Computational Geometry

The Post-Office Problem

Lecture #7

[Comp. Geom A&A : Chapter 7]
The Post-Office Problem

Tasks:  
1) Define Voronoi cells, edges and vertices!  
2) Are Voronoi cells convex?
The Voronoi diagram

Let \( P \) be a set of points in the plane and let \( p, p', p'' \in P \).

[Voronoi diagram] \[
\text{V}(\{p\}) = \bigcap_{q \neq p} h(p, q)
\]

[Voronoi cell] \[
\text{V}(\{p\}) = \left\{ x \in \mathbb{R}^2 : |xp| < |xq| \text{ for all } q \in P \setminus \{p\} \right\} \\
= \bigcap_{q \neq p} h(p, q)
\]

[Voronoi edge] \[
\text{V}(\{p, p'\}) = \left\{ x : |xp| = |xp'| \text{ and } |xp| < |xq| \text{ for all } q \neq p, p' \right\} \\
= \text{rel-int} \left( \partial \text{V}(p) \cap \partial \text{V}(p') \right), \text{ i.e., w/o the endpts}
\]

[Voronoi vertex] \[
\text{V}(\{p, p', p''\}) = \partial \text{V}(p) \cap \partial \text{V}(p') \cap \partial \text{V}(p'')
\]
Overall Shape of Vor(P)

**Theorem.** Let \( P \subset \mathbb{R}^2 \) be a set of \( n \) pts (called sites). If all sites are collinear, \( \text{Vor}(P) \) consists of \( n - 1 \) parallel lines. Otherwise, \( \text{Vor}(P) \) is connected and its edges are line segments or half-lines.

**Proof.** Assume that \( P \) is not collinear.
- Assume that \( \text{Vor}(P) \) contains an edge \( e \) that is a full line, say, \( e = b(p, q) \).

Let \( r \in P \) be not collinear with \( p \) and \( q \). Then \( b(q, r) \) is not parallel to \( e \).
\[ \Rightarrow e \cap h(r, q) \text{ is closer to } r \text{ than to } p \text{ or } q. \]
\[ \Rightarrow e \text{ is bounded on at least one side.} \]
Complexity

Task: Construct a set $P$ of point sites such that $\text{Vor}(P)$ has a cell of linear complexity!

Theorem. Given a set $P \subset \mathbb{R}^2$ of $n$ sites, $\text{Vor}(P)$ consists of at most $2n - 5$ vertices and $3n - 6$ edges.

Proof. Problem: unbounded edges!

$\Rightarrow$ can’t apply Euler directly, but...
Characterization of Voronoi vtc and edges

\[ C_P(x) := \text{largest circle centered at } x \text{ w/o sites in its interior} \]

**Theorem:**

(i) \( x \) Voronoi vtx \( \iff \) \( |C_P(x) \cap P| \geq 3 \)

(ii) \( b(p, p') \) contains a Voronoi edge \( \iff \) \( \exists x \in b(p, p') : C_P(x) \cap P = \{p, p'\} \)
**Computation**

**Brute force:** For each \( p \in P \), compute \( V(p) = \bigcap_{p' \neq p} h(p, p') \).

\[ V(p) = \bigcap_{p' \neq p} h(p, p') \]

- in total: \( O(n^2 \log n) \) time
- but the complexity of \( \text{Vor}(P) \) is linear!

**Sweep?**

Problem: We don’t know all defining sites yet :(
Sweep?

Which part of the plane above $\ell$ is fixed by what we’ve seen?

Solution:

$f_p^\ell$ is the parabola with focus $p$ and directrix $\ell$.

Task:

Compute $f_p^\ell$ for $p = (0, 1)$ and $\ell$: $y = -1$!

Definition.

*beachline* $\beta \equiv$ lower envelope of $(f_p^\ell)_{p \in P \cap \ell^+}$

Observation.

