Fall 2015

Report for Advance Algorithm Programming Assignment 1

Yeojin Kim

1 Implementation Details

1.1 Algorithm

The crust algorithm is as follows:

1. Compute Voronoi diagram of the given points.
   a. Create points in 4D using 3 coordinates : $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{x^2} + \mathbf{y^2} + \mathbf{z^2})$
   b. Compute convex hull in 4D by calling qhull[1] with 4D points.
   c. Loop through all facets in 4D convex hull,
      i. If this facet is in lower hull and the volume of facet is not zero,
      ii. Compute the center of facet. This is a Voronoi vertex of the vertices (i.e. Voronoi cites) in facet.
      iii. Save the Voronoi vertex and bounded/unbounded information in each vertex in facet.
2. Find a pole and an antipole. Loop through all Voronoi cites,
   a. If Voronoi cell is unbounded, find an average normal n^+ of adjacent triangles.
   b. If Voronoi cell is bounded, find a pole comparing the distance between the cite c and Voronoi vertices. The farthest Voronoi vertex becomes pole p. Compute a normal vector n^+ from vector $\overrightarrow{cp}$.
   c. Compute an antipole using normal n^+. Voronoi vertex which has the smallest value from inner product with n^+ is an antipole.
3. Apply Delaunay triangulation to the union set of sample points, poles, and antipoles. Print faces only when the face consists of sample points.

1.2 Software

OS : Window 8.1K
IDE : Visual Studio 2013

2 Example Output

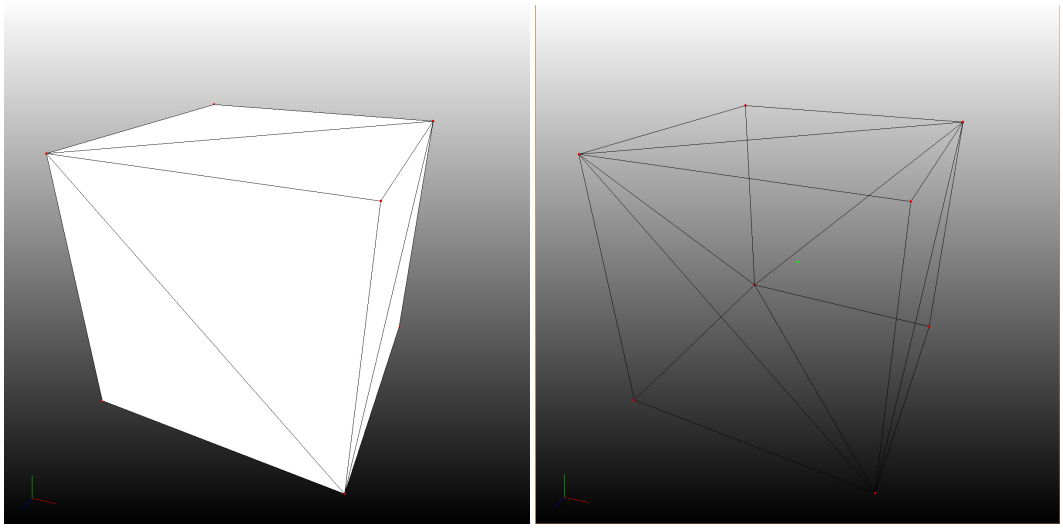
At the first time, I removed some points which have the distance between the cite and vertex is bigger than 10000, but in some cases I cannot obtain fair crust of models. Therefore, I set the distance threshold as 1000 for all models. In Fig. 1 - Fig. 12, (a) shows the result of crust. (b) - (c) shows the inside/outside of sample points. Red points are given sample points and green points are poles and antipoles. Through (b) - (c), we can observe that poles and antipoles lie close to the medial axis.

3 Know bugs/limitations

First, I set some thresholds for fair results. In section 1.1, I skipped Voronoi centers in tetrahedron which has very small volume, which is close to zero. Even if I used the DELAUNAY options, the computing Voronoi vertices generates points which are almost coplanar with four vertices and it makes unexpected holes. So I give the condition about volume. Also I mentioned in Section 2, I removed some points far away from the cites. I set this value manually regardless of models.
In addition, if program loops through vertices not facets, it can compute crust more fast. It can compute a Voronoi vertex, a pole and an antipole at the same time, while visiting a cite once.

