Delaunay Triangulation Algorithms

Maharshi Rawal (UFID: 9990-8457)

CISE Department, University of Florida, Gainesville, FL

ABSTRACT

The Delaunay Triangulation for a given set of points set **P** of discrete points in a plane is a triangulation DT(**P**) such that no point in P is inside the circumcircle of any triangle in DT(**P**). Delaunay triangulation is used to maximize the minimum angle in a triangle. This paper reviews some well-known algorithms for finding the Delaunay triangulation: Flip Algorithm, Incremental Algorithm, Divide and Conquer Algorithm, and Sweep-Hull Algorithm. We talk about the algorithms and their implications and compare the time complexity and memory requirements.

I. INTRODUCTION

The Delaunay triangulation in computational geometry is a dual graph of Voronoi diagrams which was discovered by a Soviet mathematician Boris N. Delaunay (Delone)[1]. Delaunay triangulation finds its application in various domains such as modelling of terrain, calculating density of points for packing and covering, etc. Apart from these, Delaunay triangulations find application in other algorithms such as proof of MST chain, nearest neighbors problem, largest empty circle problem.

To understand the Delaunay triangulation[6], we first need to understand the empty circle property. For a given circle:

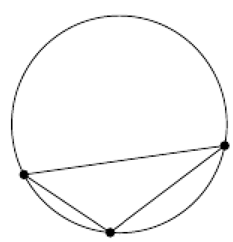
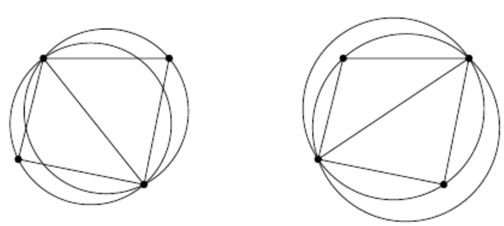


Figure 1. Circumcircle of a triangle.

We first find the circumcircle of the triangle which is a unique circle passing through the 3 points of the triangle. A triangulation of a finite point set P is called a Delaunay triangulation if the circumcircle of every triangle is empty, that is, there is no point from P in its interior. Considering the next two figures:



(a) Delaunay triangulation. **(b)** Non-Delaunay triangulation.

Figure 2. Delaunay triangulation explained

There are a number of algorithms proposed to find the Delaunay triangulation. In this survey paper we take a look at a few of those algorithms.

II. BRIEF OF ALGORITHMS

Lawson's Flip Algorithm

Lawson's Flip Algorithm [2] to compute the Delaunay triangulation is given for a set of points P by first computing some triangulation of the points in P which are sorted in ascending order of their coordinates which are then joined to form triangles within the convex hull. Until there exists a non-Delaunay triangulation, replace the sub-triangulation by other triangulation of the points which is done by edge flipping.

Incremental Algorithm

The simplest of the approaches to find the Delaunay triangulation is to take one vertex at a time and add it to the Delaunay triangulation. After this, flip edges to obtain the correct Delaunay triangulation. If a Naive approach is followed, the algorithm would take $O(n^2)$ time. There are a lot of approaches proposed to incrementally form the Delaunay triangulation. In this paper, we talk about the approach put forward by Bowyer and Watson independently.

Divide and Conquer Algorithm

The divide and conquer algorithm to compute the Delaunay triangulation was proposed by Lee and Schachter [3] which was improved by Guibas and Stolfi [4] and later by Dawyer. Here, the set of points P is subdivided into smaller sections by recursively drawing lines to separate the existing vertices into two sets. The Delaunay triangulation is obtained for each set and later merged to form the Delaunay triangulation of the complete set of points P. Divide and conquer algorithms are the fastest of the Delaunay triangulation algorithms.

Sweep Hull Algorithm

Sweep Hull Algorithm to compute the Delaunay triangulation of set of 2D points. The basic component of this algorithm is a radially propagating sweeping hull which sweeps the set of points P which are radially sorted and connects the points in the set P giving a non-overlapping triangulation which is then merged with a final triangle flipping procedure which gives the Delaunay triangulation.

III. ALGORITHMS

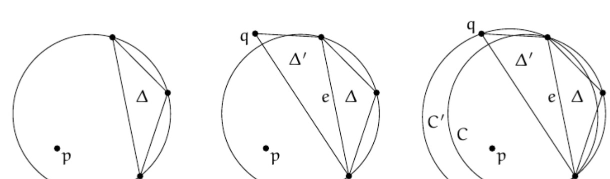
LAWSON'S FLIP ALGORITHM:

For a set P of n points:

1) Compute a triangulation of P which can be computed by using scan triangulation, i.e. forming triangles by joining points in the plane.

2) While there still exist a sub-triangulation of four points in the convex position which is not Delaunay (there is at least one point in the circumcenter of any sub-triangulation in the main triangulation), replace this sub-triangulation by other triangulation of the four points.

