 23:55

please hand in your submission through the ILIAS system.

.