Curve & Surface Reconstruction - on Slides.

Medial axis

= Voronoi diagram of edges of a polygon

points in this region are closer to e

point on Voronoi edge is equally close to two polygon edges.

Voronoi vertex is equally close to 3 polygon edges

= center of circle inside polygon, touching 3 polygon edges.

see slides
Medial axis of convex polygon can be found in $O(n)$ time.

Here is an $O(n \log n)$-time alg:

Maintain:
- for each pair of consecutive edges, their angle bisector rays
- for each pair of consecutive rays, their meeting point and time $t$ when meet
- a priority queue of these times.

Update:
- find min. time, output Voronoi point
- delete $e_i$ and 2 rays
- find new ray, bisector of $e_{i-1}, e_{i+1}$, compute its meeting point with 2 neighbour rays and add their times to priority queue.

---

Line Arrangements

Intro-problem: Recall the problem: given $n$ points, are there 3 (or more) collinear? Obvious alg: $O(n^3)$

By point-line duality this becomes:
given $n$ lines, do 3 of them intersect at a point.

Will solve this using line arrangements
n lines partition plane into edges, faces, vertices
Call this a line arrangement.

**Lemma** n lines, no 2 parallel, no 3 through same vertex \( \Rightarrow \binom{n}{2} \) vertices, \( n^2 \) edges, \( \binom{n+1}{2} + 1 \) faces
With degeneracies, these counts decrease.

**Proof**

Vertices: each pair of lines intersects, so \( \binom{n}{2} \) vertices

Edges: each line is cut by \( n-1 \) lines into \( n \) edges so \( n^2 \) edges

Faces: \( f_n = f_{n-1} + n + f_0 = 1 \)

since last line cuts \( n \) faces in two

\[ f_n = n + n - 1 + 1 + f_0 = \binom{n+1}{2} + 1 \]
Constructing arrangements

Input: \( n \) lines
Output: list of faces, edges, vertices and all incidence relationships

Note: size is \( \Theta(n^2) \)

There is an \( O(n^2) \) time alg.

First note that plane sweep would give \( O(n^2 \log n) \)
because there are \( n^2 \) events (the vertices, where lines cross) and we need a priority queue to
store them. \( O(\log n^2) = O(\log n) \) per operation.

Incremental algorithm — add lines one by one.
- find intersection point \( x \) of \( l_n \) with some other line (say \( l_1 \))
- find edge \( e \) of \( l_1 \) that contains \( x \) and face \( f \) that contains \( e \)
- walk around edges of \( f \) (start at \( e \)) to find where \( l_n \) leaves \( f \) and enters a new face
- continue walking faces until you return to \( x \)
- update arrangement as you go. "zone" of \( l_n \)

Time taken = \( O(\text{# edges in faces cut by } l_n) \)

We will show this is \( O(n) \) "Zone Theorem"!

- see slides