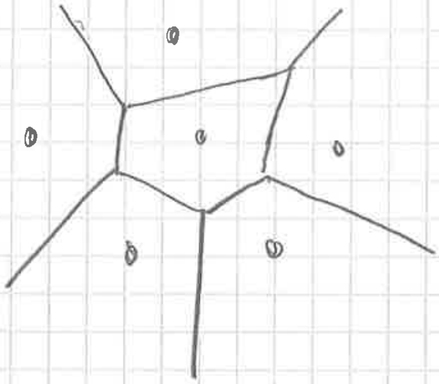
Today : Computation of Voronoi diagrams



Vor(P)

for $P = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$

Direct approach:

Find each Voronoi region separately

Not very efficient : $O(n^2 \log n)$

Several algorithms known; for example:

Hoey 1975 : Divine-and-Conquer algorithm $O(n \log n)$

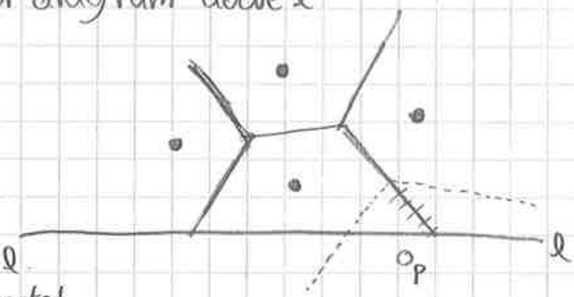
Green-Sibson 1977 : Incremental alg. $O(n^2)$

Fortune 1985 : sweep-line alg. $O(n \log n)$.

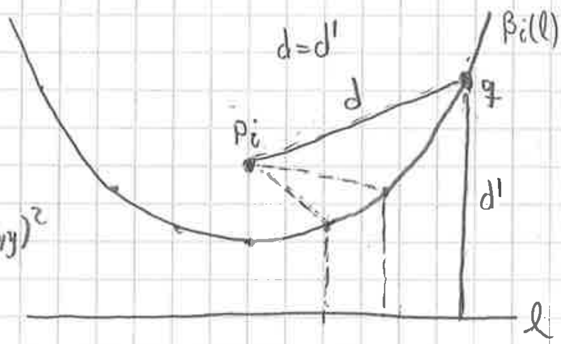
Today : sketch of Fortune's sweep-line algorithm.

Idea : Move a horizontal line ℓ down and construct the Voronoi diagram above ℓ

But : Vor(P) might be affected by sites not visible yet (below ℓ)



Solution : Identify the region above ℓ in which Vor(P) can not be affected by sites below ℓ.



$$(y - \ell_y)^2 = (x - p_{i,x})^2 + (y - p_{i,y})^2$$

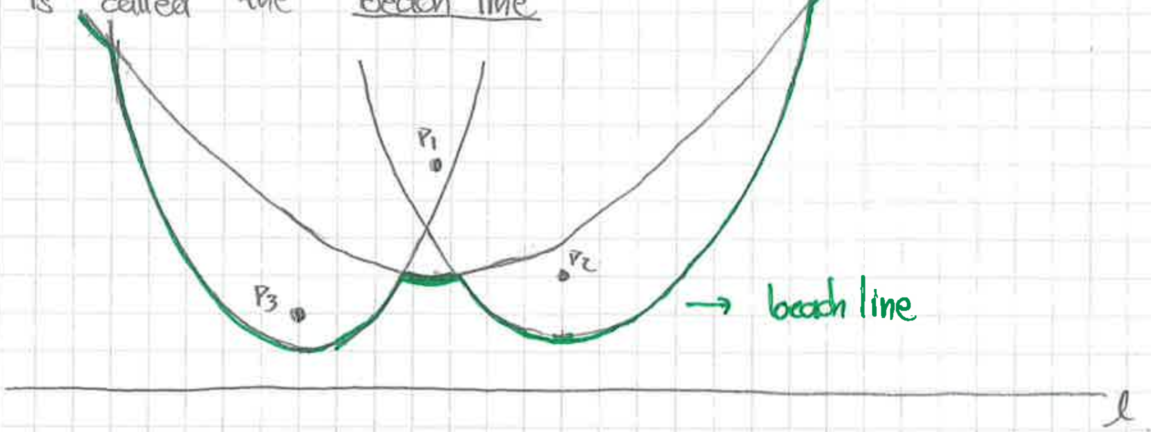
Given a site p_i and the line ℓ define the parabola

$$\beta_i(\ell) := \{ q \in \mathbb{R}^2 \mid dist(q, p_i) = dist(q, \ell) \}$$

Observations :
- all points above $\beta_i(\ell)$ are closer to p_i than to ℓ
- Everything of Vor(P) above the parabolas $\beta_i(\ell)$, for the sites p_i above ℓ, will not change as ℓ moves down.

