Advanced Algorithms	Fall 2015

Program Assignment 1

Prof. Jyh-Ming Lien

152cpg04 Yunjoo Park

yunjoopark12@gmail.com

git clone https://github.com/yunjoopark/programmingAssignment1.git

1 3D Delaunay Triangulation

1.1 Problem definition and Analysis

Let $P := \{p_1, p_2, ..., p_n\}$ be a set of points in the plane. In that, a *triangulation* of P is defined a maximal planar subdivision whose vertex set is P and it denoted by $DT(P)$. if there are three points $p_i, p_j, p_k \in P$ are vertices in the same face of the $DT(P)$, the circle through p_i, p_j, p_k contains no other points. This circle is called the circumcircle of the triangle defined by (p_i, p_j, p_k). We can relatively easily compute the *Delaunay Triangulate* by using convex hull("qhull"). In this report, 3D points are given. Lift the points to 4D so that each point is $(x, y, z, x^2 + y^2 + z^2)$ and compute the convex hull in 4D. As a result, we could get the tetrahedron as the ouput.

1.2 Platforms

Languages: C

Platform: Microsoft visual Studio 2013

1.3 Examples of Inputs

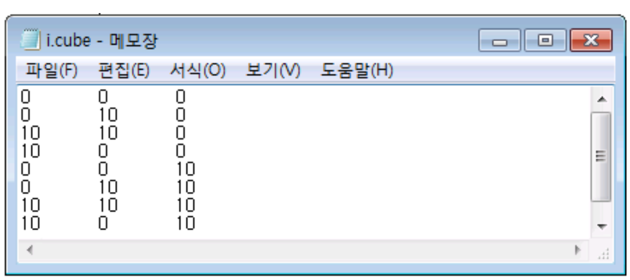

Figure 1: An Example of inputs

1. There are 11 test data.

2. Each data file has different values.

3. Each line has three value, x, y, z.

1.4 Problem-solving methods and algorithms

For lifting the given 3D points to 4D, add another $pt4 = x^2 + y^2 + z^2$. Then, we have the points in 4D. Next, compute convex hull in 4D by using qhull. Since this is in 4D, each face is a tetrahedron. Therefore, we have to call a *function MakeNullTetra()*. For each tetra, we have check its volume with a *function Volumei(face, vertex)*. Make a triangle with any three points in tetra and let the other point be a vertex. Except a tetrahedra having volume 0, make the facet. Then, we could get the Delaunay Triangulation with the set P.

1.5 Results Analysis and Discussion

In this report, render a delaunay triangulation graph from given 3D points. I could study the concept the Delaunay Triangulation. Firstly, speaking about the Delaunay Triangulation, it could be computed by convex hull. For reporting triangles as a result, I have to confirm whether a trinagle is facing down or not. The normal of a tetra has 4 values becasue the given points is 3D. Among them, if the tetra is lower facet, the last coefficient of its normal is clearly negative. Thus, I can confirm which is lower facet by comparing *normal* with 0.0. Also, I have to check the volume of tetra because there could be a tetra having no the volume. Make faces only meeting two conditions mentioned. Since using *qhull*, the algorithms has the time complexity $O(nlogn)$. Actually, I tried to complete this project with a macOS. I however, spent much of my time to fix a problem about the qhull libarary. I downloaded it and copied it to the qhull directory. But, the GUI Keys did not work. So, I completed the project in Windows. Also, I think it was a very good opportunity to learn the LaTeXand *git*. I think they are important in my study.