Theorem: Let $P \subseteq \mathbb{R}^2$ be a set of n points, equipped with some triangulation τ. The flip algorithm terminates after at most $\binom{n}{2} = O(n)^2$ flips and the resulting triangulation D is a Delaunay triangulation of P.



(a) A point p inside the circumcircle C of a triangle Δ. (b) The edge e of Δ closest to p and the second triangle Δ′ incident to e. (c) The circumcircle C′ of Δ′ also contains p, and p is closer to Δ′ than to Δ.

Figure 3. Lawson's Flip Algorithm's Correctness

The correctness of Lawson's Algorithm is observed by the fact that for two triangles which are next to each other in the triangulation, the circum-circle of any triangle does not have the vertex of the adjacent triangle in its interior, thus proving the triangulations are locally Delaunay. We can subsequently deduce by Proof of Contradiction or by using power of Circles that locally Delaunay triangulations are actually globally Delaunay and hence Lawson's Flip Algorithm correctly computes Delaunay triangulation.

Time Complexity is $\Omega(n^2)$

Space Complexity is O(n)

INCREMENTAL ALGORITHM:

The Bowyer-Watson Incremental Algorithm is a Delaunay triangulation algorithm which computes the triangulation in any dimensions. The algorithm can also be used to obtain the dual of a Delaunay i.e the Voronoi diagram.

1. To obtain the vertex structure of the tessellation, 8 points are used which give rise to seven vertices $v_1 - v_7$. Each vertex is a circumcenter of 3 data points.
2. Find a vertex which is closer to a new point. There will always be a vertex in the Delaunay structure which will be closer to a newer point than to its previous point. Delete this vertex.
3. From the deleted vertex, perform a tree search to find the list of other vertices that would be deleted by the addition of the new point in the triangulation.
4. The points contiguous to the newer point forms a list of deleted vertices.
5. The vertices which are not yet deleted and are neighboring to the newer set of points form the list of newer vertices that get added to the Delaunay triangulation.
6. The final step is to copy the newer vertices to the originally deleted vertices, thereby saving space. There would always be neighboring vertices to the originally deleted vertex set, so that newer vertices may be added to the Delaunay triangulation.

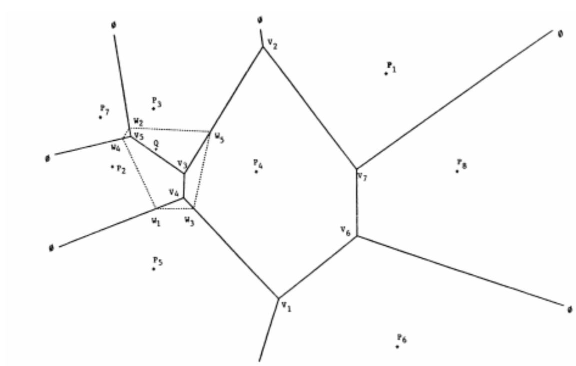


Figure 4. Bowyer-Watson Algorithm

Time-Complexity: Worst Case: $O(n^2)$, Average Case: O(nlogn)

Space-Complexity: O(n)

DIVIDE AND CONQUER ALGORITHM:

The divide and conquer algorithm to compute the Delaunay triangulation use a quad-edge data structure which gives the dual of the Delaunay triangulation i.e. a Voronoi diagram for free. The algorithm is as follows:

1. Start by partitioning the x-coordinate sorted set of points in two halves: Left Half (L) and Right Half (R).
2. Split the points in the middle, and recursively divide the left half (L) and right half (R) until there are only 3 points left (which cannot be further sub-divided and these need to be handled separately).
3. In the case of ties, the points are resolved on the increasing y-coordinate. Duplicate points are discarded. The split in the middle ensures that there are only L-R edges which are also called as cross edges and there are no L-L or R-R edges. The cross edges are produced in incremental order i.e. in ascending y- coordinates.

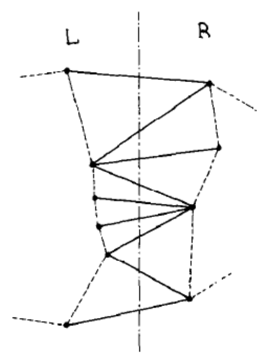


Figure 5. The L-R edges structure

The correctness of this algorithm is computed by the lemmas presented in the paper.[5]

Time-Complexity: O(nlogn)

Space-Complexity: O(n)

SWEEP-HULL ALGORITHM:

Sweep-Hull algorithm[8] is a new algorithm for finding Delaunay triangulation which is inspired by the Sweeping-Line algorithm proposed by Fortune in 1987. The time complexity for Sweep-Hull (S-Hull) algorithm is similar to Divide and Conquer Algorithms which is O(nlogn). S-Hull algorithm works in the following fashion:

