UBC CPSC 524
Computer Graphics Modelling

Final Project Report

Minsi Sung (44103232)

Mechanical Engineering

University of British Columbia

Date:2019/04/24

Instructor's Name: Alla Sheffer

1 Introduction

Remesh is the process to generate the new mesh which has better mesh quality and mesh regularity. It can be done in two ways. First is to segment the mesh surface into parameterizable pieces in 2D and project back to 3D mesh after remeshing. It's called "Global Remeshing". Second way is "Local Remeshing" which modify existing mesh using sequence of local operations. Because "Local Remeshing" is easier to implement, it is chosen to be the remesh method in this report.

2 Technical background and methods

Mario Botsch Leif Kobbelt[1] proposed the Direct Surface Remeshing. It avoids global parameterization which is numerically very sensitive and also avoid the computational expensive local parameterizations by using local operators and back –projections process. The main goal of this method is trying to equalize all the edges length and area. Given a target edge length L (here we use average edge length of the original mesh), and perform the following steps:

1. Split all the edges to their midpoint if they are longer than $\frac{4}{3}L$.
2. Collapse all the edges to their midpoint if they are shorter than $\frac{4}{5}L$
3. Flip edges to minimize the deviation from valence 6.
4. Shift the vertices on the surface by tangential smoothing

After a few of these cycles (about 5) we obtain a triangulation whose edges have length close to L and whose vertices have valence close to 6.

For the first step: Split Edges, the L_{max} is $\frac{4}{3}L$ because the difference between L_{max} and the target edge length ($|L_{max} - L|$) and the difference between $\frac{1}{2}L_{max}$ and the target edge length L ($\left|\frac{1}{2}L_{max} - L\right|$) are the same. Same as the second step: edges collapse operator. Therefore, after the iteration, the edge length deviation can be minimized.

For the third step: Edge Flipping, we not only need to minimize the valence excess $\sum_{i=0}^{n}(valance(v_i) - 6)^2$, also need to check the normal error of the triangle which the target flipping edge is on and the smoothness of the mesh after flipping.
We flip the edge only when the valence excess is lower after flipping and not violate the normal error and smoothness criteria.

The normal error criteria is that if the minimum of $N_{v_i} \cdot N_{v_{i+1}} < \cos\theta_{gap}$, then don't flip this edge. N_{v_i} is the normal vector of the vertex in original input mesh (Figure 1 and Figure 2) .

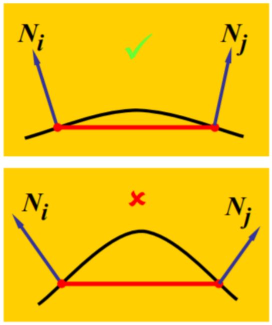


Figure 1

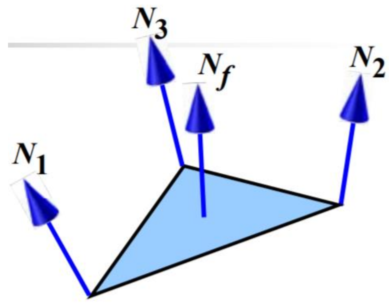


Figure 2

The smoothness criteria is that if the minimum of $N_f \cdot N_{v_{i+1}} < cos\,\theta_{smth}$, then don't flip this edge. N_f is the normal vector of the triangle face in original input mesh (Figure 2). If we flip the edge without smoothness criteria, it may create the unsmooth mesh like Figure 3.