1.6 Outputs

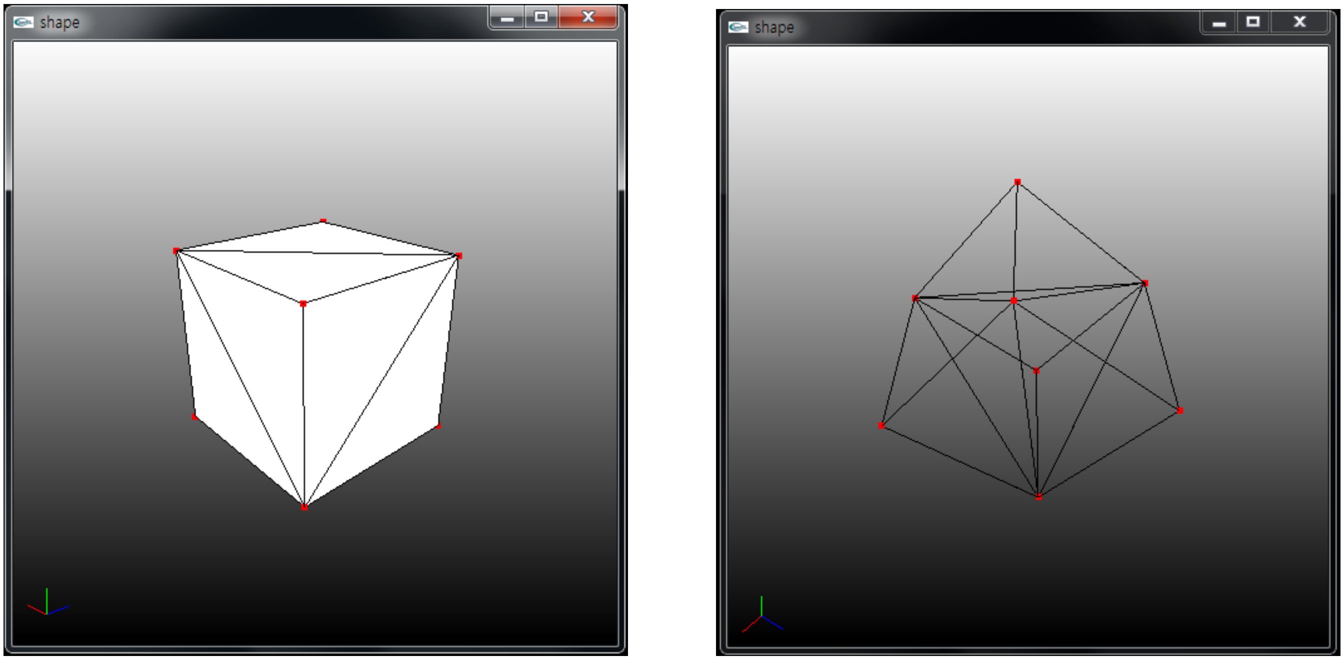

Figure 2: facets and edges *i.cube*

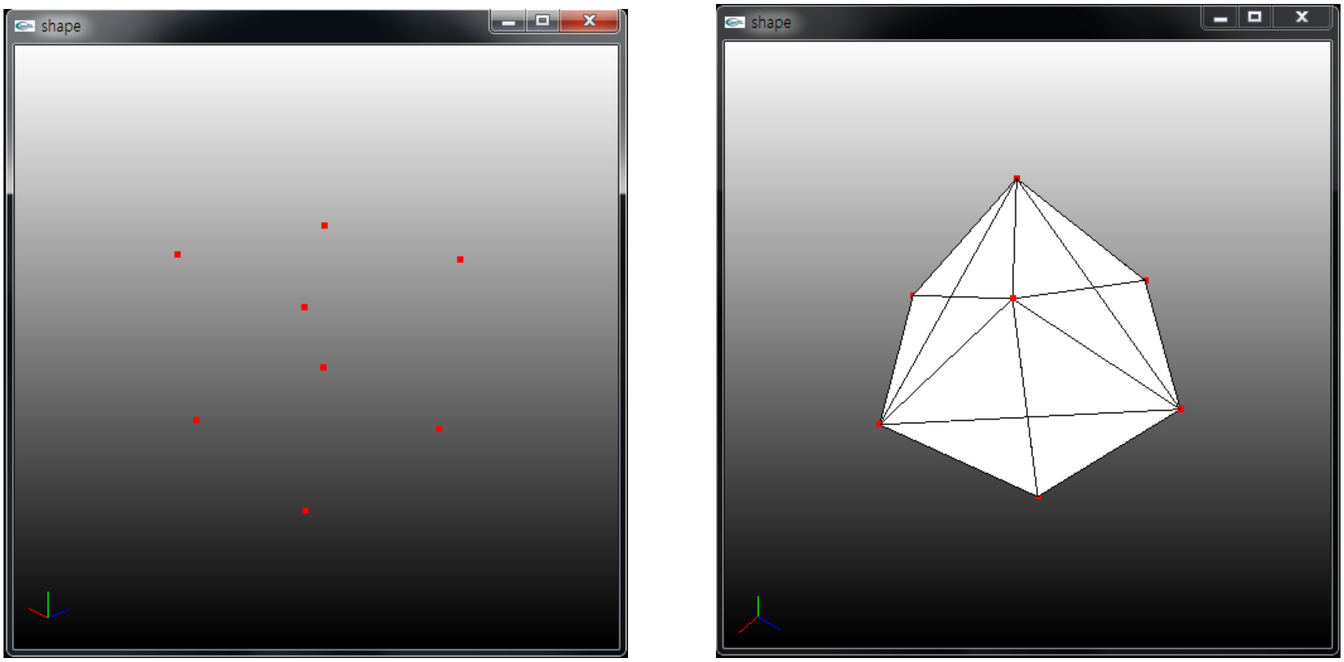