1. A seed point (x_0) is selected from the set of points (x_i) .
2. Sort the set of points according to $(x_i - x_0)^2$
3. Now, find a point which is closest to the seed point x_0 .
4. Now, given the set of points, find a 3^{rd} point x_k which forms the smallest circum-circle with x_0 and x_j. Record the center of the circum-circle as C.
5. To get the initial seed convex hull, order the points [x_0, x_j, x_k] to give a right-hand system.
6. Again sort the points depending on $|x_i - C|^2$ to give a new set of points s_i.
7. From the position of initial triangle of points [x_0, x_j, x_k], start propagating the 2D convex hull and sequentially add points s_i to the hull.
8. A non-overlapping triangulation is obtained from the set of points.
9. Flip the adjacent pairs of the triangle to create the Delaunay triangulation.

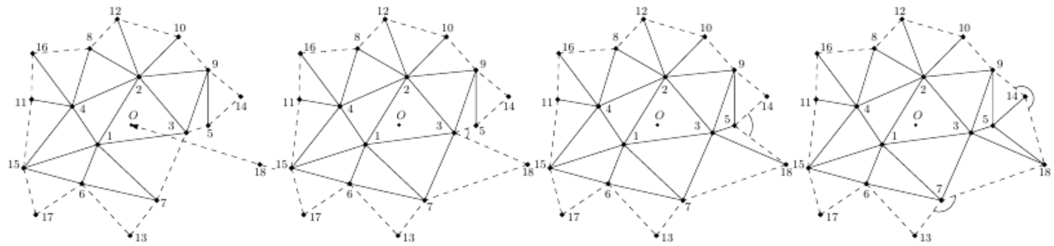


Figure 6. Addition of triangles to the Delaunay triangulation

The correctness of this algorithm is guaranteed by the fact that since the points are radially sorted according to a certain parameter, the Sweeping hull always gives out the correct non-overlapping triangulation which are flipped in the next step of the algorithm to give the Delaunay triangulation.

Time Complexity: O(nlogn)

Space Complexity: O(n)

IV. DISCUSSSION

	Flip Algorithm	Incremental	Divide and Conquer	Sweeping-Hull
Time-Complexity	$\Omega(n^2)$	$O(n^2)$	$O(nlogn)$	$O(nlogn)$
Space-Complexity	$O(n)$	$O(n)$	$O(n)$	$O(n)$

Table 1. Analysis of Algorithms used to compute Delaunay Triangulation.

In this survey paper, we discussed about Delaunay triangulation and the algorithms used to compute the Delaunay triangulation. We started of with the classical algorithms like flip algorithm, incremental algorithm and divide and conquer algorithm. Then we talked about a newer algorithm which is the Sweep-Hull algorithm which provides equivalent performance to Divide and Conquer algorithm. We also talked about the applications of Delaunay triangulation which is used in terrain modelling, finding nearest neighbors in a graph, etc. Triangulation techniques are also used in other domains such as location services (GPS) where an entity's location is determined by performing triangulation techniques. A lot of geometric graphs are subgraphs of the Delaunay triangulation such as Euclidean Minimum Spanning Tree, Gabriel Graph. Another important triangulation in computational geometry is the minimum weight triangulation i.e. triangulation with minimal weight of sum of lengths of all edges of triangulation which was shown to be a NP-complete problem. Some open problem relating to Delaunay triangulation are to find whether every planar straight-line graph conforming to Delaunay triangulation has size $O(n^2)$. Also, finding algorithms for triangulation with no obtuse angles can be generalized to inputs with curved boundaries.

REFERENCES

1. B. N. Delaunay. Sur la sph'ere vide. Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793–800, 1934

2. C. L. Lawson. Transforming triangulations. Discrete Mathematics, 3(4):365–372, 1972.

3. Lee, D. T., Schachter R, B. J. Two algorithms for constructing the Delaunay triangulation. Int. J. Comput. hf. Sci. 9, 3 (1980), 219-242.

4. A. Bowyer. Computing Dirichlet tessellations. Comp. Journal, 24(2):162–166, 1981

5. L. J. Guibas and J. Stolfi (1985), Primitives for the Manipulation of General Subdivisions and the computation of Voronoi diagrams, ACM Trans. Graph., pp. 4(2):7 4-123

6. Delaunay Triangulations, *https://www.ti.inf.ethz.ch/ew/Lehre/CG13/lecture/Chapter 6.pdf*

7. Dwyer, R.A. Algorithmica (1987) 2: 137. https://doi.org/10.1007/BF01840356

8. D.A.Sinclair, S-hull: a fast radial sweep-hull routine for Delaunay triangulation, arXiv:1604.01428 (2016)