CS 558 Assignment 3 Part 2 Analysis

Jieru Hu

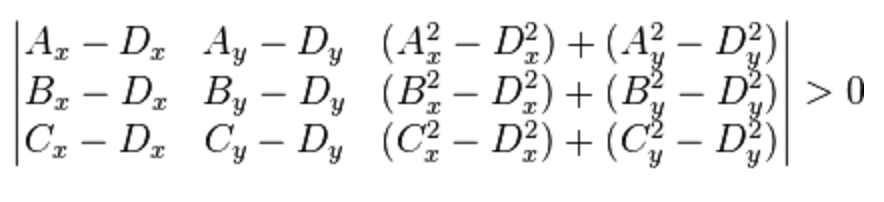
ID: 970464783

Overview:

The part 2 of the assignment develops the edge-flipping algorithm for creating Delaunay triangulation based on the arbitrary triangulation created in part 1.

Algorithm:

First, I created a method to test if a given edge is a Delaunay edge. Given an edge, suppose vertex A, B, C forms a triangle and B, C, D forms a triangle. We would like to know if point D is within the circle formed by the A, B, C. An easy way to evaluate it is to create a matrix using four vertexes, calculate its determinant and make the judgement.



Referring to the picture, A, B, C should be listed in the counterclockwise way. If the determinant is positive, then the D lies in the circle. So the edge is not a Delaunay edge.

The basic idea of the edge-flipping algorithm is to flip all the edges until there is not non-Delaunay edge in the triangulation. We use a stack to keep track of all non-locally Delaunay edges. The stack can only have one copy of the edge.

To start with, we are given an arbitrary triangulation of points. We use the method mentioned above to push all on-locally Delaunay edges into the stack and mark them. While the stack is not empty, we pop an edge out from the stack and set it as curr. We unmark the curr. If the curr is a non-locally Delaunay edge, we simply flip the edge, which includes the process of deleting two old triangles and add two new triangles. Next, we determine if the four surrounding edges of the curr is marked. If they are unmarked, we mark them and push them into the stack. If they are marked, we just skip them. At the time, when the stack is empty, we are finished with the Delaunay triangulation.

Running time:

For the worst case, supposed all the edges are non-locally Delaunay edges, which is usually unlikely to happen. Each edge may result in pushing at most another n-1 edges into the stack. So the running time of the algorithm is kept below O(n2).

Note: When I was doing part 2, I found some little bugs in part 1. So I resubmitted it. Sorry for the trouble.