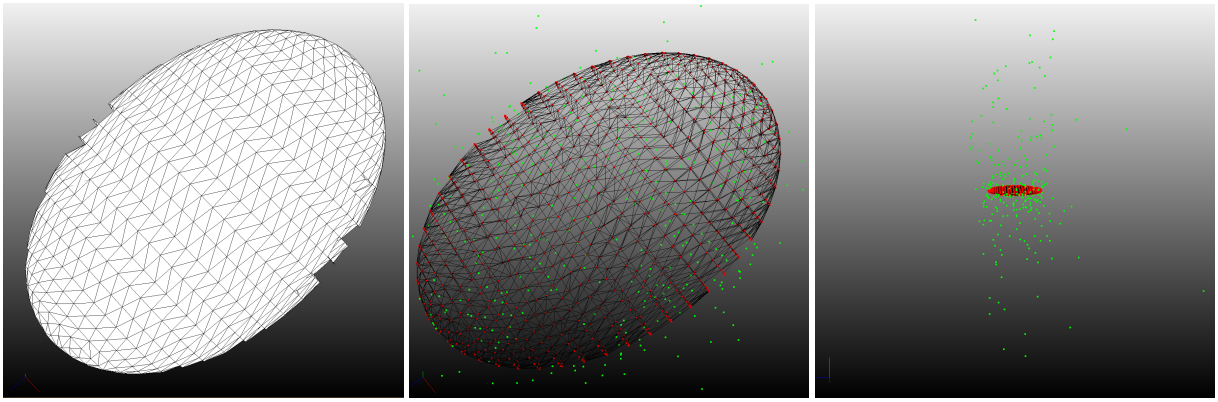
References

[1] Qhull. http://www.qhull.org/. Accessed: 2015-10-14. b.



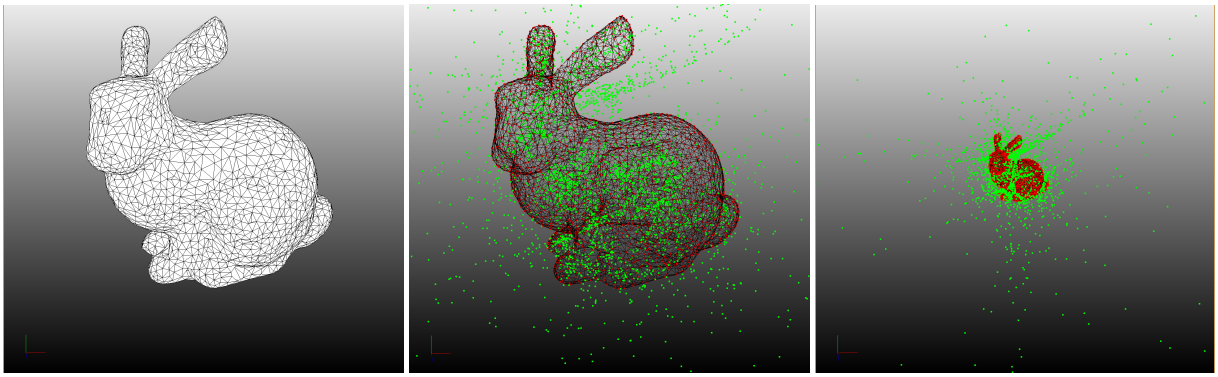
(a) (b)

Figure 1: A cube example



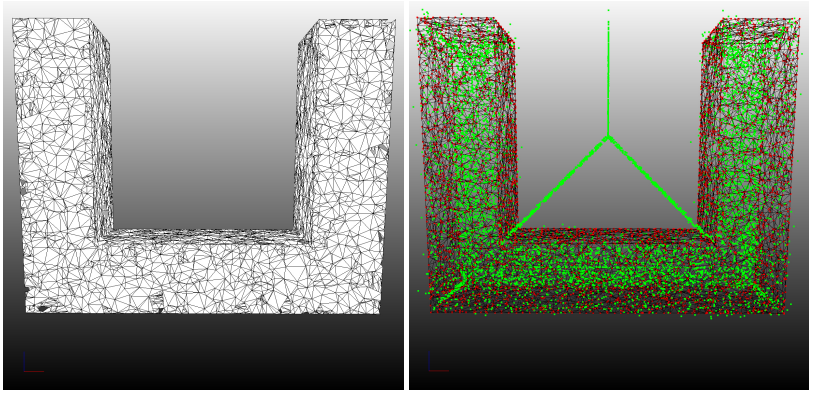
(a) (b) (c)

