02585 COMPUTATIONAL GEOMETRY PROCESSING

Delaunay Triangulation

Michael Mc Donnell **s052319** _____
Olivier Rouiller **s090842** _____

The Technical University of Denmark
Department of Infomatics and Mathematical Modeling

1 Introduction

The Delaunay triangulation is one the most famous algorithms of computational geometry processing and was discovered in 1934 by Biris Delaunay. Usefull to sore geometrical data efficiently, to falicitate neighbourhood queries in geographical databases or to make a triangle mesh look nicer, it is also a basis for many algorithms of computational geometry processing such as mesh refinement or meshing.

A Delaunay triangulation can be defined as a triangulation for wich each triangle has an empty circumbcircle. This triangulation has many desirable properties such as maximizing the angles between edges.

In this exercise, we implement an incremental algorithm to build a Delaunay triangulation. We will detail this algorithm in the remaining.

2 Incremental algorithm

Here we present an overview of the incremental algorithm. It is presented in listing 1. After an initialization, we insert the points of the point set one by one. We first find the face that contains the point to insert and subdivide this face with the point. Insertnig a point in the triangulation is likely to break the empty circle property in the neighbourhood of the new vertex. We used edge flips to restaure the Delaunay property.

Algorithm 1 Delaunay triangulation incremental algorithm

```
initializeTriangulation()
for all point in pointSet do
  face = locate(point)
  subdivide(face,point)
  restoreDelaunayProperty()
end for
```

3 Initialisation

For a precision purpose, we transform the point set so that all the points are located in the square $[-1, 1] \times [-1, 1]$ In order to avoid to deal with the case of inserting points outside the convex hull of the previous points. To do so we translate the point set by the opposite of the vector from the origine to the barycenter of the point set. Then we scale the points by the maximum of the absolute value of the coordinates of the points.

we initialize the triangulation with a triangle containing the convex hull of the point set. As the point set in included in the unit square, the points $p1 = (0, 2.5)$, $p2 = (3.5, -1.5)$ and $p3 = (-3.5, -1.5)$ are suitable as shown.

4 Location in the triangulation

To find the face where to insert a new point, we iterate over the faces and check if the point is inside the face until we find the right face.

To check whether a point is inside or outside a face, we perform a test based on orientations. A Point is inside a triangle if and only if for each edge, the point is in the same side of the edge than the opposite vertex of the edge.

The pointInFace predicate for a triangle $p_1p_2p_3$ and a point p can then be expressed as $sameSide(p_1, p_2, p, p_3)\&sameSide(p_2, p_3, p, p_1)\&sameSide(p_3, p_1, p, p_2)$ where the predicate $sameSide(p_1, p_2, p, q)$ is true if p and q are on the same side of the line passing through p_1 and p_2.

the predicate $sameSide$ compares the signs of the cross products $(p_2 - p_1) \times (p - p_1)$ and $(p_2 - p_1) \times (q - p_1)$.

5 Restoring the Delaunay property

Their are two common approach to restaure the Delaunay property after inserting a new vertex. One can visit the neighbooring faces of the face that has been subdivided, mark the one whose circumcicle contains the new vertex, delete them and link the vertices of the boundary of the hole with the new vertex.

We used the other aproach here, we perform edge flips on the illegal edges that are created after inserting the point. This method is alsow known as the Lawson algorithm. An illegal edge is an edge of a triangle whose circumcircle is not empty. It is known that when inserting a new vertex in the triangulation, illegal edges, if any, are in a neighbourhood of the new vertex. After inserting the points it is only necessary to check the edges of the face that as bee subdivided. If a flip occures on one edge, we extend the boundary to check with the adjescent edges of the fliped edge.

Figure 1 to 2 shows an insertion in a Delaunay algorithm using our method.

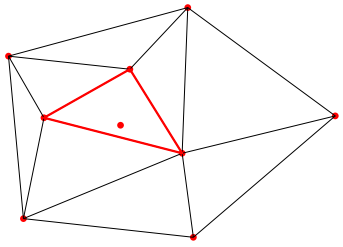

Figure 1: The point is located and the region to check is marked in red.