$\beta$ is $x$-monotone.
The beachline $\beta$

**Question:** What does $\beta$ have to do with $\operatorname{Vor}(P)$?

**Answer:** “Breakpoints” of $\beta$ trace out the Voronoi edges!

**Lemma.** New arcs on $\beta$ only appear through site events, that is, whenever $\ell$ hits a new site.

**Corollary.** $\beta$ consists of at most $2n - 1$ arcs.

**Definition.** *Circle event:* $\ell$ reaches lowest pt of a circle through three sites above $\ell$ whose arcs are consecutive on $\beta$.

**Lemma.** Arcs disappear from $\beta$ only at circle events.

**Lemma.** The Voronoi vtc correspond 1:1 to circle events.
Fortune’s Sweep

VoronoiDiagram\( (P \subset \mathbb{R}^2) \)

\[
Q \leftarrow \text{new PriorityQueue}(P) \quad \text{// site events sorted acc. \(y\)-coord.}
\]

\[
T \leftarrow \text{new BalancedBinarySearchTree()} \quad \text{// sweep status (\(\beta\))}
\]

\[
D \leftarrow \text{new DCEL()} \quad \text{// to-be Vor}(P)
\]

while not \(Q\).empty() do

\[
p \leftarrow Q.\text{ExtractMax()}
\]

if \(p\) site event then

\[
\text{HandleSiteEvent}(p)
\]

else

\[
\alpha \leftarrow \text{arc on } \beta \text{ that will disappear}
\]

\[
\text{HandleCircleEvent}(\alpha)
\]

\[
treat \text{ remaining internal nodes of } T \ (\equiv \text{unbnd. edges of Vor}(P))
\]

return \(D\)
Handling Events

HandleSiteEvent(point \( p \))

- Search in \( T \) for the arc \( \alpha \) vertically above \( p \).
  If \( \alpha \) has pointer to circle event in \( Q \), delete this event.

- Split \( \alpha \) into \( \alpha_0 \) and \( \alpha_2 \).
  Let \( \alpha_1 \) be the new arc of \( p \).

- Add Vor-edges \( \langle q, p \rangle \) and \( \langle p, q \rangle \) to DCEL.

- Check \( \langle \cdot, \alpha_0, \alpha_1 \rangle \) and \( \langle \alpha_1, \alpha_2, \cdot \rangle \) for circle events.

HandleCircleEvent(arc \( \alpha \))

- \( T \).delete(\( \alpha \)); update breakpts

- Delete all circle events involving \( \alpha \) from \( Q \).

- Add Vor-vtx \( \alpha_{\text{left}} \cap \alpha_{\text{right}} \) and Vor-edge \( \langle \alpha_{\text{left}}, \alpha_{\text{right}} \rangle \) to DCEL.

- Check \( \langle \cdot, \alpha_{\text{left}}, \alpha_{\text{right}} \rangle \) and \( \langle \alpha_{\text{left}}, \alpha_{\text{right}}, \cdot \rangle \) for circle events.

Running time? \( O(\log n) \) per event...
Running Time?

VoronoiDiagram($P \subset \mathbb{R}^2$)

$Q \leftarrow$ new PriorityQueue($P$)  // site events sorted acc. $y$-coord.
$T \leftarrow$ new BalancedBinarySearchTree()  // sweep status ($\beta$)
$D \leftarrow$ new DCEL()  // to-be Vor($P$)

while not $Q$.empty() do
    $p \leftarrow Q$.ExtractMax()

    if $p$ site event then
        HandleSiteEvent($p$)  // exactly $n$ such events
    else
        $\alpha \leftarrow$ arc on $\beta$ that will disappear
        HandleCircleEvent($\alpha$)  // at most $2n - 5$ such events

    treat remaining internal nodes of $T$ ($\equiv$ unbd. edges of Vor($P$))

return $D$
Summary

**Theorem.** Given a set $P$ of $n$ pts in the plane, Fortune’s sweep computes $\text{Vor}(P)$ in $O(n \log n)$ time and $O(n)$ space.

Steven Fortune
Bell Labs