COSC 4370 – Homework 5

Name: Grishma Vemireddy

November 2024

1 Objective

In this assignment, we will implement two classic mesh algorithms: Loop Subdivision and QEM Simplification. The goal of this assignment is to smoothly subdivide and simplify the mesh.

2 Methods

This assignment has two tasks: loop subdivision and simplification using Quadric Error Metrics. For loop subdivision, we are to modify ComputeNewEdgePoints(), updateOldVertices(), and buildNewFaces() functions in subdivide.cpp for task 1. For task 2, we have to modify calcDeltaV(), updateQudraticError(), updateCost(), buildAdjacency(), and QEM_Simplify() in simplify.cpp and updateFace(), computeQuadricMatrix() functions in common.cpp
Reference material: https://graphics.cs.wisc.edu/Courses/559-f2010/pubs/pub_RTR-Subdivision.pdf, https://graphics.stanford.edu/~mdfisher/subdivision.html and Surface Simplification Using Quadric Error Metrics

3 Implementation

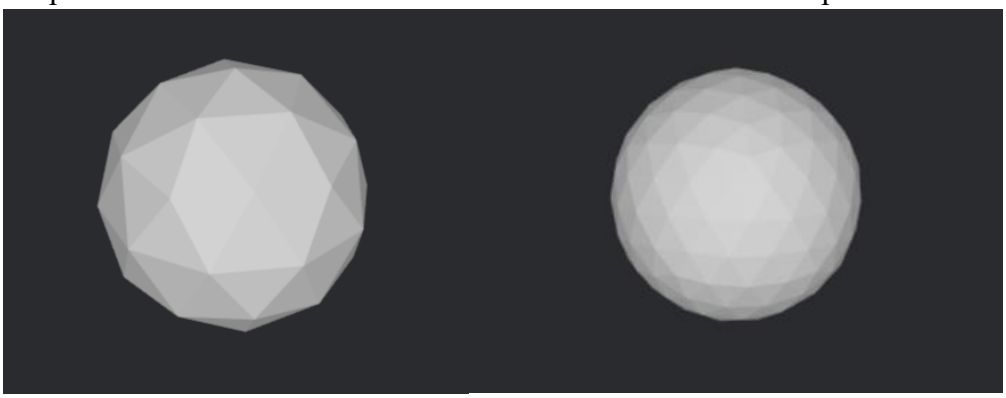
Task 1: Loop subdivision is a subdivision scheme for triangles which updates each existing vertex and creates a new vertex for each edge. Each triangle is subdivided into four new triangles. To create a new edge point, I used the following formula $P_i^{k+1} = \frac{3p^k + 3p_i^k + p_{i-1}^k + p_{i+1}^k}{8}$ in ComputeNewEdgePoints() newEdgePoint.x = (3.0 / 8.0) * (v1.x + v2.x) + (1.0 / 8.0) * (ov1.x + ov2.x), and since it is 3d computed for x, y, z. For boundary edges, $P_i^{k+1} = \frac{1}{2}(p_1 + p_2)$ as there are no opposite vertices. To compute new positions for old vertices, I get the number of neighbors of the vertex, then using Warren's method to compute the beta so that if $n > 3$ $the\ beta = 3/8n$ else beta is 3/16. Then I compute the weighted average of the neighbor positions after which I update the vertex position and append it newVertices[i] = newVertex. This is my implementation of updateOldVertices() function. To implement buildNewFaces() function,

iterate through the faces and find the three vertices and their corresponding new edge points by e1 = (*edges.find(Edge(v1, v2))).new_edgepoint_id, then update mesh by adding the faces to the subdivided mesh by using push_back(Face({ v1, e1, e3 })). This would be the implementation of loop subdivision.

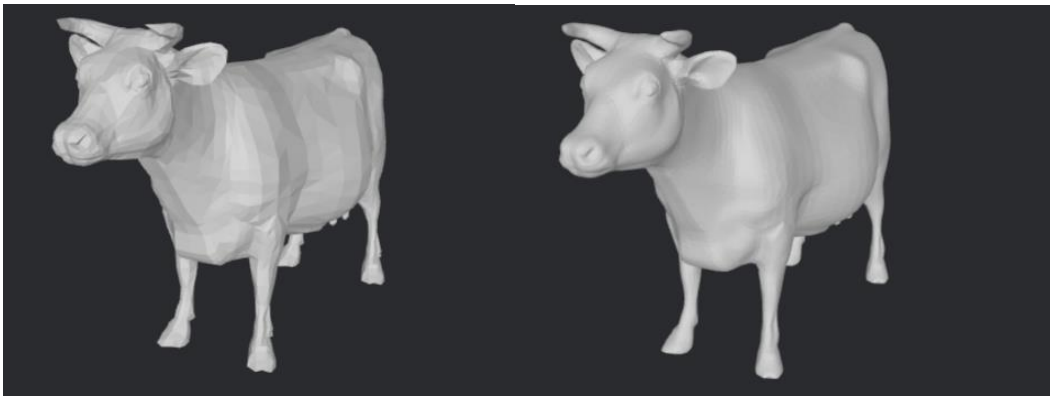
Task 2: Simplification using Quadric Error Metrics (QEM) the algorithm implementation is found in the paper https://www.cs.cmu.edu/~./garland/Papers/quadrics.pdf. For clacDeltaV() function, in section 4 of the paper $\Delta v = v^T Q v$. The quadric error matrix Q is updated by adding the all the faces adjacent to the vertex for updateQuadricError() function. The updateCost() is computed $\Delta v = v^T Q v \; where \; Q = Q_{v1} + Q_{v2} \; and \; v = \frac{v_1 + v_2}{2}$. In buildAdjacency() function, I am to update Q of v1. In QEM_Simplify() function, I am to update quadric error of affected vertices. In common.cpp, in the updateFace() function after a vertex merge and computeQuadricMatrix() given a equation $ax + by + cz + d = 0$, a 4x4 matric is constructed as $k_p k_p^T, with \; k_p = [a \; b \; c \; d]^T$. I could not complete this portion of the assignment.

4 Results

These are the screenshots of the .obj files. I was unable to complete and write the code for simplification task so there is no screenshot of the result for simplification.



Right: Original sphere Left: Subdivided sphere



Right: Original cow Left: Subdivided cow