CPSC 524 Project Report (Remesher)

Xindong Lin

1 Motivation and Goals

Remeshing is a process of bringing a badly formulated mesh to a well-defined mesh that can be easily used and processed by other platforms or algorithms [Alliez et al., 2008] or more efficiently keeps shape features[Alliez et al., 2003a]. In this project, I implemented an isotropic remeshing method [Alliez et al., 2003b] which focuses more on creating equilateral triangles, which can improve the numerical stability of other algorithms. In the family of isotropic remeshing, parametrization-based methods are compelling due to the simplicity of connecting Delaunay triangles from central Voronoi tessellation in 2D. However, building each part of the parametrization-based methods like parametrization [Lévy et al., 2023, Sheffer et al., 2005], Voronoi tessellation[Alliez et al., 2005, Zheng and Tan, 2020], Lloyd relaxation, Delaunay triangle connections [Qi et al., 2012], and back-projection to 3D needs careful consideration in detail and optimization. This amount of engineering work cannot be done within the scope of the course project so I decided to choose another isotropic paper that directly remeshes the surface[Botsch and Kobbelt, 2004] to implement.

2 Algorithms

In the paper[Botsch and Kobbelt, 2004], I decided to implement the 4 steps in Section 4: Area equalizing remeshing. Also in this document(page 51), there is a good explanation of the 4 steps for my reference.

Given an edge length l, perform

1. Split all edges at their midpoint that are longer than $\frac{4}{3}l$
2. Collapse all edges shorter than $\frac{4}{5}l$ into their midpoint
3. Flip edges in order to minimize the deviation from valence 6 (or 4 on boundaries).
4. Relocate vertices on the surface by tangential smoothing.

Selecting l: The target edge length l for the remeshing is chosen to be slightly less than the average edge length of the original surface.

Tangential smoothing: for each vertex p and its area $A(p)$, a tangential smoothing process moves each vertex p_i to its gravity-weighted centroid

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

The updates of p_i become

$$p_i \leftarrow p_i + \lambda (I - n_i n_i^T)(g_i - p_i)$$

The rest operations like edge split/collapse/flips are more straightforward and can be found in the aforementioned document.

3 Tools to use

This remesher is built on top of minimesh. I also found a good resource, which is used for debugging purposes.

4 Results

4.1 Experiment Setup

For each remeshing process, I choose the target length l to be 0.98 of the average edge length. I also added **boundary conditions** so that this remesher can handle both open and closed meshes. The above algorithm is executed 4 iterations as instructed in the paper.

4.2 Mesh Data

To compare to the original paper's result, I found the *max-planck.obj* from this website. I also tested the remesher on some other meshes provided in previous assignments. The tested meshes contain both open and closed ones.

4.3 Compile Instruction

The code can be compiled and run with the exact same commands in cmake-examples.txt. The submission doesn't contain any mesh or third-party library. Those can be found in the original minimesh repository. Please do compile and run it as I think a live demo is really cool to play with.

4.4 Runtime

The remesher runs about 1.5s for max-planck (99991 faces) on an AMD Ryzen 7800X3D CPU.

4.5 Qualitative Results

See the figures 1 to 7. We can see that triangles after remeshing have similar areas. However, this isotropic method doesn't keep the sharp features very well. For Fig.4, 5 and 7, the sharp features are lost due to the smoothing. I noticed that the sharp features will lose more especially for low-resolution meshes and also for meshes that have uneven distribution of triangles.

5 Ablation Study

In the ablation, I separated each step in the algorithm to see how they affect the mesh. See Fig.8. I also tried only using 1 iteration versus 4 iterations, see Fig.9.