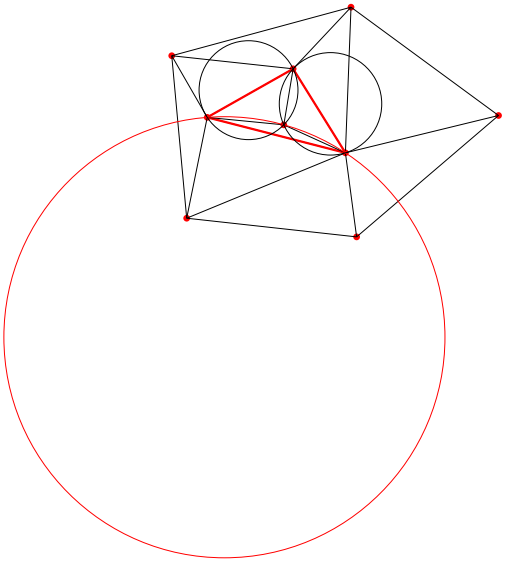

Figure 2: The face is subdivided and we check the first three edges, here one edge needs to be fliped.

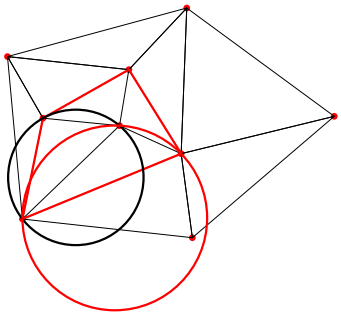

Figure 3: The illegal edge is fliped and the region extended, another flip is needed.

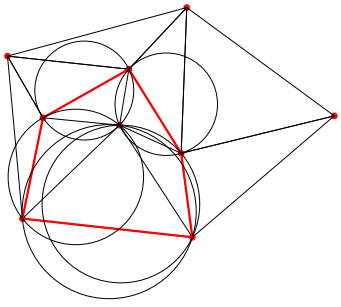

Figure 4: The illegal edge is fliped and nothing else needs to be done.

6 results

We implemented the presented algorithms and we could generate the Delaunay triangulation of the given point sets. Figure 5 shows the final tri-

angulation, figure 6 shows the same triangulation with empty circumcircles displayed and figure 7 show the Delaunay triangulation of the second data set.

Figure 5: The Delaunay triangulation of the first point set

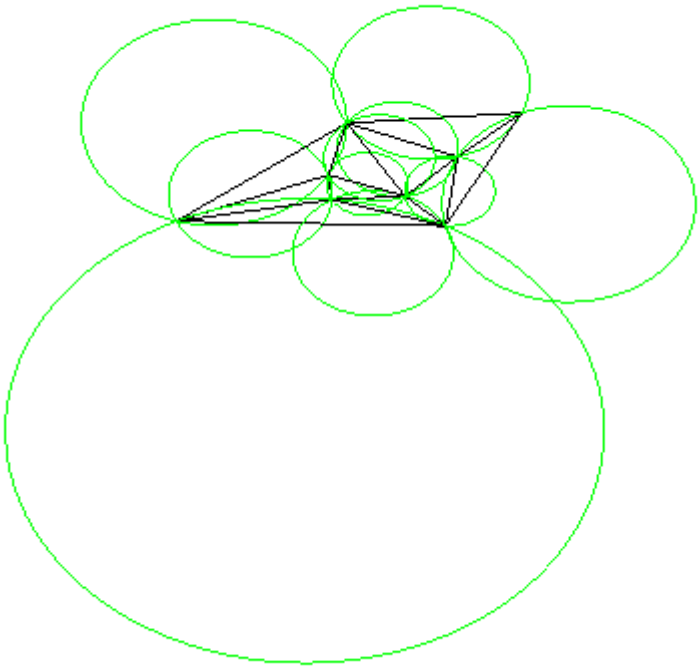

Figure 6: The Delaunay triangulation of the first point set with displayed circumcircles

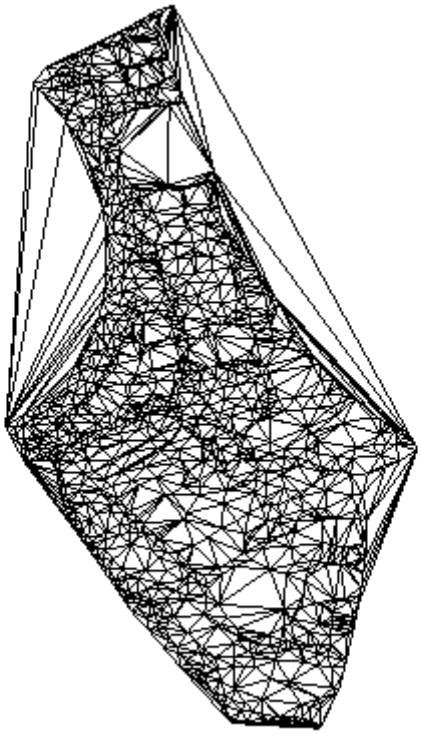

Figure 7: The Delaunay triangulation of the second data set