02561 COMPUTER GRAPHICS **IMM.DTU**

Exercise 02561-13 **Volume Visualization using Raycasting**

Readings

Class notes, Watt chap. 13.4 (Raycasting)

Introduction

Hardware ray casting allows fast and better volume visualization as compared to the slice based volume visualization. In this exercise you will explore and implement a very simple ray caster to visualize a volume model. We will use OpenGL to setup and trigger a fragment shader that performs the actual ray casting.

Part 1

Setup OpenGL and load the volume texture. Your tasks are to:

- Enable back face culling.
- Enable and load a 3D texture.
- Set texture min/mag filters to GL_LINEAR and set texture wrapping modes to GL_CLAMP.

Question: Why is back-face culling necessary?

Part 2

Display a unit cube and enable shader program:

- Set active texture to GL_TEXTURE0 and bind GL_TEXTURE_3D to the texture object.
- Enable shader program and pass texture and camera information (uniform variables)
- Draw a unit cube centered at origin.

Part 3

Write the shaders:

- Pass object space vertex position from the vertex shader to the fragment shader
- In the fragment shader, create a ray directon starting from the camera position to the vertex position.
- March the ray starting from the vertex position in small steps until √3 (length of the longest diagonal in a unit cube).
- Accumulate color while marching the ray. (refer to assignment class notes).

Question: How would you compute the ray direction in a fragment shader if the vertex position were not sent from the vertex shader?

Part 4

Modify the fragment shader so that the rays terminate at the rear end of the cube. (Optional)

Delivery

Submit source code and images showing the visualized volume and answer the questions.

OS.02.12.2008