Simplification Using Quadric Error Metrics

Mesh simplification algorithms transform a polygonal mesh into another mesh with fewer vertices, edges and faces. The approximation produced generally satisfies a user defined criterion such as target face count or a maximum tolerable error.

Iteratively contracting edges is an efficient and commonly employed simplification approach that preserves mesh topology. It associates a cost of collapse with each edge that determines the contraction to perform during each iteration. To estimate this cost, Garland et al. [1] compute the error quadric Q for each original vertex v by summing over all triangles directly adjacent to v:

$$\mathbf{E(v) = \sum_{N(v)} v'^T Q_i v' = v'^T\left(\sum_{N(v)} Q_i\right)v' = v'^T Q v'}$$

Here v' equals $[v_x\ v_y\ v_z\ 1]^T$, $v'^T Q_i v'$ represents the squared distance of vertex v from the supporting plane $p_i = [a\ b\ c\ d]^T$ of an adjacent triangle and Q_i equals pp^T. If an edge between two vertices v1 and v2 is to be collapsed, the error quadric for the new position r for v1 is given by the vector sum $Q_r = Q_{v1} + Q_{v2}$. If Q_r is invertible, r is found by minimizing $v'^T Q_r v'$. If Q_r is not invertible, r is chosen to be the position of v1, v2 or (v1 + v2) / 2, whichever has the least error.

The algorithm to iteratively collapse edges can be summarized as follows:

1. Compute Q_i for all mesh vertices
2. Associate a new vertex position r and cost of collapse E(v1 -> r) with each mesh edge
3. Place all the edges in a min heap keyed on cost of collapse
4. Iteratively collapse edges from the heap. Change v1's position to r and update the cost of all edges with v1 as a vertex.

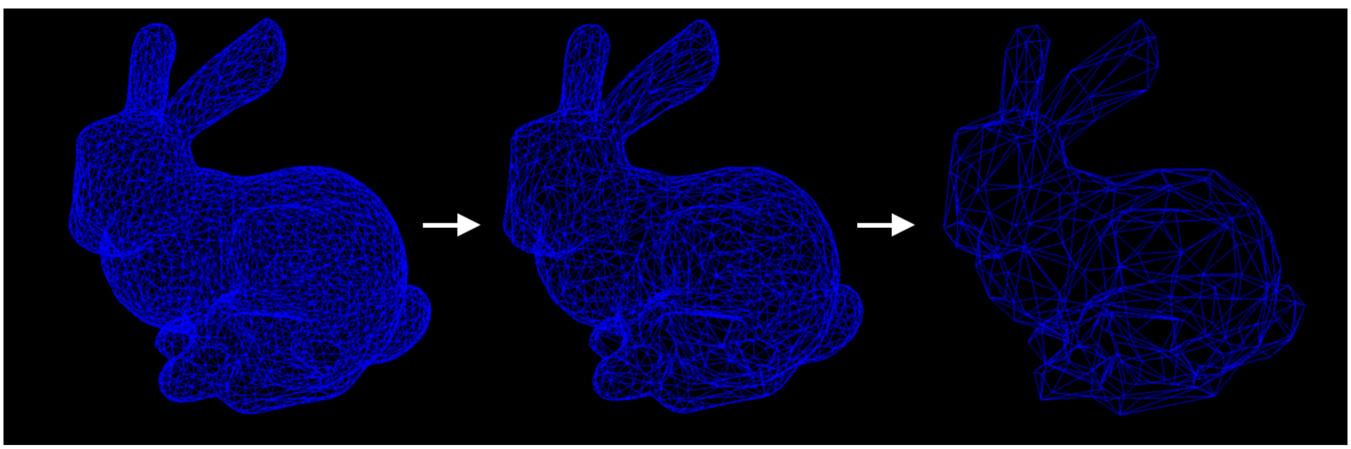

Figure 1: *Left*: Original Mesh. *Middle*: 50 % reduced. *Right*: 90% reduced

Implementation: https://github.com/rohan-sawhney/simplification

[1] Garland et al. Surface Simplification Using Quadric Error Metrics