Point Location in Planar Subdivisions

Taught By : Prof. Sandeep Sen
Submitted By : Garima Lahoti, 2006MCS2269
Deptt. of Computer Science & Engineering
IIT Delhi

November 30, 2006

1 Introduction

A planar subdivision is a partition of plane into polygonal regions by a finite collection of line segments whose pairwise intersections are restricted to segment end-points.

Point Location Problem: Given a subdivision S with n line segments and a query point P, determine which region of S contains P.

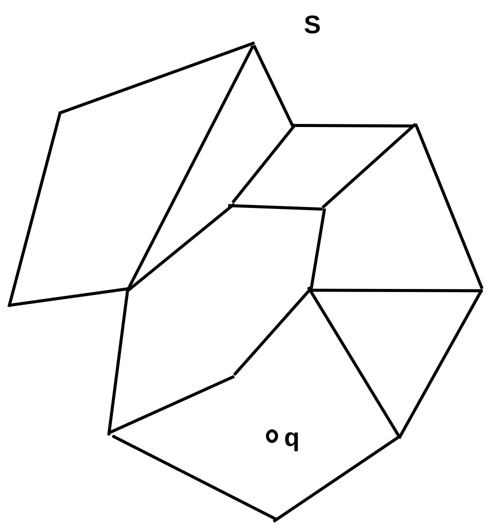

Figure 1: Point Location Problem

Algorithm given by *D.Kirkpatrick* suggests the search to be reduced to triangular subdivision of S. Following section explains the algorithm known as *Kirkpatrick's decompostion* for decomposing planar subdivision into its constituent triangular subdivisions.

2 Kirkpatrick's Decompostion

1. *Triangulate the given planar subdivision* using any of the triangulation methods described in the previous scribe notes.

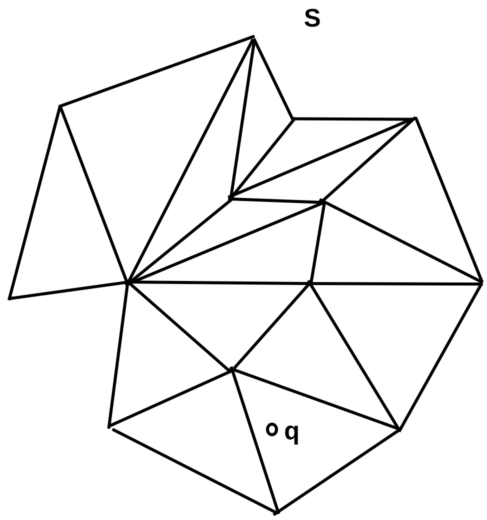

Figure 2: Triangulated Subdivision

2. *Remove a set of internal vertices* (satisfying properties described in Section 4) from the triangulated region. Let v be any internal(non-boundary) vertex in the set and let *deg(v)* denote its degree. Then, exactly *deg(v)* regions of S are incident with v. These regions form neighbourhood of v. Union of these regions form a star-shaped polygonal region with *deg(v)* bounding edges. Now, v and its *deg(v)* edges are removed from S. Similarly remove other vertices in the set.

3. *Re-triangulate* the region obtained by removal of vertices. Regardless of how the neighbourhood of each vertex v is re-triangulated, each new triangle intersects atmost *deg(v)* triangles of S.

4. *Repeat* steps (2) and (3) until the number of vertices left in the subdivision is some small constant.

3 Data Structure

We maintain a Rooted Tree with each level storing constituent triangles of the triangulated subdivision at that level. Root of the tree is a dummy node

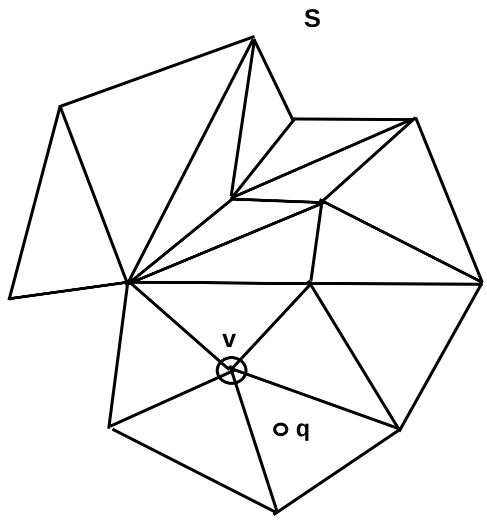

Figure 3: Vertex v to be removed

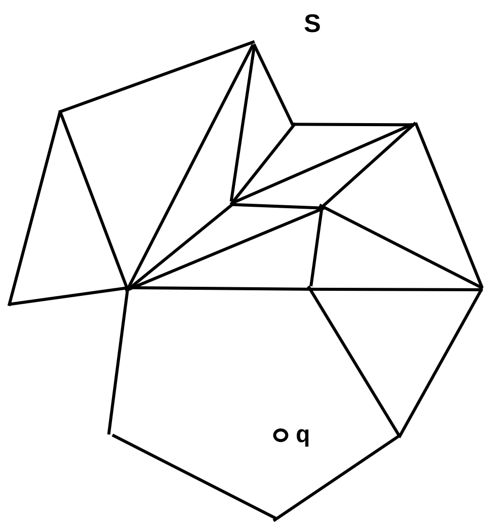

Figure 4: Merged Neighbourhood of v

whose children store the top-level triangulated subdivision with a minimum number of triangles in it. (Obtained in the last step where we stop decompostion). Each node in the tree is a triangle. Each child of an internal node v is the triangle intersected by v in the lower level subdivision.