Figure 2: An ellipsoid example



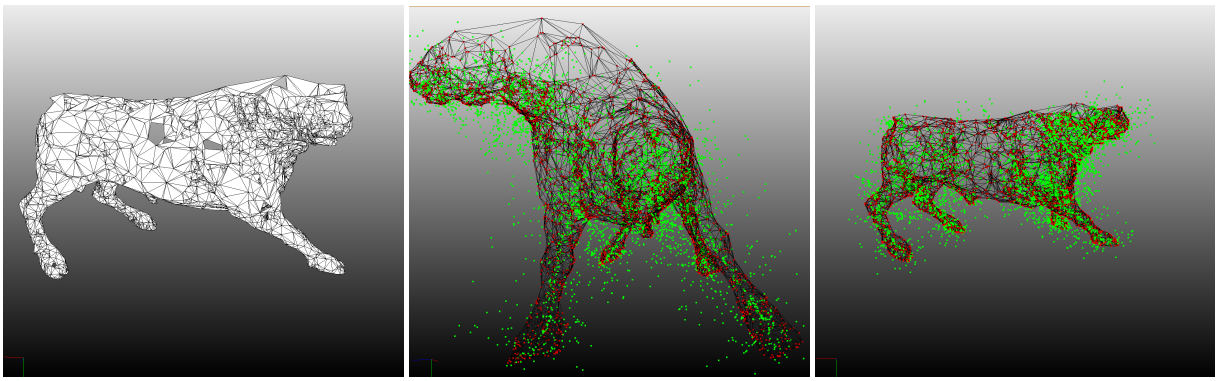
(a) (b) (c)

Figure 3: A bunny example



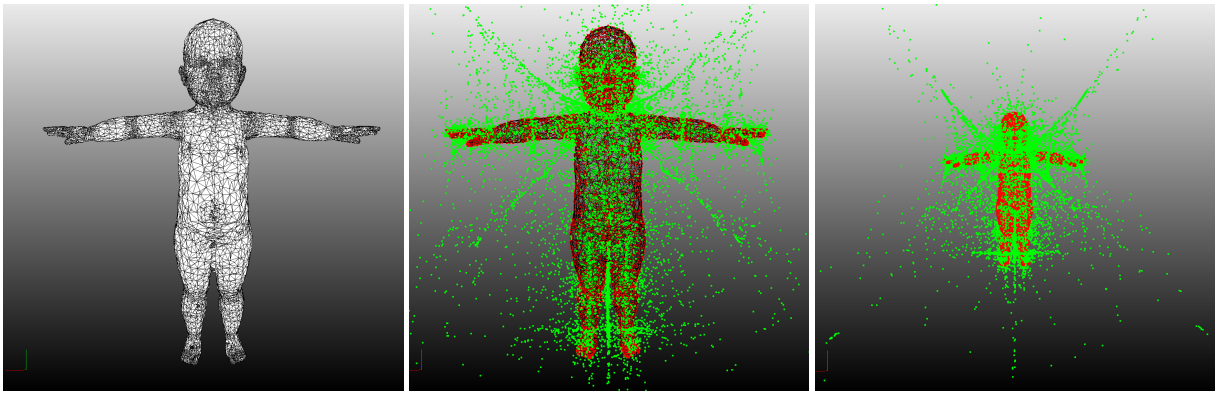
(a) (b)

Figure 4: An 'U' example



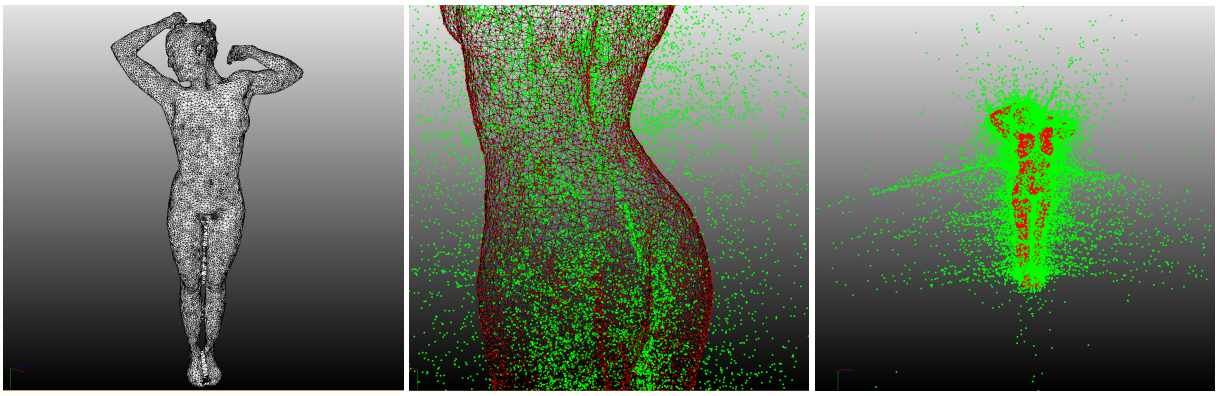
(a) (b) (c)