Figure 3: vertices and tetrahedra *i.cube*

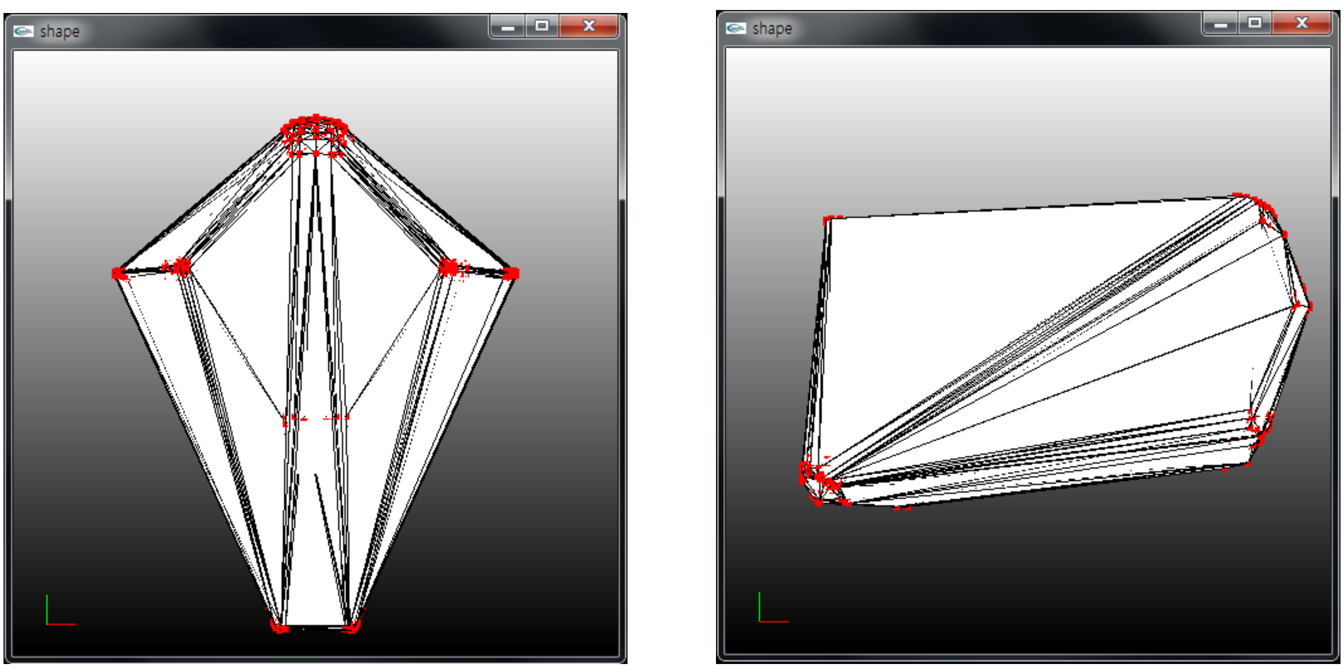

Figure 4: tetrahedra *i.bb and i.bull*

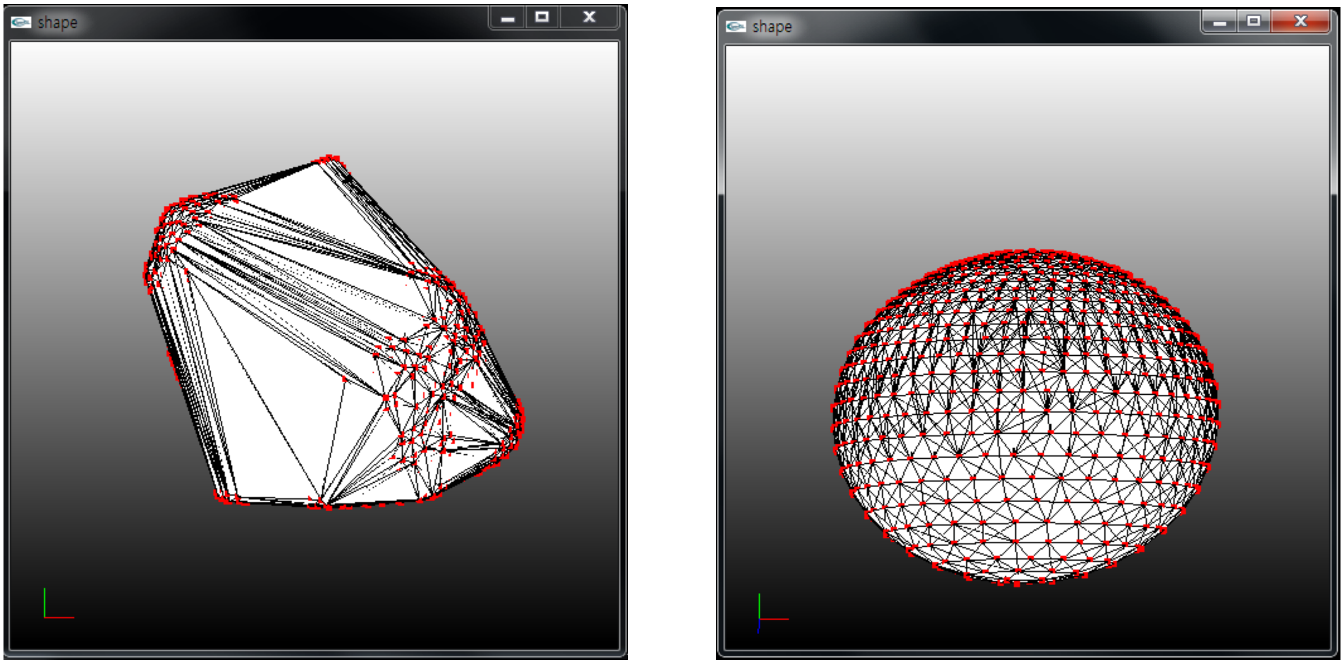

