CS348a Homework assignment 4

---------------------------------------

Team:

Nora Coler

Alexandra Tamplin

---------------------------------------

**Part 1:**

**Mesh Decimation:**

**Functions:**

*initDecimation:*

Inside the for loop over the vertices of the mesh, a quadric for that vertex is calculated. Given the point, we iterate over the faces that contain our current point and calculate the face's contribution.

To calculate the contribution, we complete the formula ax + by + cz + d = 0, as defined in the paper "Surface Simplifications Using Quadric Error Metrics". <a, b, c> is defined as the normal at the point as it is orthogonal to the point. d is calculated as the negative dot product of the normal and the point.

With these values <a, b, c, d>, we create a Quadricd qi and add it to the quadric for the vertex. In the end, the vertex quadric is the sum of the quadrics of its faces.

*priority:*

To calculate the priority of a half edge, we first sum the quadrics from its two vertices (to and from vertites). We then return the calculation v.transpose(Q)v as our error. For simplicity, we are using only the calculation as a result from the to vertex.

*decimate:*

For the iteration, the number of iterations is \_n\_vertices. In a while loop that continues until the number of vertices is equal to \_n\_vertices, the verties in the queue is examined. We find the half edge that originates at that vertex. If that half edge is legal to collapse, we break out and collapse the edge. The quadric from the to vertex is added to our current vertex (the from) before collapse.

Once the half edge has been collapsed, the vertices from and including the current vertex are requeued and the number of vertices is decreased by one.

**Images:**

|  |  |  |
| --- | --- | --- |
| Original mesh | Mesh with 50% vertices | Mesh with 10% vertices |
| armadillotri100.png | armadillotri50.png | armadillotri10.png |
| bunnytri100.png | bunnytri50.png | bunnytri10.png |
| homertri100.png | homertri50.png | homertri10.png |
| horsetri100.png | horsetri50.png | horsetri10.png |
| humantri100.png | humantri50.png | humantri10.png |

---------------------------------------

**Part 2:**

**Suggestive Contours**

**Functions:**

*isSilhouette:*

We obtain the two halfedges from the handle e. Then the view ray for the vertex from the first halfedge is calculated by subtracting the cameraPos from the vertex position.

With the view ray, we are able to calculate the dot product of it with the normal of each halfedge. If the products have different signs, the edge is a silhouette.

*isSharpEdge:*

We obtain the two halfedges from the handle e and their respective normals. If the dot product of those two normals is less than .5, the edge is a sharp edge.

*computeViewCurvature:*

*renderSuggestiveContours:*

**Images:**

---------------------------------------

**Part 3:**

**Additional Feature: Highlights**

**Functions:**

**Images:**