At any step of the sweeping process, we consider the line ℓ and the union of the parabolas $\beta_i(\ell)$ for the points p_i above ℓ

The sequence of parabolic arcs that you can see from below is called the <u>beach line</u>

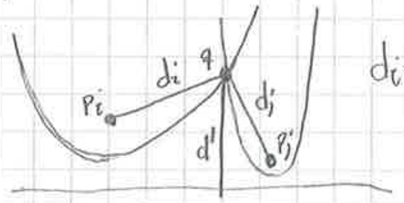


Key of the algorithm : $\boxed{\text{Break points on the beach line trace out the Voronoi diagram while sweep line progresses}}$

Watch video !

In fact : a point $q \in \beta_i(\ell) \cap \beta_j(\ell)$ satisfies

q is equidistant to p_i and p_j $\Leftarrow$ $d(q,p_i) = d(q,\ell) = d(q,p_j)$

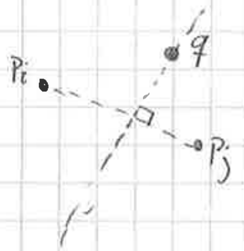


$d_i = d = d_j$

If the other parabolas are (weakly) above q then

$$q \in \mathcal{V}(p_i) \quad \text{and} \quad q \in \mathcal{V}(p_j)$$

That is, the other sites are farther to q than to p_i and p_j.



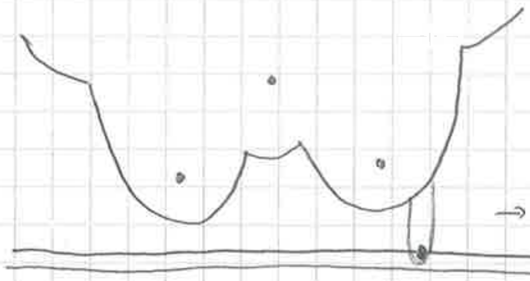
So q lies on the edge of Vor(P) defined by the bisector ℓ_{ij}.

Moreover, if a point q in the beach line is the intersection of 3 (or more) parabolas $\beta_i(\ell), \beta_j(\ell), \beta_k(\ell)$, then q is a vertex of Vor(P) shared by the Voronoi regions $\mathcal{V}(p_i), \mathcal{V}(p_j), \mathcal{V}(p_k)$.

As the sweep line moves down, the beach lines may change in two possible ways:

① Site event:

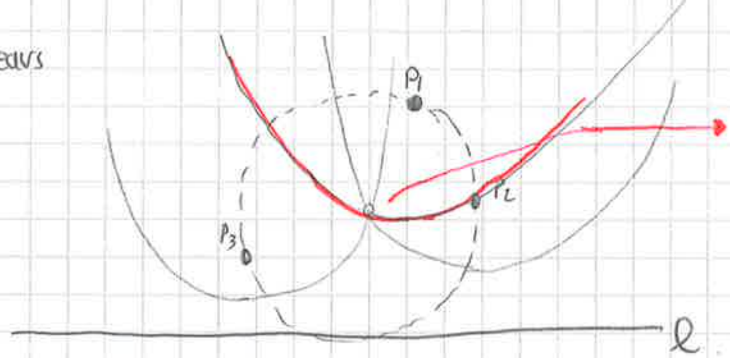
Sweep line moves over a new site



new parabolic arc splits another arc into two pieces

② Circle event

A parabolic arc disappears from the beach line



happens when circumcircle of 3 involved sites is tangent to ℓ.

Theorem The Voronoi diagram can be computed in $O(n \log n)$ time using $O(n)$ storage.

Algorithm:

- Maintain the beach line combinatorially (balanced binary search tree)
- Create a queue with
  - site events (know from the beginning)
  - circle events (for 3 consecutive arcs on the beach line)
- Update at event points, and construct Vor(P) "on the fly"

\+ many details.

Analysis sketch:

$$\left.\begin{array}{l} \text{time to process an event} : O(\log n) \\ \#\text{ site events} = n \\ \#\text{ circle events} = \#\text{ vertices of } \mathrm{Vor}(P) \leq 2n-5 \end{array}\right\} O(n \log n).$$