

SRGGE - Assignment 1

Diego Campos Milian

April 23, 2020

1 Model instancing and framerate

The application allows to visualize $N \times N$ copies of the selected model in a grid via instancing. To change selected the N value change the value in NumElements. Also, is possible to see the framerate at the right bottom corner of the screen, below Vertices. Notice that, due to the fact that QT only perform screen updates after an action, like moving the camera or changing any values, this will not reflect the maximum possible framerate, unless you perform enough actions per second.

2 Vertex Clustering

The vertex clustering implemented is created using an octree, therefore the application provides the possibility to view five different levels of details of the mesh. from depth 2 to 6 of the octree. Is also possible to see the entire octree, though this can create a several downside in performance, or just the depth level selected, root by default.

Note that, though the octree construction is done on loading, the mesh reconstruction is done in run time when a different LOD level is selected, therefore meshes with higher faces count will take more time to reconstruct.

2.1 Octree construction

Both the octree and node classes implementation can be found at `octree.h/.cc`. Each node n is compose of the subsequent variables:

- Position of the cell n_p .
- Cell size n_{size} .
- List of childs n_{childs} .
- List of vertex indices contain in the cell n_{verts} .
- Position of the new vertex resulted from clustering.
- Normal of the new vertex.

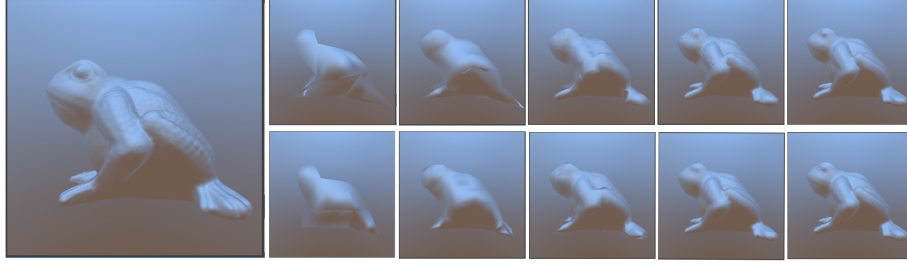


Figure 1: At left original mesh. At top same mesh using quadric error metrics with different LOD levels and at bottom the same but using the mean.

Vertex clustering is performed as the octree is constructed, therefore in each recursive iteration parent nodes n creates its childrens $n_{childs}(i)$ ($i = 1..8$), unless the stop rule is met, and assign to them the vertices contained in n_{verts} that are in its child cell space. Then, for each child the new vertex position and normal are calculated as the average of the same of each vertex index.

Since, with this implementation each node only receives part of the vertex index list of its parent, not the complete, and subsequently for its childrens, the computational cost of it should be $O(N_v * \log(N_v))$, where N_v is the number of vertices.

2.2 Mesh reconstruction

The implemented mesh reconstruction algorithm is based in the idea that for all the triangles faces of the given mesh the ones that should only remain for a certain level of the octree will be those that each of its three vertices are contained in different cell/nodes, with the only change that those vertices will be changed for the ones resulted of the vertex clustering of the cell they are in.

2.3 Quadric error metrics

The quadric error metric implementation can be found inside the `SetLodLevel(lvl)` function in the `GLWidget` class. The implementation is the proposed by Garland et al. where the new vertex resulted of clustering is calculated using all the planes present in a cell. A plane is considered inside a cell whenever any of its three vertices is contained in it.