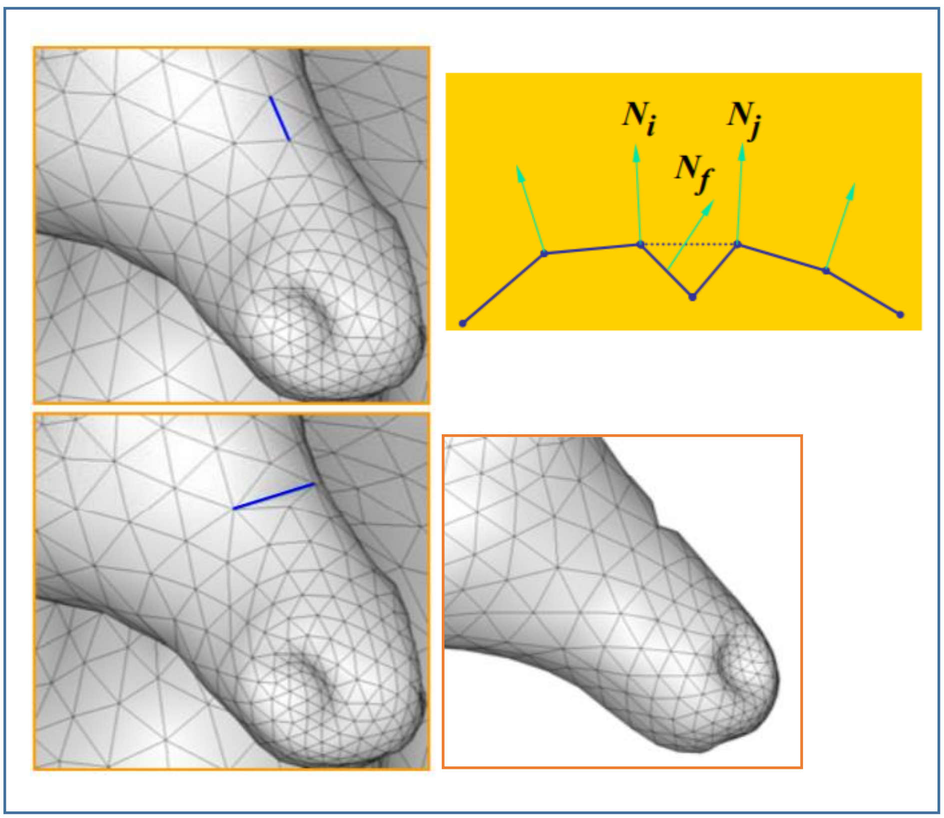


Figure 3

For the fourth step: Vertex Shift, each vertex p is assigned a gravity that equals its area A(p). A tangential smoothing process moves each vertex pi to its gravity-weighted centroid

$$g_i = \frac{1}{\sum_{p_j \in N(p_i)} A(p_i)} \sum_{p_j \in N(p_i)} A(p_i)\, p_j$$

In order to make sure a tangential smoothing on the surface, the update vertices need to be projected back to the tangent plane of p_i :

p_i'= $g_i - n_i n_i^T (g_i - p_i)$
Where n_i is the normal vector of the vertex p_i

Note here that we need to restrict the vertex shift vertex in order to avoid change the geometry. So, if the distance between the original vertex position and new position is large than K×H, then don't shift this vertex. K here is a constant that user can adjust, in this report, I use 0.6. H is the average length of all the edges connect to the vertex p_i.

The vertex shift iterate around 5 times for each remeshing local operations iteration suffice to reduce the total variance of vertex areas.

3 Results and discussion

The first input mesh to test my code is in figure 4 which has 5804 triangles. The iteration number of the local operations is 5 and each iteration contains 5 times vertex shift iteration. It takes around 4 seconds to finish remesh. The mesh after remesh is shown in figure 5.
The second test input mesh in figure 6 has 49980 triangles. The iteration number of the local operations is 5 and each iteration contains 5 times vertex shift iteration. It takes around 15 seconds to finish remesh. The comparison between before and after remesh is shown in figure 7.

