Documentation
Definitions, Theorems, and Algorithms for The Delaunay Trangulation Application

Yu Xuan Hong & Dani Demas

February 22, 2016

Overview

The Delaunay Triangulation Application is an interactive application that allows the user to construct a Delauney triangulation by flipping edges of a certain triangulation on a given point set. The application automatically triangulates a point set using the incremental algorithm, and draws the circumcircles of each triangle. Using the Empty Circle Property, it shows the user the illegal edges of the triangulation by highlighting the circumcircles that contains points from the original point set. The user can choose to flip an illegal edge, and the application will return a new triangulation and corresponding circumcircles and illegal edges. Eventually, the Delaunay triangulation will be attained, where there is no illegal edge.

1 The Incremental Triangulation

Let the given set of points (input by user) be S. Sort the points in S in ascending order according to their x-coordinates. We will triangulate the point set using the *Incremental Triangulation Algorithm* (DCG, 62) in ascending order of their x-coordinates.

1.1 Incremental Convex Hull

For each incremental inclusion of a new point into the triangulation, we recalculate the convex hull of the triangulation.

Definition 1 *(Next Point)* A point that is either one of the three points with the smallest x-coordinates or has the smallest x-coordinate among points that have not been included in the existing triangulation is a *next point.*

Algorithm 2 *(Incremental Convex Hull)*

1. Include the edges of the triangle defined by the three points with the smallest x-coordinates in the convex hull.
2. For each *next point*, calculate the slope between that point and all points on the convex hull of the existing triangulation.
3. Find the maximum and minimum values of those slopes, and save the edges with those slopes.
4. Include those two edges into the new convex hull.

1.2 Connecting Lines

Definition 3 *(Visibility) (DCG, 14)*

Definition 4 *(Visible Point)* A point is a *visible point* corresponding to a *next point* if it is *visible* to the *next point* and lies on the convex hull of the existing triangulation.

Formula 5 *(Slope of Two Points) Given points* $P_1 : (x_1, y_1), P_2 : (x_2, y_2)$*:*
$l : (y - y_2)/(y_1 - y_2) = (x - x_2)/(x_1 - x_2)$
Slope: $k = (y_1 - y_2)/(x_1 - x_2)$

Algorithm 6 *(Connecting Next Point to its Visible Points)* For each *next point*:

1. find the two points P_1, P_2 with which the *next point* forms a line with the maximum and minimum *slopes*.
2. find the set of points T to the right of (with x-coordinates larger than points on) the line defined by P_1, P_2.
3. connect all points that belong to T to the *next point*.

2 The Delaunay Triangulation Test

Theorem 7 *(Empty Circle Principle) (DCG 3.53)*

2.1 Incremental Circumcircles

Definition 8 *(Next Line)* The line(s) that connect the *next point* with the *visible points* of the existing triangulation are denoted *next line(s).*

Formula 9 *(Circumcircle of Three Non-colinear Points) Given non-colinear points* $P_1 : (x_1, y_1), P_2 : (x_2, y_2), P_2 : (x_2, y_2)$*:*
Center $O(x, y)$:
$x = ((y_2 - y_1)(y_3^2 - y_1^2 + x_3^2 - x_1^2) - (y_3 - y_1)(y_2^2 - y_1^2 + x_2^2 - x_1^2))/2((x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1))$
$y = ((x_2 - x_1)(x_3^2 - x_1^2 + y_3^2 - y_1^2) - (x_3 - x_1)(x_2^2 - x_1^2 + y_2^2 - y_1^2))/2((y_3 - y_1)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_1))$
Radius $r = \sqrt{(x - x_1)^2 + (y - y_1)^2}$

Algorithm 10 *(Constructing Circumcircles Incrementally)* For each *next point*:

1. reorder its corresponding *visible points* in ascending order of *slope*.
2. construct a circumcircle defined by each pair of adjacent points (by the aforementioned ordering) and the *next point*.
3. construct a circumcircle defined by the three points with the smallest x-coordinates in the point set.

2.2 Point in Circumcircle Test

Formula 11 *(Interior Point Test) Given a point $P : (x, y)$ and a circle $O(x_0, y_0)$ with radius r:*
P is in the interior of the circle iff $\sqrt{(x - x_0)^2 + (y - y_0)^2} < r$.