Figure 5: tetrahedra *i.bunny and i.ellippsoid*

Figure 6: tetrahedra *i.screwdriver and i.teeth*

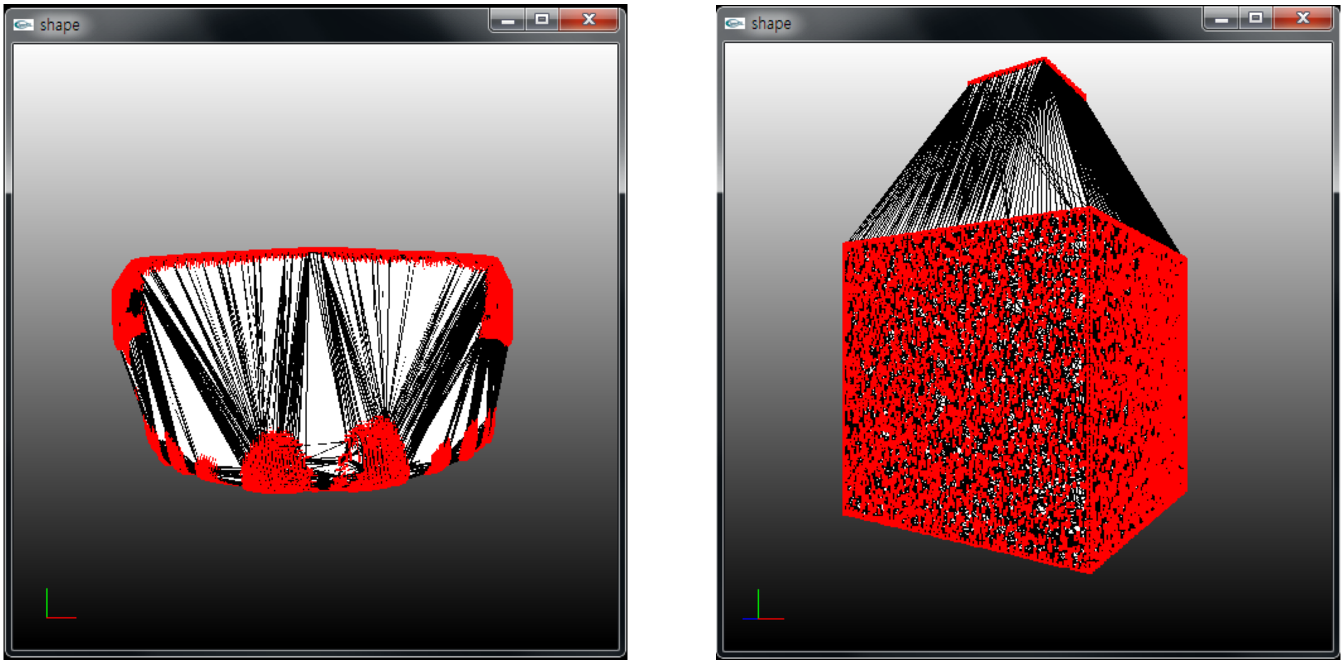

