E0 271: Graphics and Visualization Assignment #2

Weightage: 25%

Due: Sept 30, 2021

Goals

- Learn and implement lighting, shading and textures.
- Learn and implement some algorithms to visualize scientific data.

Data and Tasks

The data for this assignment can be found in `https://klacansky.com/open-scivis-datasets/`. Each dataset contains scalar fields in .raw format and can be read using the rawfileparser. Wherever applicable use **appropriate colormaps** as discussed in Assignment 1.**[N]** denotes the weightage of each task out of **100**.

1. **[10] Lighting:** Implement Phong shading with Phong illumination model using shaders. Use the "1duk.off" dataset (white coloured) to demonstrate. Perform following experiments.
   - **a. [7]** Have 3 different coloured light sources (ambient, diffuse and specular).
   - **b. [3]** On pressing "1", rotate specular light source continuously around y-axis.
2. **[35] Scalar field visualization: Slicing**
   - **a. [15]** Given an axis parallel plane, define a slice then compute and display the scalar field restricted to the slice. Also display the bounding box for scalar field.
   - **b. [15]** Repeat the same task on GPU. Use 3D textures within shaders to represent the scalar field. Use 1D texture in shader for color mapping.
   - **c. [5]** Comprehensively, compare the CPU-based and GPU-based approaches in terms of image quality and performance, and report your observations. Also report the time spent for preprocessing in both the approaches.
3. **[55] Scalar field visualization: Isocontours and Isosurfaces**
   - **a. [10]** Given a scalar value, implement marching triangles algorithm to extract isocontour for the scalar field restricted to the slice on CPU. Store the extracted isocontour in a VBO and display using standard OpenGL draw() call as GL_LINES.
   - **b. [10]** Extract and display isocontour using GLSL shaders by highlighting appropriate pixels in the output image.
   - **c. [15]** Given a scalar value, implement marching tetrahedra algorithm to extract isosurface of the scalar field on CPU.
   - **d. [15]** Implement marching tetrahedra or marching cubes algorithm to extract the isosurface using GLSL shaders. Use geometry shader for this.
   - **e. [5]** Compare the CPU-based and GPU-based approaches in terms of quality, performance and pre-processing effort, and report your observations.

Notes

General:

1. While user input and interactions won't be evaluated, it would help in the demo and also in debugging. So while not mandatory, virtual trackball can be explored to enhance user interaction. You may use this code available online[1].
2. Take the inputs like slicing plane information or isovalue from command line.
3. Use continuous rotation for demonstration.
4. Experiment with different data sizes for performance comparisons.

Scalar field visualization:

1. To compute scalar values at points which aren't grid vertices, use *trilinear interpolation*[2].
2. For slicing, consider the input to be a plane represented by its equation.

Submission

1. Format (zip following)
   (a) Directory with name <Student name>
      i. Different directory for each task named <task no.>. For example 1a, 2a etc.
         A. Code files.
         B. Videos/snapshots of outputs.
         C. Make file.
         D. Readme file explaining how to execute/use your code.
      ii. Single observation file for all the tasks.
      iii. <data> folder
         A. Corresponding data files. (If data files are too large to upload then just provide file names)
2. Strictly follow the submission format.
3. All files required to run the code should be submitted, including common util files.

Additional tasks (will not be graded)

1. Render self shadows on the protein mesh using shadow mapping technique. Refer to tutorials 23 and 24 on OGLdev tutorials.
2. Implement contour tracing alogrithm to extract isocontours, explore various seed point selection criteria.

[1] https://www.visgraf.impa.br/Projects/3dp/doc/html2/trackball_8cpp-source.html
[2] https://en.wikipedia.org/wiki/Trilinear_interpolation