Implementation of 2-D Voronoi Diagrams & Delaunay Triangulation

Program Outline:-

Text Input

- Enter in the length of the side for a square workspace
- Input the number of points to generate a voronoi diagram

GUI Input

• User clicks on the desired location in the workspace for the input points

Step 1

• Setting up constraints for plotting on workspace (tackle semi-infinite rays)

Step 2

Calculating the perpendicular bisectors

Step 3

- Compute all permutations of circumcentres
- Remove non empty circumcircles

Step 4

 Pruning the perpendicular bisectors to line segments using the information of valid circumcircles

Step 5

 Get the Delaunay Triangulation using the vertex and edges information of Voronoi Diagram.

Output

- Array of voronoi edges in form [x1, y1, x2, y2]
- Output window displaying Voronoi Diagram and Delaunay Triangulation.

Remark:-

The asymptotic complexity of this straight forward method was O (n^4) which is bad compared to other algorithms discussed. They have a complexity ranging from O(n^3) to O (n^2 .log(n))