Figure 5: A bull example



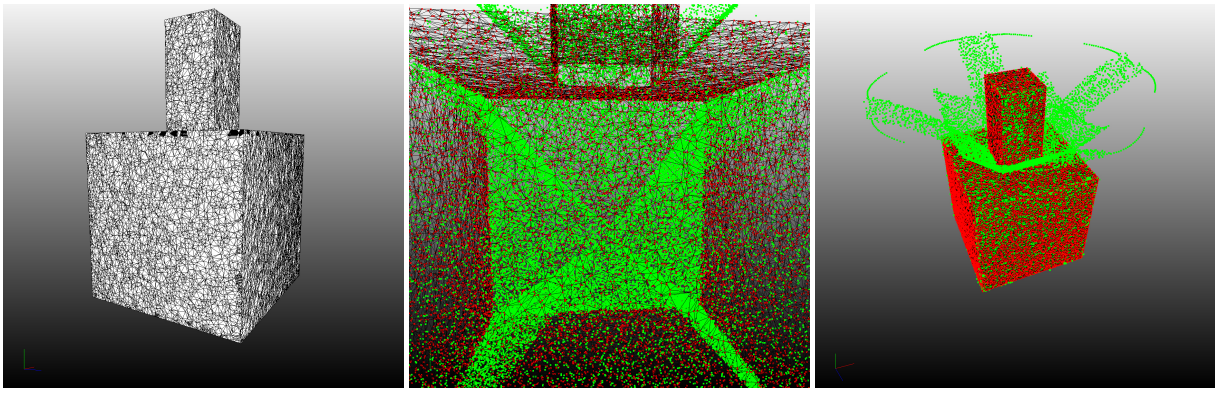
(a) (b) (c)

Figure 6: An baby example



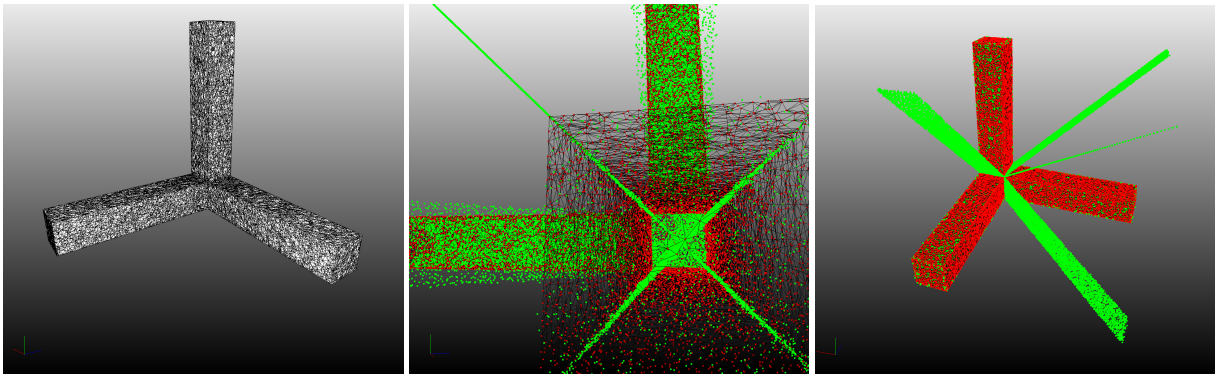
(a) (b) (c)

