COMP 5821M: Geometric Processing 2023-2024

ASSIGNMENT 2: Mesh Operators & Loop Subdivision Surfaces **[30 marks total]**

We have seen in class how we can define a limit surface from a mesh by subdividing the polygons in the mesh. One of the simplest such subdivisions is the Loop subdivision, where each triangle is broken into four smaller triangles.

Your code should read in a directed edge data structure from a text file and output a new directed edge data structure in a new text file, as follows:

1. Each vertex in the input should have the same index in the output.

2. Each edge will have a new vertex in the output, added sequentially at the end of the array

3. Each face will be subdivided in 4, with the central face retaining the original index, while the remaining faces are added sequentially at the end of the array.

There are three basic approaches, which are worth different numbers of marks in order of difficulty:

A. You can generate a new triangle file & rebuild the half-edge data structure at each level. This will give you a sequence of files which represent the different levels of refinement. **[18 marks]**

B. You can create a second mesh in memory and add triangles to it one at a time based on the old mesh, then run the halfedge pairing, &c. to create the final version. **[24 marks]**

C. You can edit the original mesh ***in place****,* applying mesh operations to delete the old triangles and replace them with new. **[30 marks]**

Obviously, C is the best solution for general purpose, and is therefore worth the most marks, although any valid solution properly coded will achieve pass marks. If you adopt solution C, it would be ideal if you used the same renumbering scheme as me, to ease marking, and we will go over this scheme in the lab this week.

You will be provided with a modified triangle renderer that reads the directed edge data structure directly, to allow you to check your results. You will also be provided with an implementation of the directed edge data structure to use, although you are encouraged to use your own version instead.

Marking will be based on whether your code produces the correct mesh topology, the correct vertex positions, and has a logical relationship between old and new triangles.

All code should compile on the School's Linux machines without installation of any extra libraries or applications. You should include a makefile and a readme.txt file with any additional instructions. Other than this, you should only hand in code files (.cpp/.h).

**PENALTIES:**

Poorly structured or badly commented code may be penalised by up to 2.5 marks.

Poorly formatted output files may also be penalised by up to a further 2.5 marks.

Code without a readme or makefile may be penalised by up to 1 mark.

Code that does not compile properly will be assigned a mark of 0, but I will usually give the student one chance to correct this.

**DUE DATE:** Friday, December 7, 2023, 10:00am