Algorithm 12 *(Delaunay Triangulation Test, by Empty Circle Property)*
For each circumcircle O:
If there exists a point of S that is in the interior of the circumcircle, then it is not a Delaunay Triangulation; else, it is a Delaunay Triangulation.
Highlight all circumcircles that contain at least one point of S. Call the set of those circumcircles U.

3 Illegal Edge Detection

Definition 13 *(Interior Points of a Fixed Circle)* A point in S is an *interior point* corresponding to a fixed circumcircle if it is in the interior of that circumcircle.

Definition 14 *(All Existing Lines in Triangulations)* Let L the set of all lines in a complete triangulation of S.

Formula 15 *(Point-Line Distance) Given points $P_1 : (x_1, y_1), P_2 : (x_2, y_2)$ and $P(x_0, y_0)$*
l defined by P_1, P_2 : $(y - y_2)/(y_1 - y_2) = (x - x_2)/(x_1 - x_2)$
Let $A = y_1 - y_2$, $B = -(x_1 - x_2)$, $C = y_2(x_1 - x_2) - x_2(y_1 - y_2)$
Then
$d(P, l)$: $(Ax_0 + By_0 + C)/\sqrt{A^2 + B^2}$

Algorithm 16 *(Detecting Illegal Edges)* For each circumcircle O_u in U (highlighted): Let the three points of S that (define) lie on that circumcircle be denoted P_1, P_2, P_3,
For each *interior point of that fixed circumcircle* P:
Check if P and P_1, P_2, P_3 are connected – in other words, check if the lines $\overline{PP_1}$, $\overline{PP_2}$, $\overline{PP_3}$ belong to L. If exactly two of them do, then save those four points to a list of quadruples Q=[[P, P_1, P_2, P_3]]; else, pass.
Next, calculate the distances d_1, d_2, d_3 between P and $\overline{PP_1}$, $\overline{PP_2}$, $\overline{PP_3}$. The line with the smallest distance to P is *illegal*. **Highlight** the *illegal* lines.

Theorem 17 *(Shortest Distance)* For a given circumcircle O, among its *interior points*, the one with the least distance to the *illegal edge* is connected to the two points that define that edge. (Hence, it forms a *quadrilateral* with the three points that define O, as will be defined in 4.1.

4 Edge Flip

Definition 18 *(Flippable)* An edge is *flippable* if it is not on the convex hull of the existing triangulation.

We allow the user to flip any *flippable* edge in the existing triangulation.

4.1 Flipping an Illegal Edge

Definition 19 *(Illegal Edge) (DCG, Prop 3.51)*

Definition 20 *(Quadrilateral of Connected Point and Corresponding Triangle)* In the previous section, we have constructed Q, a list of quadruples of points. Each of these quadruples define a *quadrilateral of connected point and corresponding triangle*.

Algorithm 21 *(Flipping One Illegal Edge)* For the selected (highlighted) *illegal edge*, determine the *quadrilateral* (quadruple point set) it belongs to. Delete the selected line; add line between remaining two points in *quadrilateral* (quadruple point set).

4.2 Flipping a Legal Edge

Definition 22 *(Legal Edge)* An edge is a *legal edge* if it is *flippable* and not an *illegal edge*. *(For our purposes, this definition is different from that in DCG, since we have excluded edges that are not flippable.)*

Algorithm 23 *(Flipping One Legal Edge)* For the selected (highlighted) *illegal edge*, determine the *quadrilateral* (quadruple point set) it belongs to. Delete the selected line; add line between remaining two points in *quadrilateral* (quadruple point set).

5 Recalculation

5.1 Recalculating Circumcircles

For each flip, we must recalculate the corresponding circumcircles and illegal edges.

Algorithm 24 *(Recalculate Circumcircles)* For each flip (P_1,P_3) to (P_2,P_4) in a *quadrilateral*:

1. delete the original circumcircle associated to the *quadrilateral*.
2. construct two new circumcircles defined by (P_3,P_2,P_4) and (P_1,P_2,P_4).
3. **Highlight** accordingly.

5.2 Recalculating Illegal Edges

Recalculate *illegal edges* according to 3 with new circumcircles.

6 References

S. Devadoss, J. O'Rourke, *Discrete and Computational Geometry*, Princeton University Press, 2011.