Planar Point Location

Search Problem!

$N=$ Straight Line Planar Map/ Polygonal Subdivision.

$\tilde{H}(N) =$ A search structure to answer queries/
Given a query point, which face of the planar subdivision it belongs to?

Building the Data Structure - $O(n)$-time.

Query Time - $O(\log n)$ time.

Kirkpatrick's Method

Assume that the Planar Graph is Triangulated and the outer face is a triangle.

If not, there are various ways to triangulate it, e.g., using trapezoidal decomposition.

We report which triangle the query point lies.

Basic Idea

Let T_0 be the initial triangle.

1. Construct a Hierarchy of triangles: $T_0, T_1, \cdots, T_k$, where $k = O(\log n)$ and $T(k)$ consists of only 1 triangle, i.e., the exterior face.

2. Each triangle in T_{i+1} overlaps a constant number of triangles in T_i.

3. Preserve Planarity.

Basic Observation

1. Vertices deleted will have a *constant degree* ($\leq d = 8$). The deletion of such vertices creates a hole consisting of at most $d - 2$ triangles. In the retriangulation, each of the new triangle can overlap at most d triangles of the previous triangulation.

2. No two vertices that will be deleted in a stage of the algorithm will be adjacent. Therefore, vertices form an independent set and therefore the holes are independent!

Is this possible?

Theorem: Given a planar graph with n vertices, there is an independent set consisting of vertices of degree at most 8, with at least $n/18$ vertices. The independent set can be computed in linear time.

Proof:

1. Euler's Formula for triangulated planar graphs, $e = 3n - 6$.

2. Average degree in the graph $\sum_v deg(v) = 2e = 6n - 12 < 6n$.

3. Therefore, average degree is about 6.

4. There exists at least $n/2$ vertices of degree ≤ 8.

5. Compute an independent set as follows.

   (a) Mark all vertices whose $degree > 8$.
   (b) While *there is an unmark vertex* v do
       - Append v to the set of independent vertices.
       - Mark v and all of its neighbors.

6. At least $n/18$ vertices in the independent set.

7. Algorithm runs in Linear Time.

How to Answer the Query

A tree (DAG) of Triangles

Let q be the query point.

1. Root of the tree is triangle T_k, corresponding to the external face.
   If $q \notin T_k$ then *q is out of range*.

2. $\{q \in T_k\}$ Each node in T_i points to nodes in T_{i-1}, i.e., each triangle node in T_i points to all triangles it overlaps in T_{i-1}.
   Find out which triangle in the next level the point q lies.
   Continue this search, till we reach the triangle of the original planar subdivision.

Analysis

Query Time: $O(\log n)$

1. Tree has $O(\log n)$ levels.

2. It takes $O(1)$ time to move from one level to the next.

Size of the Data Structure: $O(n)$

$$n(1 + 17/18 + (17/18^2) + \cdots) \leq 18n$$

Constructing the Data Structure: $O(n)$.

Theorem

An n-vertex planar straightline subdivision can be preprocessed in linear time such that point location queries can be answered in $O(\log n)$ time.

Is this practical?