Figure 7: An woman example



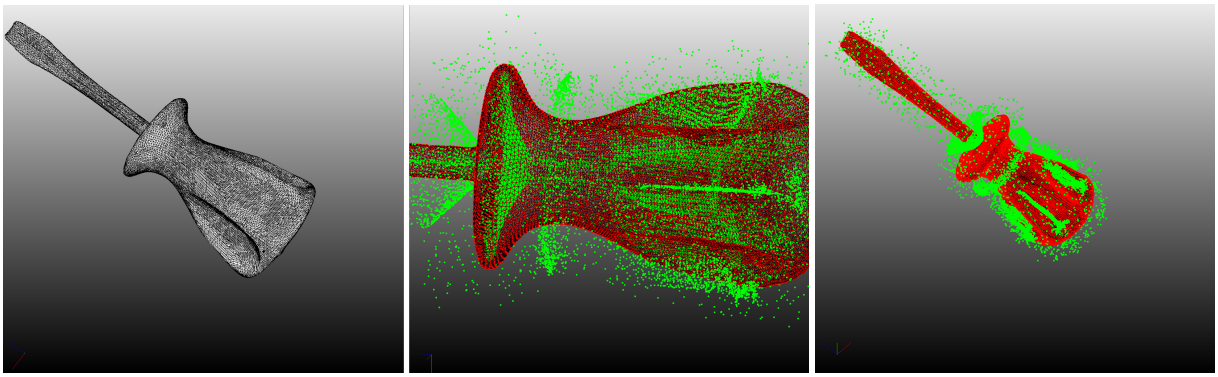
(a) (b) (c)

Figure 8: A 'T' example



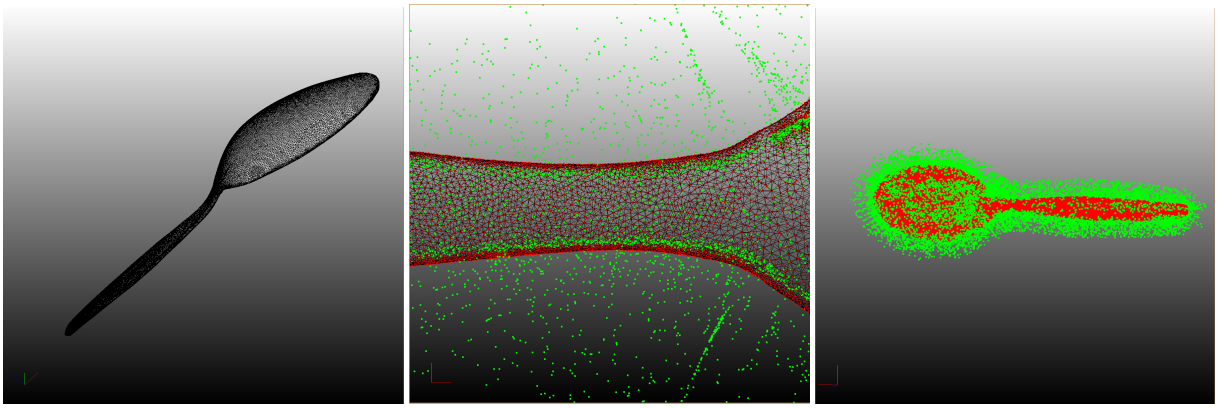
(a) (b) (c)

Figure 9: An 'Y' example



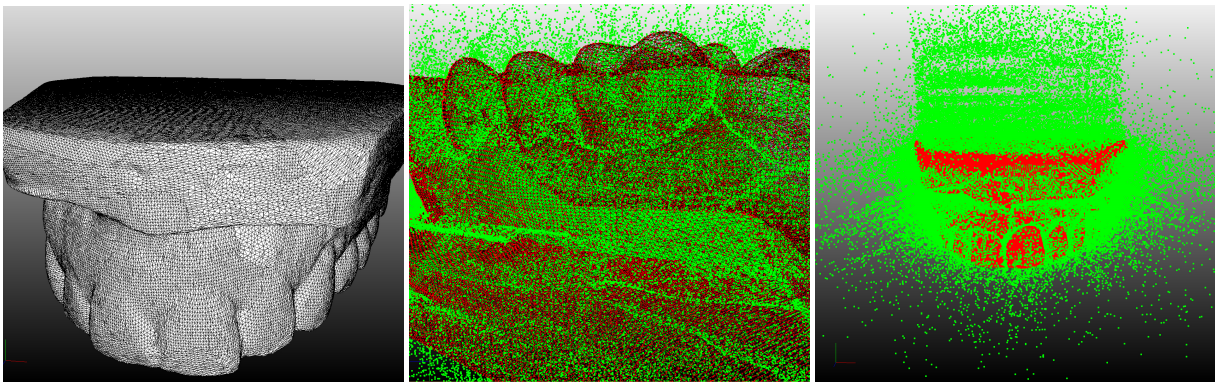
(a) (b) (c)

Figure 10: A screwdriver example



(a) (b) (c)

Figure 11: A spoon example



(a) (b) (c)

Figure 12: A teeth example