Figure 7: tetrahedra *i.T and i.ellippsoid*

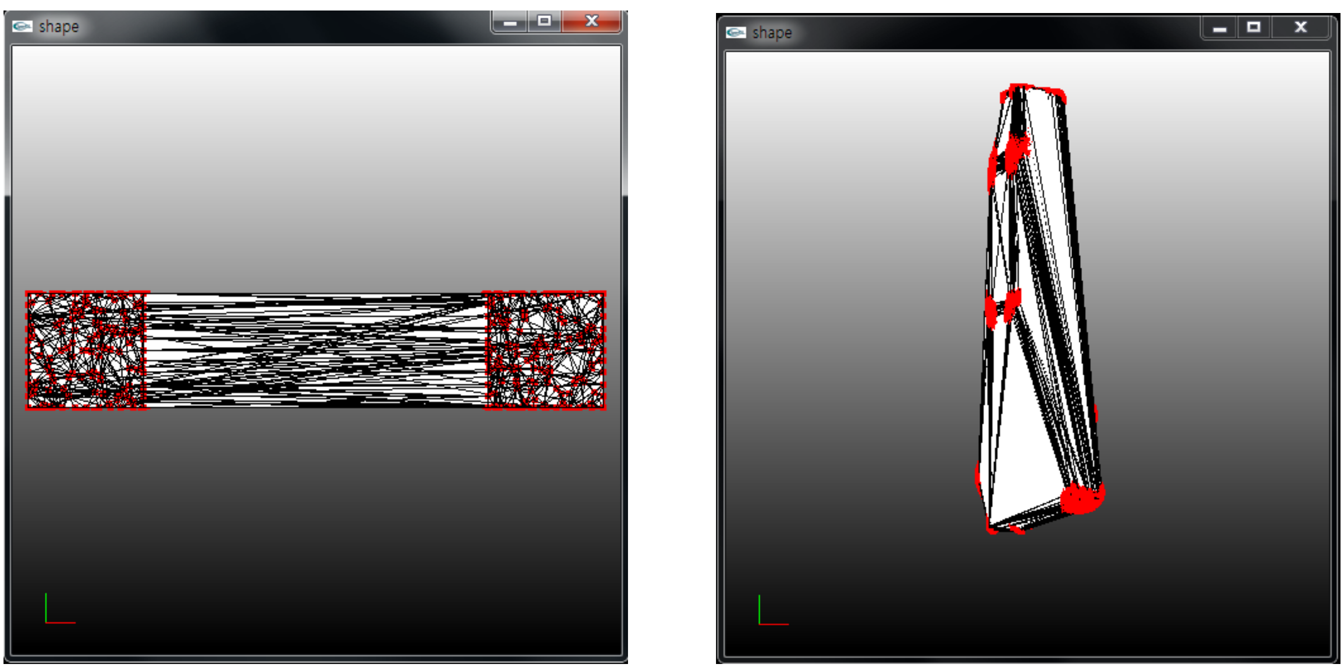

Figure 8: tetrahedra *i.U and i.woman*