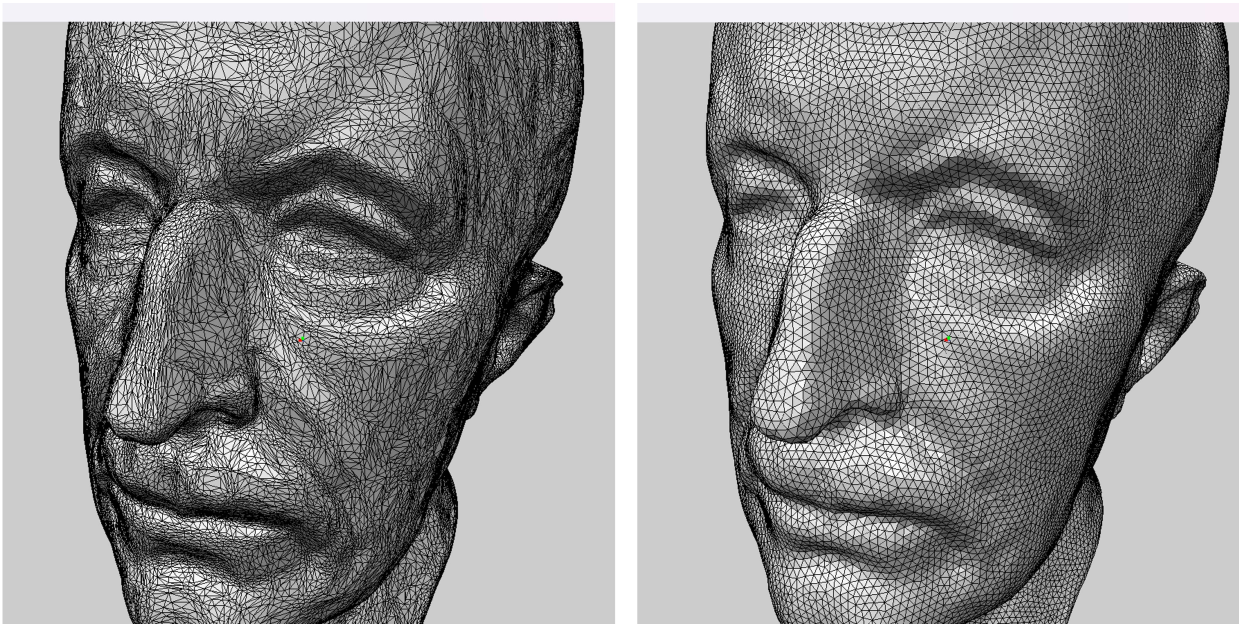

Figure 1: Left: Original, Right: Remeshed

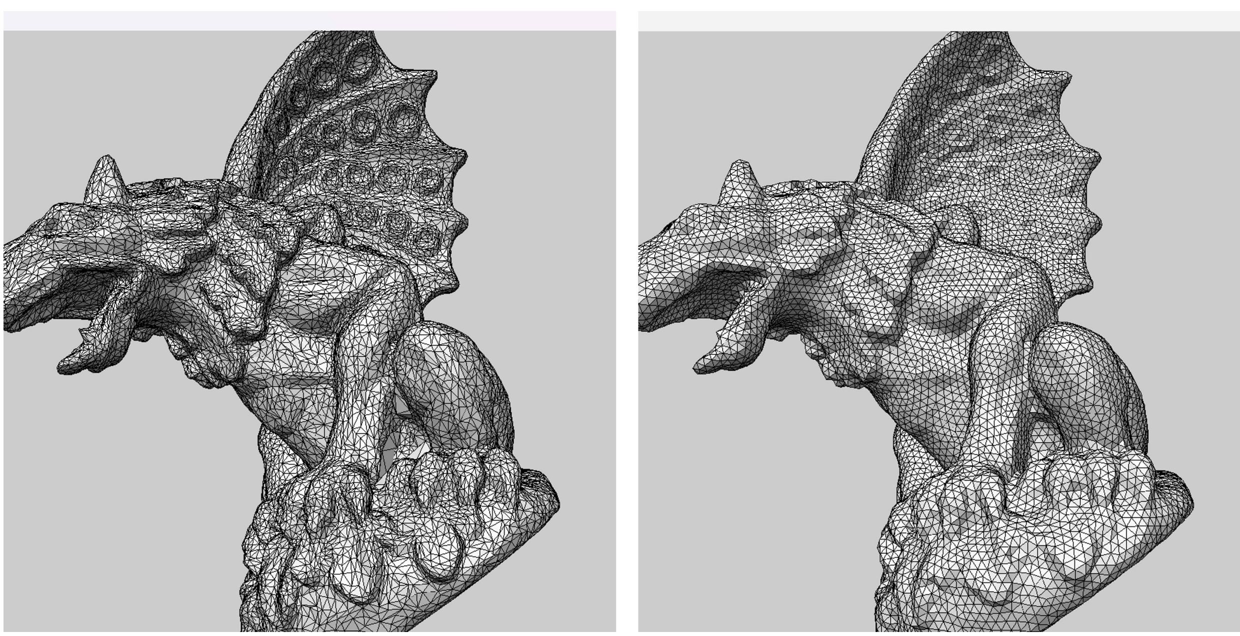

Figure 2: Left: Original, Right: Remeshed

