COMP 5821M: Geometric Processing 2023-2024

ASSIGNMENT 3: Texture & Normal Map Synthesis **[40 marks]**

You have been given a model with colours defined for each vertex, and a set of triangles, along with a simple renderer which renders the triangles with flat shading and ignores the colours.

Your task is to convert this model to a properly textured surface and to compute a normal map, with the xyz of the normals mapped to rgb.

However, since texture coordinates have not been provided, the first step is to assign texture coordinates. As we saw in class, the best simple algorithm is Floater’s algorithm, which you will implement. This requires data to be available in patches, and we have simplified by providing a single patch in a modified object file, along with code to read it in and display it. The UI includes controls that you will need later.

**I. Build a Directed Edge Data Structure [5 marks]**

This is essentially the same as in Assignment 1, but for a mesh with a boundary – i.e. not all of the edges pass the manifold tests.

**II. Identify the Boundary [5 marks]**

Floater’s algorithm requires the boundary of the patch to be put around the edge of the patch. This requires finding an edge of the patch and following the hole around to find all of the edges.

**III: Generate Texture Coordinates. [10 marks]**

Using Floater’s algorithm, assign texture coordinates to each vertex in the mesh.

**IV: Render the texture coordinates as colours** **[5 marks]**

Modify the renderer so that enabling “uvw to RGB” allows you to show the texture coordinates on the surface as a visual check of accuracy.

**V: Render the Mesh in the Texture Domain [5 marks]**

Render the patches into the window in 2D so that you (and I) can see the mesh laid out in the plane. The wireframe checkbox will come in useful for this.

**VI: Generate a Texture Map. [5 marks]**

Render the triangles of the mesh as triangles in the texture domain to generate the texture map, both as uvw colours and as the original rgb colours, depending on the settings in the UI.

**VII: Generate a Normal Map. [5 marks]**

Compute a normal vector for each vertex, then convert to RGB and render these to a normal map.

**COMPILATION:**

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.

**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:** Thursday, January 11, 2024, 10:00am

Screenshots for your Guidance:

I have implemented the assignment – it took about 3 hours of coding and 12-15 hours of data preparation. You should end up with the ability to generate images like these:

Geometric Rendering and Wireframe:

A picture containing text, screenshot, person

Description automatically generatedA picture containing text

Description automatically generated

Texture Map with Wireframe:

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Normal Map with Wireframe:

A screenshot of a computer

Description automatically generated with low confidenceGraphical user interface

Description automatically generated