COMP 5893M Modelling And Animation A1

Professor Hamish Carr

October 28, 2024

Overview

This assignment is worth 50% of the overall module mark, and is intended to build your experience with geometric modelling and processing. In the first part, you will build a directed-edge data structure and use it to test a variety of surfaces to determine whether or not they are manifold. In the second, you will take an input mesh and repair it, then simplify it.

Time

The time budget for this assignment is 50 hours. It will be due at the end of Week 9. By the middle of Week 5 (the second week of lectures in this module), you will have seen the material needed for building and testing the data structure, enabling you to start the assignment.
The assignment will be marked out of 50 overall, as shown in the per question mark allocation.

Coding Standards

All code submitted must compile with the stages of compilation in the readme.txt file accompanying the code handout, at the first attempt, on the University's Red Hat lab machines. Students will normally be given exactly one chance to fix compilation issues, but this is not guaranteed.
All code submitted must be written by the student themselves, with no outside assistance of any kind, unless written approval is obtained from the

instructor in advance. Students are permitted to use the code handed out, and the following libraries: OpenGL, OpenMP, Qt and the C++ STL.
Students are expected to apply professional coding standards, including meaningful variable names, well-structured OO classes, and meaningful comments. Poor coding practices will attract a penalty of up to 20 marks out of the 50 available.

Marking Standards

When the assignments are marked, at least one data file other than the ones provided will be used for testing. Make sure your code does not make any unnecessary assumptions.

1 Task I: Directed Edges [10 marks]

You have been given a simple GUI application that reads in a geometric object stored as triangle soup and renders it on screen. Rendering is not needed for this task, but it has been provided to make your task easier.
You have also been given a set of sample triangle soup files with the suffix ".tri", where the first line is the number of triangles (n), followed by 3n triplets giving the coordinates of each vertex.
You should: a) [5 marks] Write a program called 'face2faceindex' that reads in a .tri file, computes the face index format and outputs a file with ".face" suffix in the format provided in Appendix I of this assignment.
b) [5 marks] Write a second program called '"faceindex2directededge' that reads in a .face file, computes the directed edges and outputs a file with ".diredge" suffix in the format in Appendix II.
You are expected to produce command-line utilities rather than a visual application. This is because, in the real world, you will need to convert file formats once rather than every time you load. If you wish, you may also modify the code handed out to read in a .face or .diredge file and render it directly to cross-check, but this is not required. It is recommended that you start with the smallest files provided and work your way up in size.

2 Task II: Manifold Testing [10 marks]

For each file provided determine whether it is manifold or not. If it is not, print out the ID number of the edge or vertex that fails. Report your results in a text file called 'manifold_test_results.txt'

3 Task III: The Euler Formula [5 marks]

For each mesh that you have demonstrated to be manifold, use the Euler formula to determine the genus of the surface. Add this to the '"manifold_test_results.txt' file.

4 Task IV: Mesh Repair [10 marks]

You have been provided with a file with a model of the instructor's head. However, there are some holes in the mesh, as you will discover if you run your manifold testing code on it. You are, however, guaranteed that there is only one connected component, and that no other form of mesh repair is required.

Your next task, therefore, is to repair the mesh. You should iterate through the mesh, finding unpaired edges. When you find an unpaired edge, you should use the directed edge data structure to walk around the boundary of the hole to find how many edges are involved.

You should set the position of the vertex to the average of the values of the vertices on the boundary of the hole: i.e. to their centre of gravity.

Once there are no holes in the mesh, this task is complete, and you should output the file in the same format in which you read it in. Note that the code should deal with all of the holes automatically, rather than by user input.

5 Task V: Simplification [15 marks]

You should now have a single manifold surface, and need to simplify it with the standard greedy algorithm. For each vertex, you should compute the mean and Gaussian curvatures using the Discrete Laplace-Beltrami Operator as discussed in class.

At each step, remove the vertex with smallest Gaussian curvature - this is normally one of the flattest vertices in the mesh. Remember to check for the Eulerian condition to preserve the manifold.