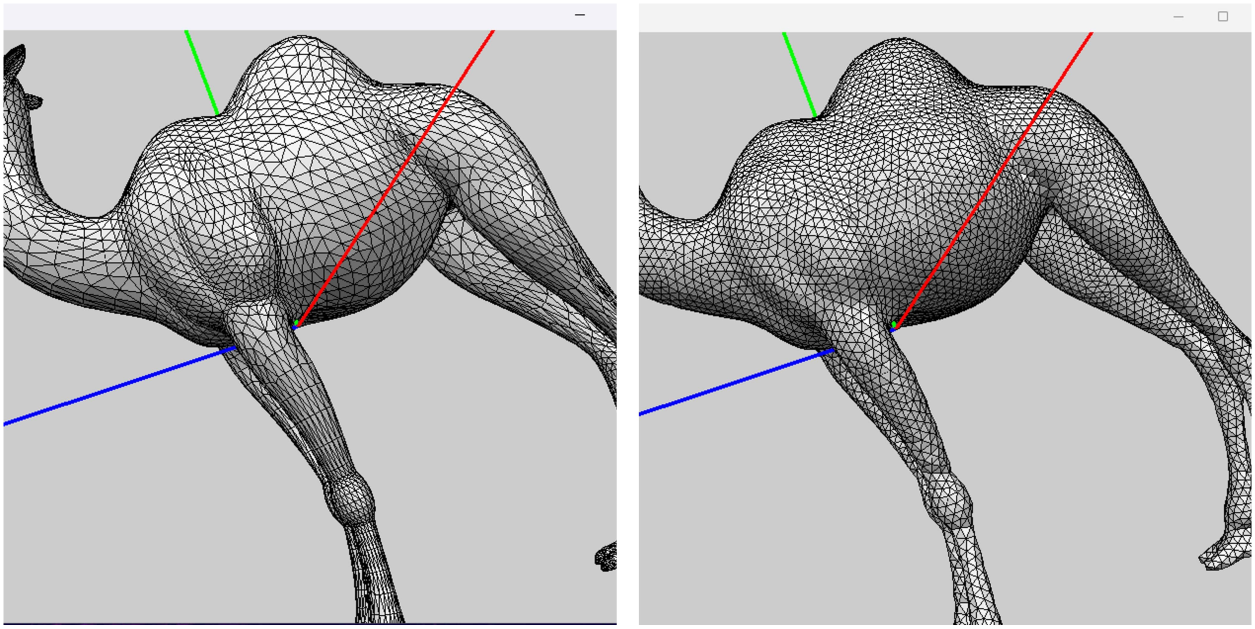

Figure 3: Left: Original, Right: Remeshed

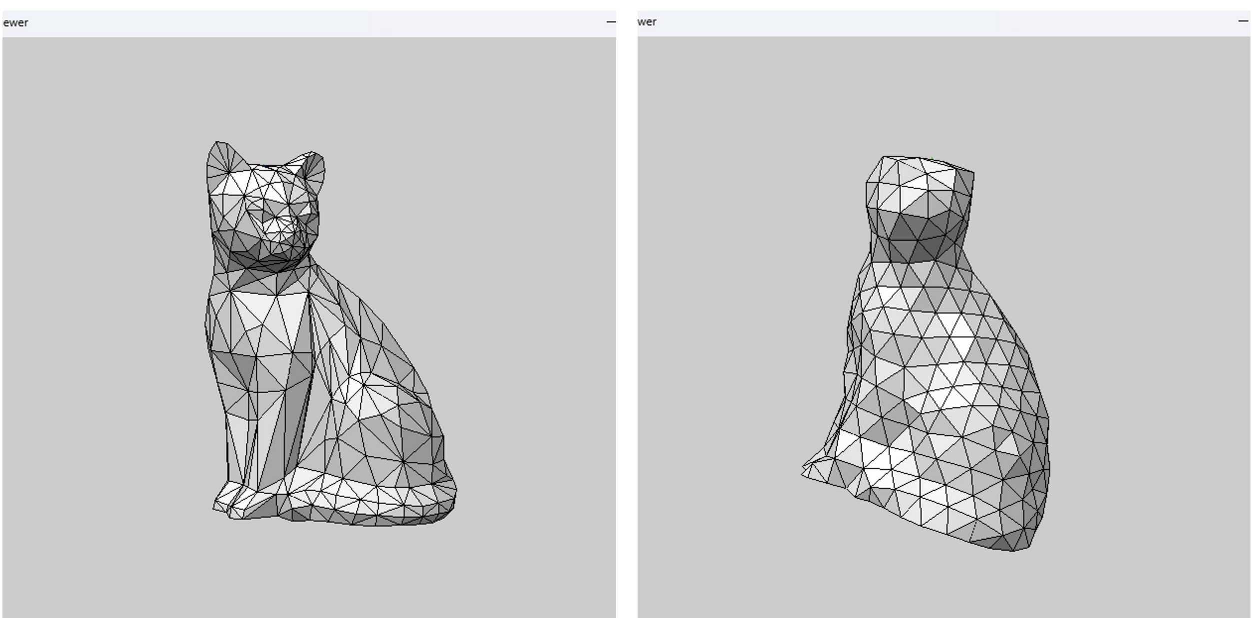

