

EECS 267 FINAL PROJECT - README.PDF - Yusuke Niiro

//Description

In this project, I implemented “Mesh Simplification” by using C++.

< Key Board >

You can change the number of vertices that you want to clip by clicking certain button.

If you click “a”, it will decrease the number vertices that will be clipped, so it will increase the number of vertices in your object. On contrary, if you click “s”, it will increase the number vertices that will be clipped, so it will decrease the number of vertices in your object.

//Processes

- 1) Load the object from OBJ file.
EX) sphere.obj, cow.obj
(Compute the number of vertices/faces and put into ver_num/face_num.)
- 2) Construct the edge array by using face in that object. Additionally, there will be duplicates by getting edges with this method so remove the duplicates to make each edge only one.
- 3) Calculate the error for each vertices and sort it in order, minimum to maximum.
Because, it will be used when you remove the vertices that have less error.
In this program, error will be the maximum distance to the adjacent edges.
- 4) Create two models besides the original model.
One will be using to render the object faces that was not made from removing vertices.
And another will be using to render the faces that will modify the object. (You will need this cover the hole that was made by removing the vertices. Use the triangles that makes small differences.)
- 5) Render the two object that was made in process 4.

//Evaluation

Compare the Constructing Time by changing the condition of how many vertices you remove from the object. In this evaluation, I used cow object with 5 conditions (0/500/1000/1500/2000/2500). Average is from 30 times of result.

I used the `gs_time()` function to calculate the constructing time. I put this function in the main file.

Constructing time will show up if you run the code.

The results are in the Table 1.

Table 1: *Average Constructing Time for *Cow Object

Number of Vertices that was removed	Average Constructing Time (sec)
0	0.355973367
500	0.485137933
1000	0.595540733
1500	0.766241033
2000	0.961357867
2500	1.230834333

*Average Building Time: Average of 30 samples

*Cow Object: 2903 vertices / 5804 faces

By looking at the table, I figured out that computing time for getting vertices that has little error, removing process and getting the triangles that cover the hole effects the constructing time.

If you remove the vertices, it means that the number of triangles that will be rendered in the scene will be decreasing as well.

So, you can say that computing process effects more than rendering process.