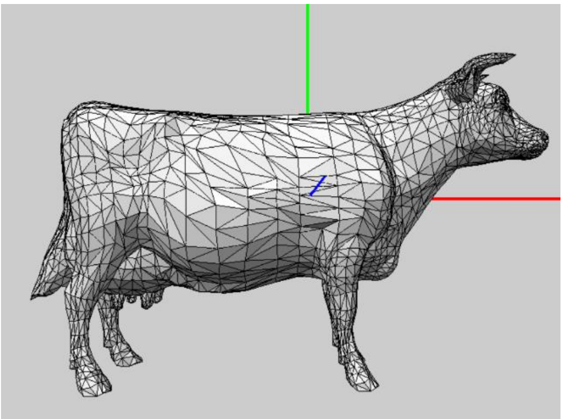


Figure 4

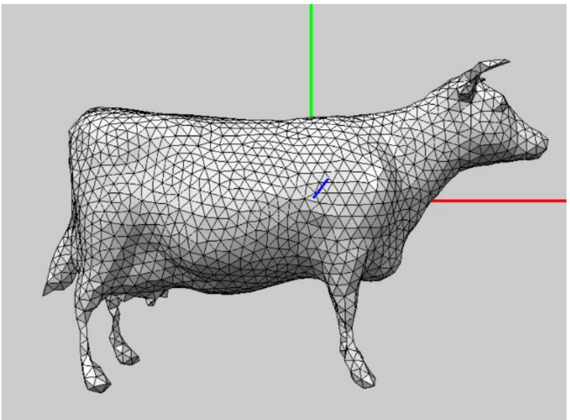


Figure 5

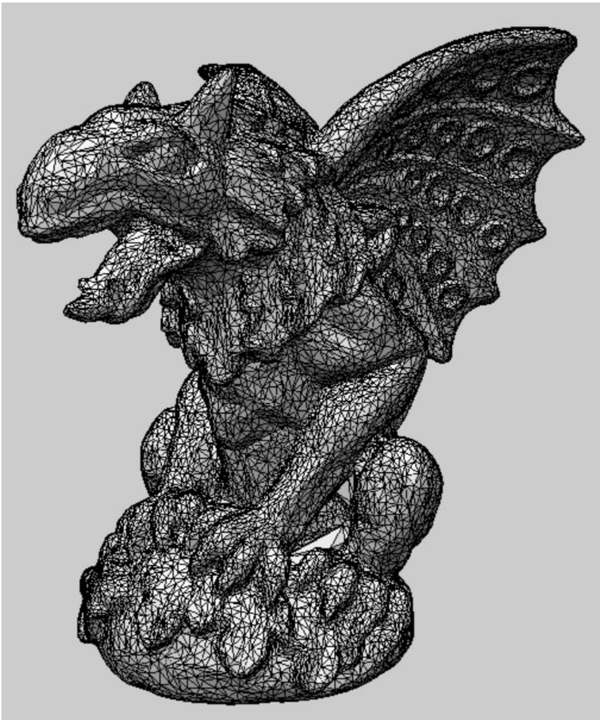


Figure 6

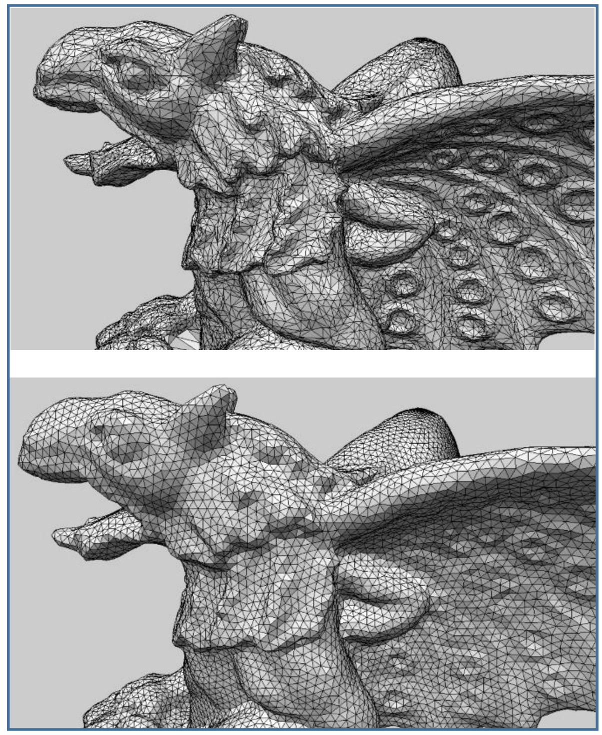


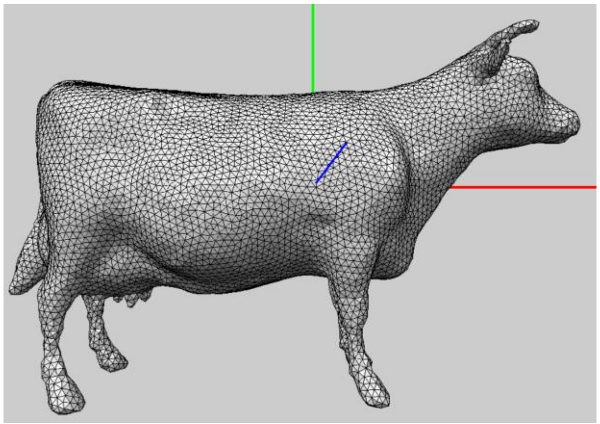
Figure 7

From the results and comparisons above we can observe that the remesher I implemented in this report can decrease the deviation of edges length and areas. But the problem here is it erases the details of the original mesh and shrink a little bit. The erasing details problem is because we are trying to make all the triangles has the same area and edge length even for the high curvature area.

The solution of this problem is to use the smaller target edge length to keep all the details, but it would cause too many redundant triangles in the smooth area and waste lots of time and memory (figure 8 and figure 9). The cow model takes12 second and Gargoyle model takes 90 seconds by decreasing the target edge length to 0.4 times of average edge length of original mesh.