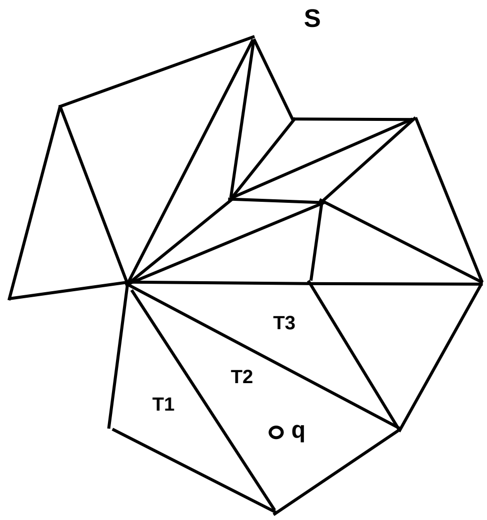

Figure 5: Re-triangulated Neighbourhood of v

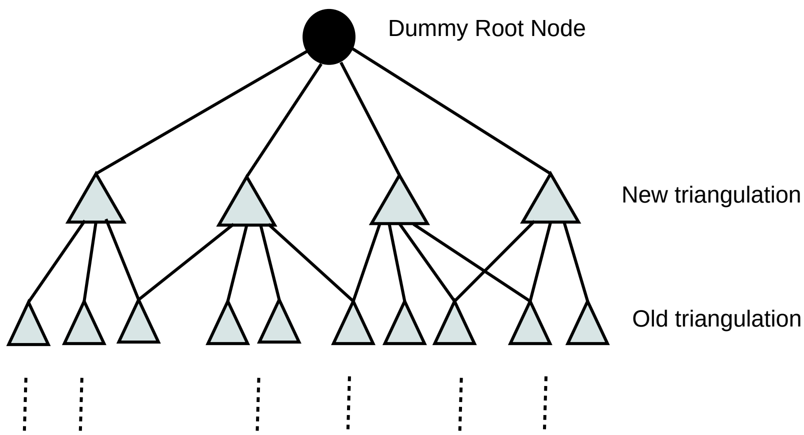

Figure 6: Data Structure for storing intermediate Triangular Subdivisions

4 Desirable Properties

- Out-degree of each triangle in the Tree should be small, i.e, new triangle obtained should not intersect too many old triangles as then our motive of reducing the search space (total regions to be searched) is no more satisfied.
- Maximum path length (height of the tree) should be small.

Strategies followed for removing vertices:

- Remove vertices of constant degree every time.

- Remove large set of such vertices.

- No two such vertices should share an edge, i.e, the vertices should be independent.

- Number of vertices removed should be a constant fraction of n so that the height of tree is small and can be bound to O(log n) to the base some constant.

If w is any vertex which is independent of (nonadjacent to) v in S, then the neighbourhoods of v and w do not intersect except possibly along one or more edges of S. Hence such a pair of vertices can be removed in parallel and the triangular subdivision created by re-triangulating their vacated neighbourhoods has the property that each of its regions intersects atmost max(*deg(v)*,*deg(w)*) regions of S.

We can find the independent set of constant degree vertices by taking a constant degree vertex and throwing away its neighbours and so on. Do we have enough large set of independent vertices? Here is a claim regarding this :

Claim 1 : There are atleast n/2 vertices with degree <= 12.

This claim is justified by another claim which says:

Claim 2 : The average degree of a planar graph <= 6.

Proof: Let E be the total number of edges, F be the total number of triangular regions and V be the total number of vertices. Since each edge is a part of exactly two triangular regions and each triangular region is bounded by 3 edges, we have

$$\begin{aligned} & 2 * E = 3 * F \\ => \quad & \mathrm{F} = (2/3) * \mathrm{E} \end{aligned}$$

From Euler's equation for planar graphs we have

$$\begin{aligned} & V - \mathrm{E} + \mathrm{F} = 2 \\ => \quad & \mathrm{V} - 2 = \mathrm{E} - \mathrm{F} \\ => \quad & \mathrm{V} - 2 = \mathrm{E} \,/\, 3 \\ => \quad & \mathrm{E} <= 3 * \mathrm{V} \end{aligned}$$

Sum of all the degrees in graph = 2 ∗ E <= 6 ∗ V. Thus, average degree of the planar graph is <= 6.
Clearly, Claim 1 follows if Claim 2 is true.

Thus, starting with a set V of vertices of degree atmost 12, a straight forward elimination procedure identifies an independent set containing atleast n/26 vertices which is a constant fraction of n.

5 Query Processing

1. Start with the minimum set of triangles stored as the children of root of the tree.

2. Locate the point in the set of triangles. We get a triangle T containing that point.

3. We move down to lower level in the tree and try to locate point in the children of T. These are the regions(triangles) intersected by the parent triangle T.

4. Repeat steps (2) and (3) until we reach a leaf of the tree. This is the region in the actual (original) planar subdivision that contains the query point.

6 Complexities

1. *Pre-processing Time:* Pre-processing includes:

   - Initial triangulation of the given planar subdivision which requires O(n lg n) time.
   - Construction of tree containing set of triangular regions at each stage of elimination of independent vertices in the subdivision at that time. Since a vertex, if removed, is removed only once and there are atmost 2n vertices. Each vertex v removal requires us to re-triangulate its merged neighbourhood region. This new region to be triangulated contains atmost *deg(v)* vertices which is a constant. Thus, such a triangulation can be performed in a constant time. Thus, total time taken in this step is O(n).

   So, total pre-processing time is O(n lg n).

2. *Space:* Space required is O(n).

3. *Search Time:* There are O(log n) levels in the tree. Membership in any triangular region can be tested in constant time. Since we construct tree such that at each level we try to minize the candidate triangular regions to be searched (*refer Section on Desirable Properties*). Thus, query time is O(height) = O(log n).