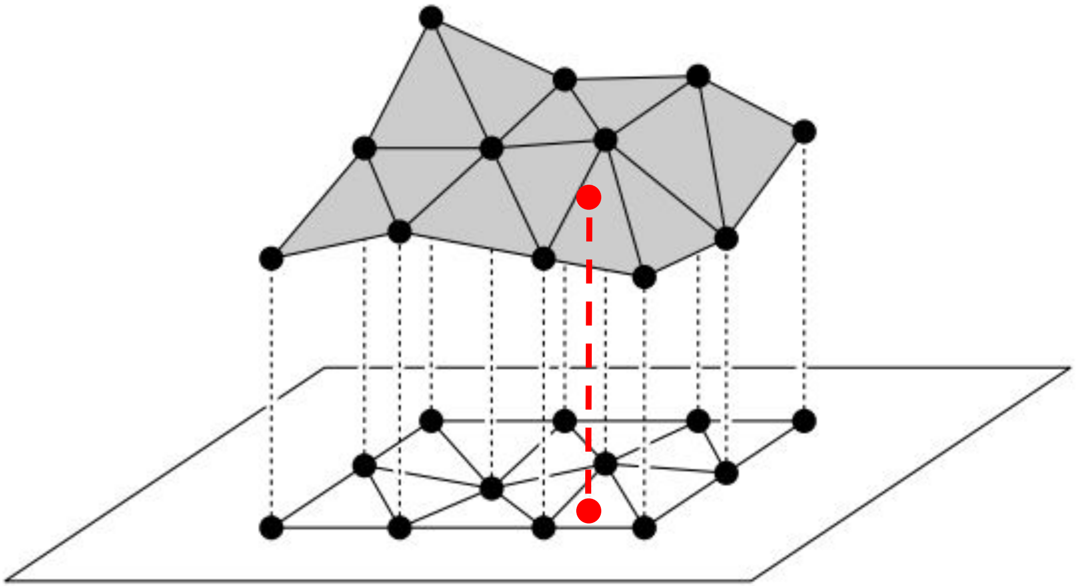
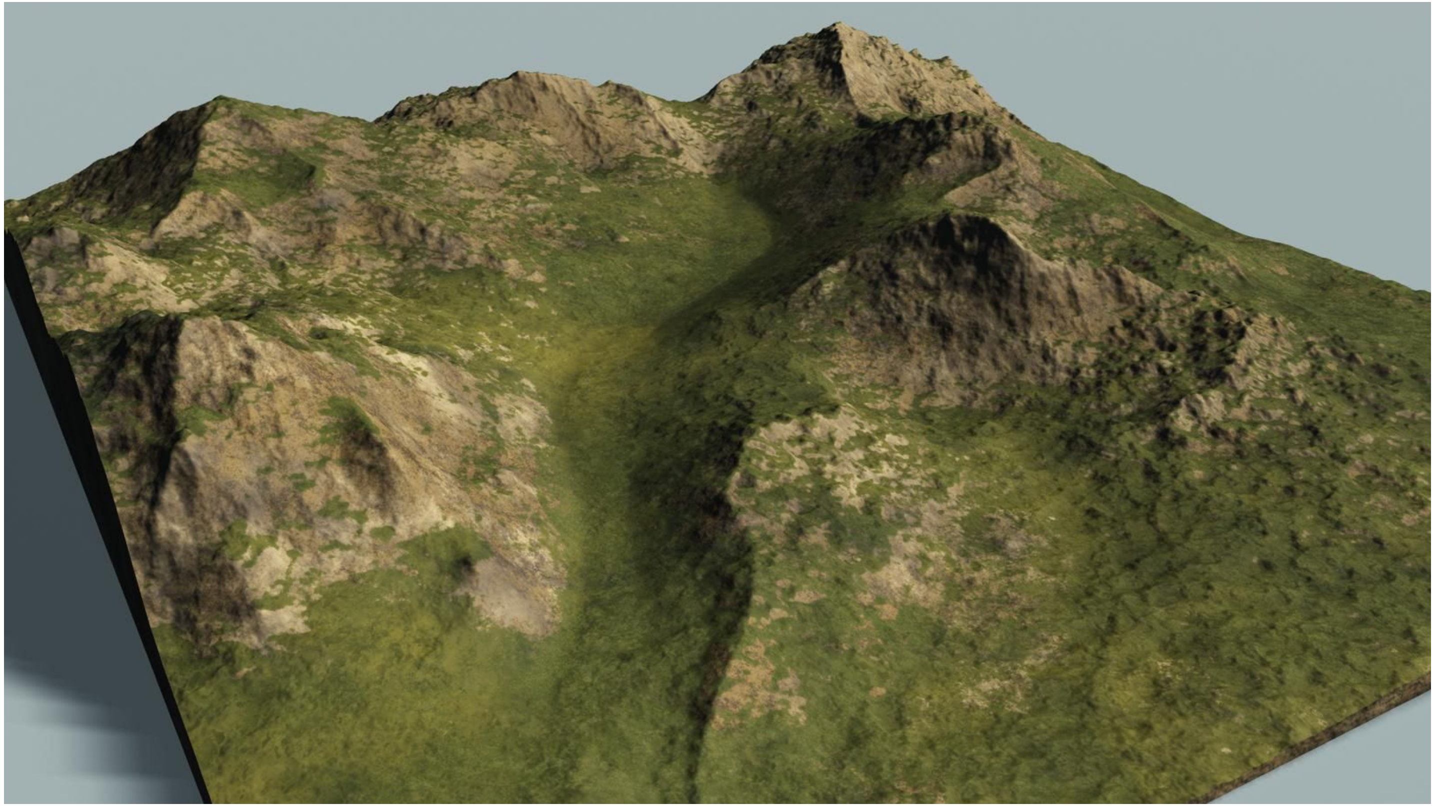
Delaunay Triangulations