This problem can also be solved by use the adaptive remesher to do the remesh depending on the mesh curvature.

The shrinking problem may because of the projection to the tangential has some bug inside. With the smaller target edge length, shrinking problem can also be fixed.



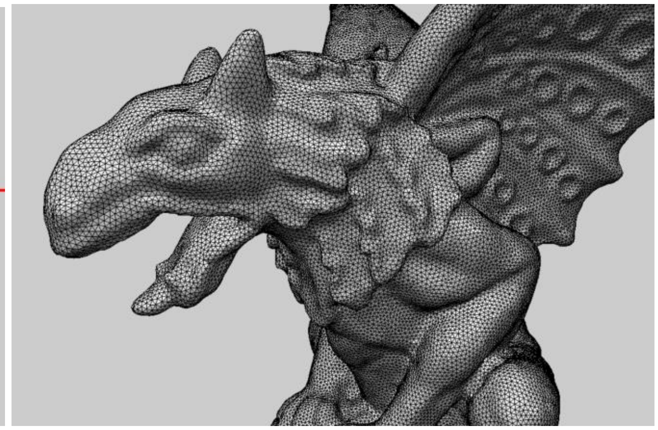


Figure 8

Figure 9

The other problem is that after the remesh using my remesher, there are some self-intersection like figure 10. But the test self-intersection is too expensive, it’s usually be ignored.

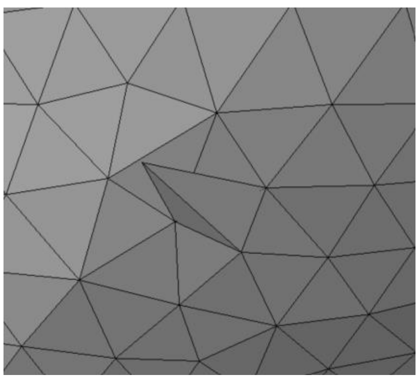


Figure 10

4 References

[1]
Mario Botsch Leif Kobbelt, "A Remeshing Approach to Multiresolution Modeling," Computer Graphics Group RWTH Aachen University.
[2]
Alla Sheffer, " Computer Graphics Modelling Course Slides," Univeristy of British Columbia