Figure 4: Left: Original, Right: Remeshed

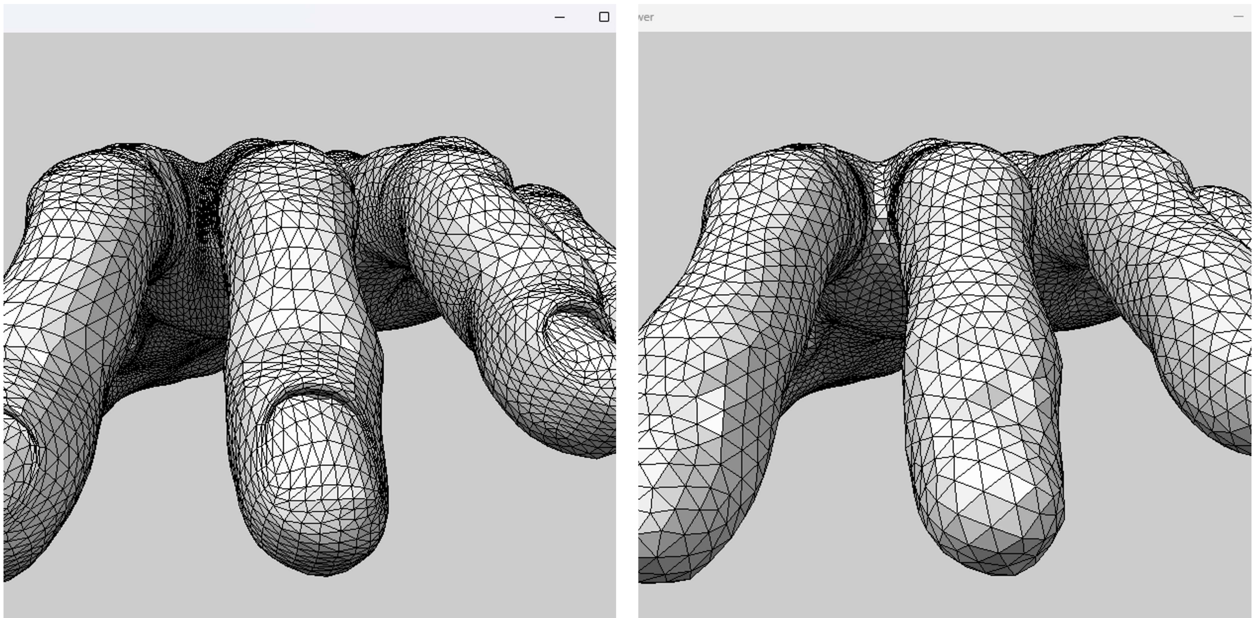

Figure 5: Left: Original, Right: Remeshed

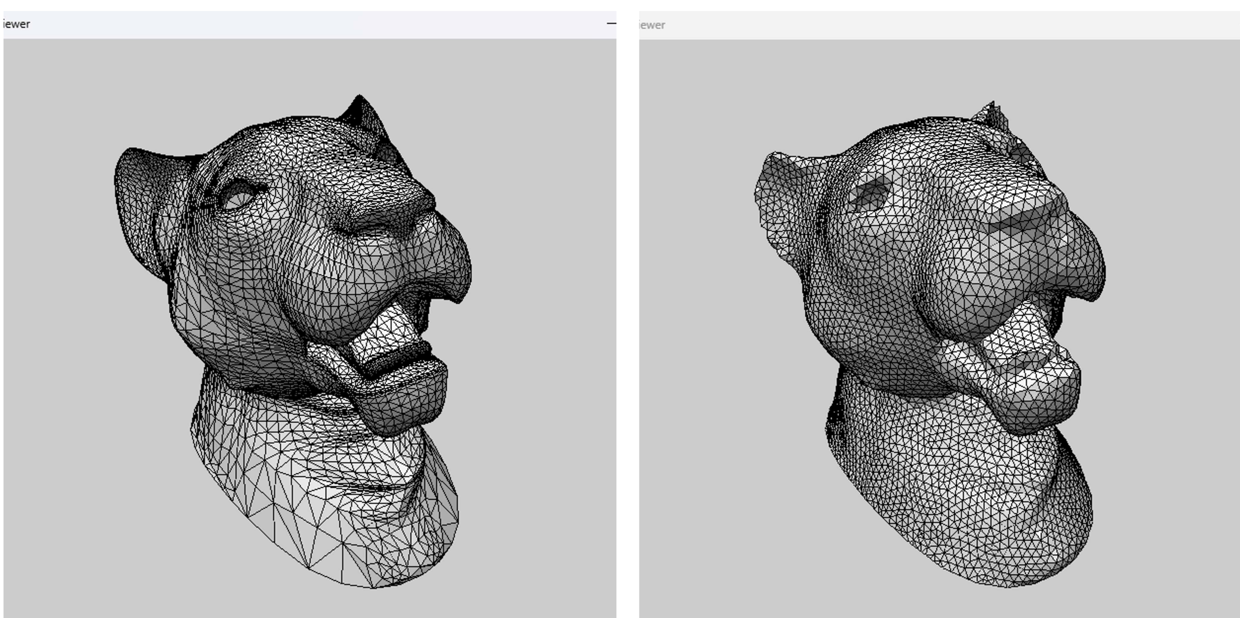

