E0 271: Graphics and Visualization Assignment #2

Weightage: 15%

Due: September 28, 2018

Goals

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

Data and the tasks

The data for this assignment can be found in `http://vgl.csa.iisc.ac.in/e0271/protein_scalar_data.zip`. This file contains a few protein meshes stored in OFF format, along with a corresponding 3D electro-static potential scalar field as a .pot file. The code to read these file formats is provided. **[N]** denotes the weightage of each task out of **100**.

[20] Lighting:

- **[20]** Implement Phong shading with Phong illumination model using shaders

[45] Scalar field visualization:

- **[20]** Visualize the scalar field on the protein surface and on a slice (in CPU).
  - i. Compute the restriction of the scalar field on the protein mesh. Assign colors to the mesh vertices according to their scalar values using an appropriate color map.
  - ii. Given an arbitrary plane, define a slice and compute the scalar field restricted to the slice.
  - iii. Also use the plane to slice the protein mesh into two halves. Display the slice and in alternating fashion display/hide the two halves based on user input.
- **[20]** Repeat the same task on GPU. Use 3D textures within shaders to determine the scalar values at the mesh vertices. Use 1D texture in shader for color mapping. Use geometry shaders to do slicing.
- **[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.

[35] Iso-contour visualization:

- **[15]** Given scalar value, implement marching triangles algorithm to extract iso-contour for the scalar field defined on the protein mesh on CPU. Store the extracted iso-contour in a VBO and display using standard OpenGL draw() call as GL_LINES. Allow the scalar value to be modified on the fly.
- **[15]** Extract and display iso-contour using GLSL shaders by highlighting appropriate pixels in the output image.
- **[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. Sample code can be found on the internet which can be used[1].

Scalar field visualization:

1. Each vertex of the protein surface is a point within the domain of the 3D scalar field. Use *trilinear interpolation*[2] to compute the scalar value at each vertex of the protein mesh.

2. For slicing, consider the input to be a plane represented by its equation.

Submission

1. Prepare a .zip file of the code (with makefile, readme, etc) and a text file with a description of the methods used along with the analysis and discussion of running times.

2. Submit the .zip file in VGL (room number 241, CSA) between **3:00-5:00 pm,Sep 28th**. Fix a time with the TA if you want to submit earlier.

3. Any submission beyond **5:00 pm, Sep 28th** will be considered as a **late submission**.

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 iso-contours, 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`