

Final Project - Proposal

For my final project, I have decided to implement the algorithms described in “Surface Simplification Using Quadric Error Metrics”. In the following paragraphs, I will describe in general the process I think I will be using to implement these algorithms using OpenGL.

The input for this process is fairly straightforward - the bunny.off file that has been used in both of our past projects. A pair of matrices will keep track of the locations of each vertex and the indices of the vertices that make up each face, respectively.

From here, we will find the 4x4 matrix Q for each vertex that helps represent the cost of removing said vertex. The matrix Q is equal to the sum of all “fundamental error quadrics” K for that vertex. A single K exists for each face that the vertex is a part of and is defined by a matrix $[a \ b \ c \ d]$ multiplied by its transpose. a , b , c , and d are given by the equation of the plane that the face coincides with in the form $ax + by + cz + d = 0$.

At the same time, the program will determine what pairs of vertices can be considered for contraction (effectively combining the two into a single vertex). This is the case for any vertices that share a side, but also for vertices that are particularly close to one another (the exact distance is not something I have decided on, yet).

The program then decides where each pair of vertices would contract to. The exact process is described in section 4 of the original paper, as it is a bit lengthy to include here, but effectively involves trying to minimize the error when possible. The cost of this new vertex can now be calculated using its position and the Q values of the original vertices. With each cost now calculated, the contraction that has the least cost is executed. All possible vertex pairs that involved these two vertices are updated to reflect a new contracted position and cost. Repeat this process of removing vertices until we have reached the desired number of remaining vertices.

As per the expectations of this project, it should be possible to export the updated image currently displayed on the screen as an object file, and also to be able to update the requested number of vertices on the fly without restarting the program.