Figure 6: Left: Original, Right: Remeshed

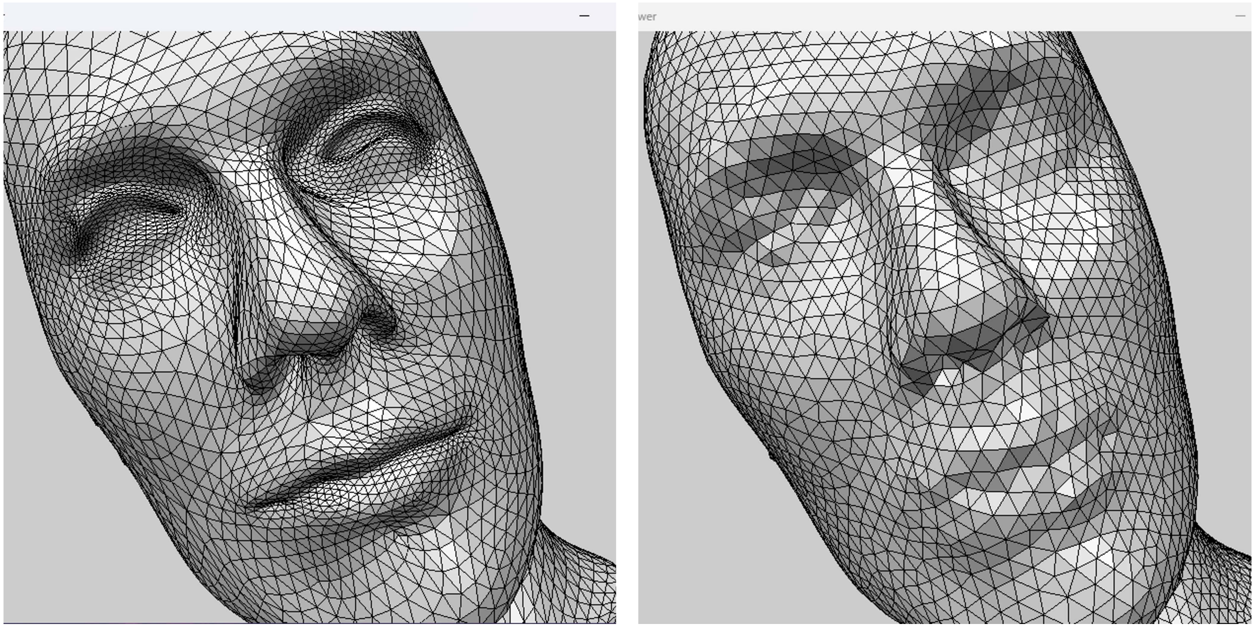

Figure 7: Left: Original, Right: Remeshed

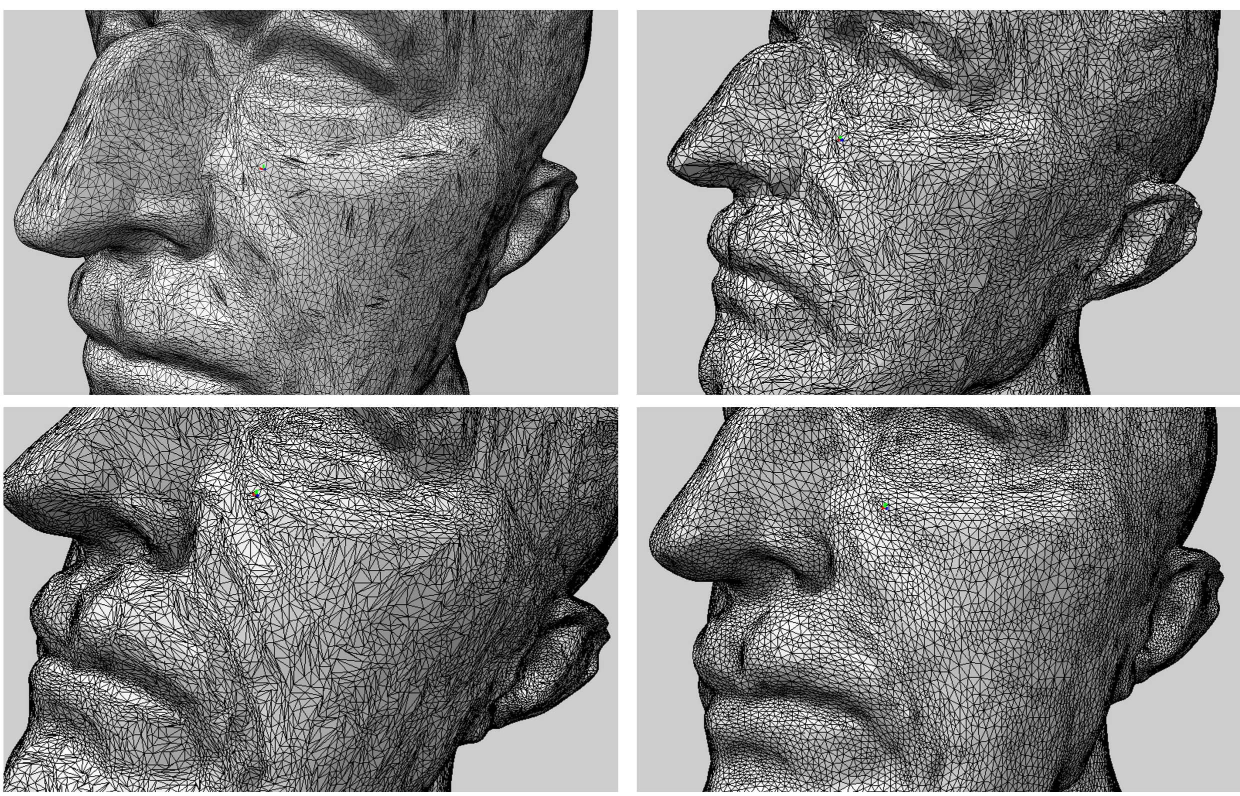

Figure 8: Top left: Only split long edge, Top right: Only collapse short edge, Bottom left: Only flip edge, Bottom right: Only tangential shift