Computational Geometry - Algorithms and Applications (Ch. 9)

Problem

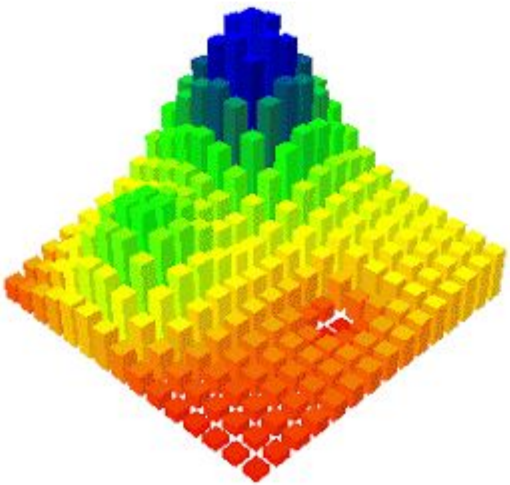


Motivation

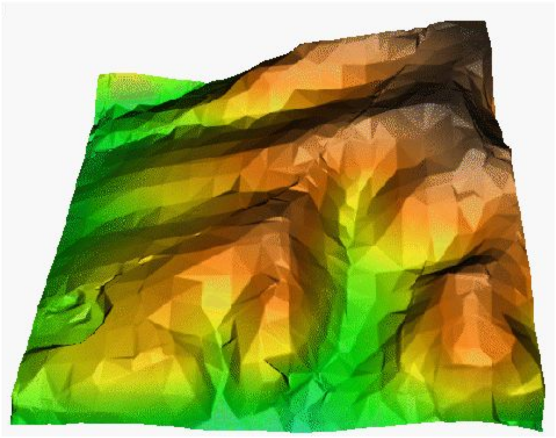


Approaches

Nearest Neighbor

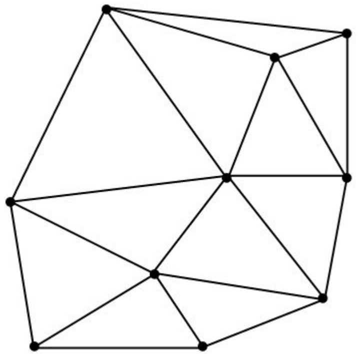


Triangulation

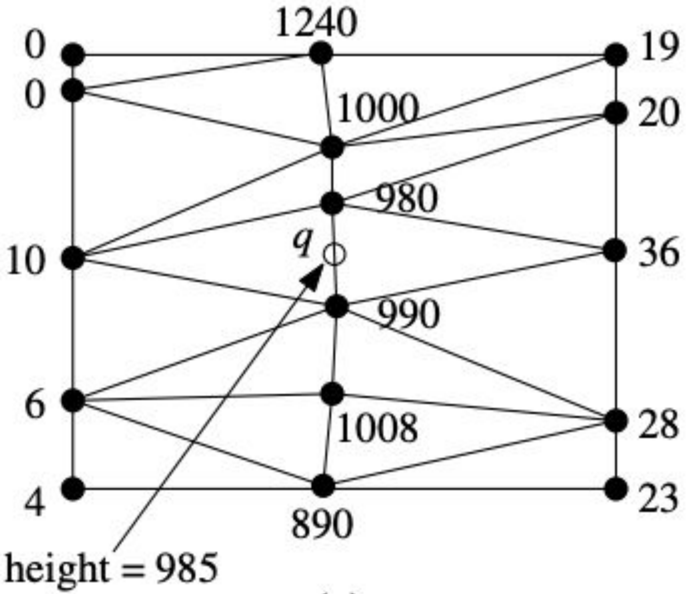


Definitions

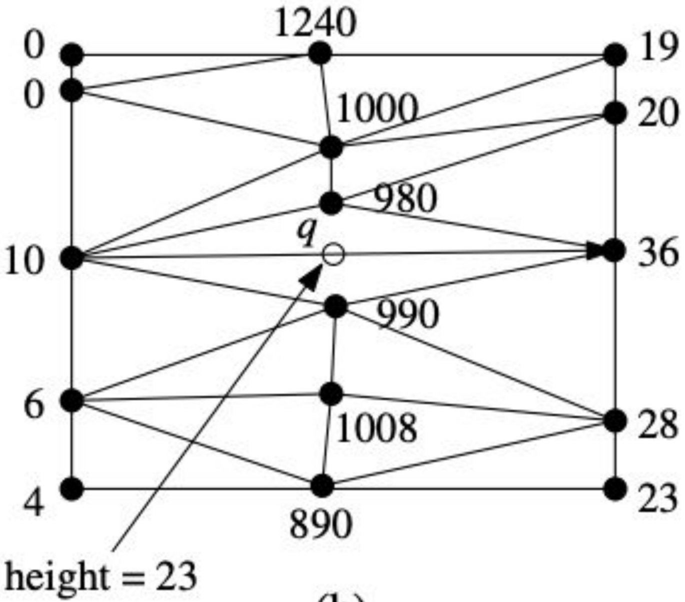
- $P = \{p_1, p_2, \ldots, p_n\}$:= set of reference points in the plane
- Triangulation := maximal planar graph of a given vertex set



How to pick good triangulation?



(a)



(b)

Small angles are bad

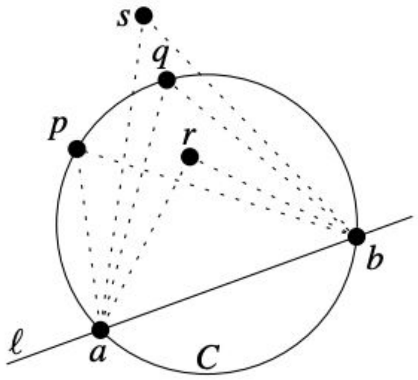
Angle Vectors

- A(T) := sorted list of all angles in increasing order
- Angle-optimal triangulation has largest lexicographical angle vector