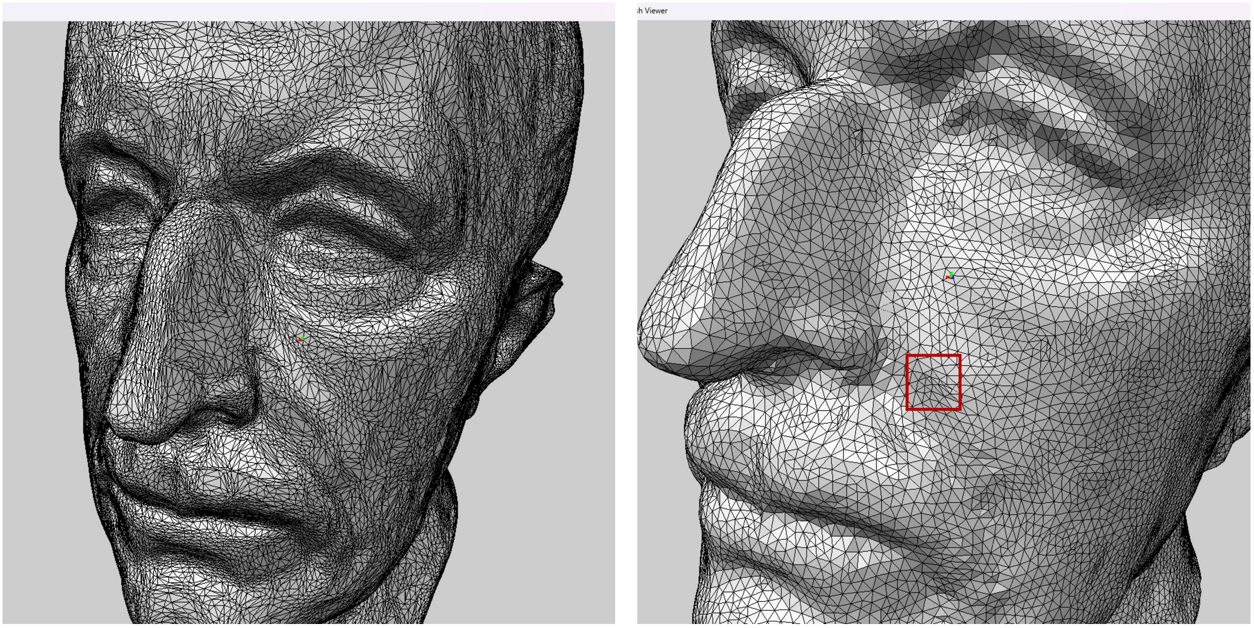

Figure 9: Left: 4 iterations, Right: 1 iterations

6 Conclusion

In this project, I implemented an isotropic remesher that aims to equalize the area of each triangle and distribute them evenly. The method splits long edges at their midpoints, collapses short edges at their midpoints, flips edges to let vertex valences close to 6 (boundary 4), and shifts vertex to their centroids using tangential smoothing. The method will suffer sharp feature loss, especially for low-resolution areas. However, it improves the numerical stability during other DGP algorithms.

References

[Alliez et al., 2003a] Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., and Desbrun, M. (2003a). Anisotropic polygonal remeshing. In *ACM SIGGRAPH 2003 Papers*, pages 485–493.

[Alliez et al., 2005] Alliez, P., De Verdière, É. C., Devillers, O., and Isenburg, M. (2005). Centroidal voronoi diagrams for isotropic surface remeshing. *Graphical models*, 67(3):204–231.

[Alliez et al., 2003b] Alliez, P., De Verdire, E. C., Devillers, O., and Isenburg, M. (2003b). Isotropic surface remeshing. In *2003 Shape Modeling International.*, pages 49–58. IEEE.

[Alliez et al., 2008] Alliez, P., Ucelli, G., Gotsman, C., and Attene, M. (2008). Recent advances in remeshing of surfaces. *Shape analysis and structuring*, pages 53–82.

[Botsch and Kobbelt, 2004] Botsch, M. and Kobbelt, L. (2004). A remeshing approach to multiresolution modeling. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 185–192.

[Lévy et al., 2023] Lévy, B., Petitjean, S., Ray, N., and Maillot, J. (2023). Least squares conformal maps for automatic texture atlas generation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 193–202.

[Qi et al., 2012] Qi, M., Cao, T.-T., and Tan, T.-S. (2012). Computing 2d constrained delaunay triangulation using the gpu. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 39–46.

[Sheffer et al., 2005] Sheffer, A., Lévy, B., Mogilnitsky, M., and Bogomyakov, A. (2005). Abf++: fast and robust angle based flattening. *ACM Transactions on Graphics (TOG)*, 24(2):311–330.

[Zheng and Tan, 2020] Zheng, J. and Tan, T.-S. (2020). Computing centroidal voronoi tessellation using the gpu. In *Symposium on Interactive 3D Graphics and Games*, pages 1–9.