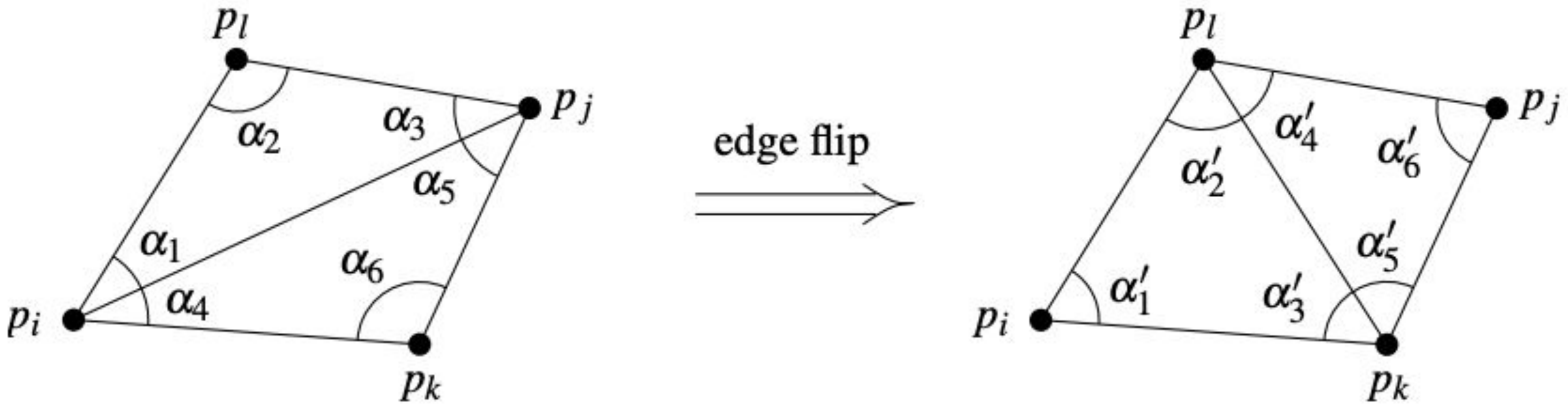
Thales Theorem

Theorem 9.2 *Let C be a circle, ℓ a line intersecting C in points a and b, and p, q, r, and s points lying on the same side of ℓ. Suppose that p and q lie on C, that r lies inside C, and that s lies outside C. Then*

$$\measuredangle arb > \measuredangle apb = \measuredangle aqb > \measuredangle asb.$$



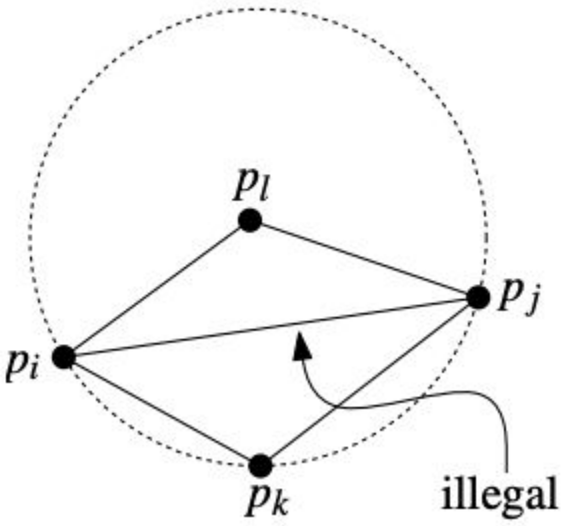
Illegal Edges



$$\min_{1\leqslant i\leqslant 6} \alpha_i \;<\; \min_{1\leqslant i\leqslant 6} \alpha'_i.$$

Illegal Edges

Observation 9.3



Simple Algorithm

Algorithm LEGALTRIANGULATION($\mathcal{T}$)

Input. Some triangulation $\mathcal{T}$ of a point set P.

Output. A legal triangulation of P.

while $\mathcal{T}$ contains an illegal edge $\overline{p_i p_j}$

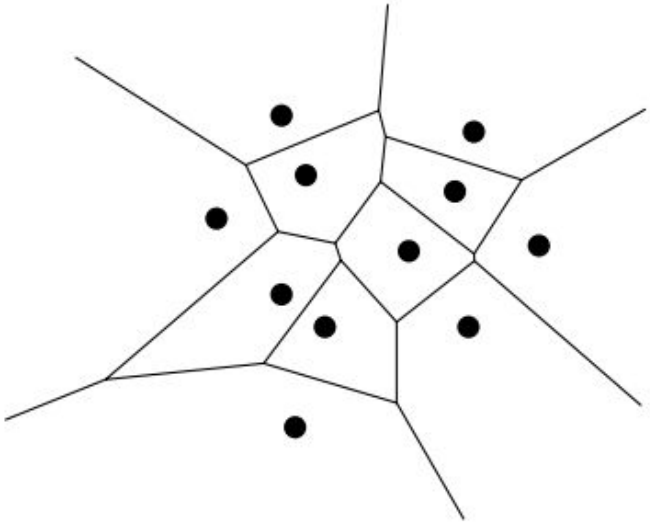
  do (∗ Flip $\overline{p_i p_j}$ ∗)

    Let $p_i p_j p_k$ and $p_i p_j p_l$ be the two triangles adjacent to $\overline{p_i p_j}$.

    Remove $\overline{p_i p_j}$ from $\mathcal{T}$, and add $\overline{p_k p_l}$ instead.

return $\mathcal{T}$

Voronoi diagram

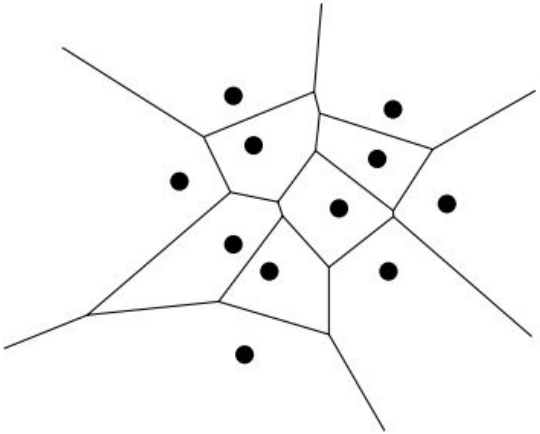


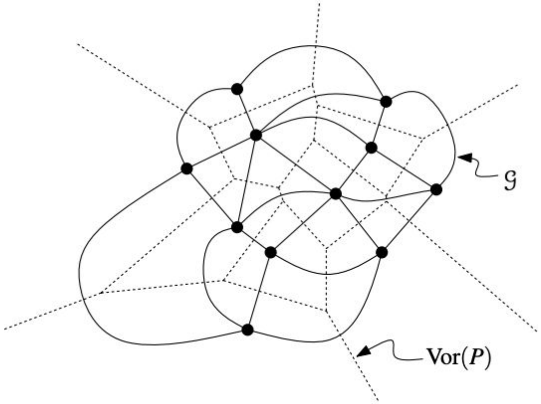
Theorem 7.4 *For the Voronoi diagram* Vor(P) *of a set of points P the following holds:*

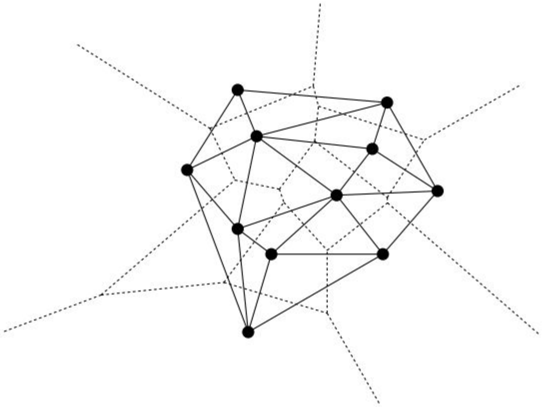
(i) *A point q is a vertex of* Vor(P) *if and only if its largest empty circle $C_P(q)$ contains three or more sites on its boundary.*

(ii) *The bisector between sites p_i and p_j defines an edge of* Vor(P) *if and only if there is a point q on the bisector such that $C_P(q)$ contains both p_i and p_j on its boundary but no other site.*

Delaunay Graph







Theorem 9.5 *The Delaunay graph of a planar point set is a plane graph.*

General Position

- All planes of DG(P) are triangles if no four points of P lie on a circle
- Delaunay Triangulation (may or may not need to add edges to DG(P))

Theorem 9.6 *Let P be a set of points in the plane.*

(i) Three points $p_i, p_j, p_k \in P$ are vertices of the same face of the Delaunay graph of P if and only if the circle through p_i, p_j, p_k contains no point of P in its interior.

(ii) Two points $p_i, p_j \in P$ form an edge of the Delaunay graph of P if and only if there is a closed disc C that contains p_i and p_j on its boundary and does

Theorem 9.6 readily implies the following characterization of Delaunay triangulations.

Theorem 9.7 *Let P be a set of points in the plane, and let $\mathcal{T}$ be a triangulation of P. Then $\mathcal{T}$ is a Delaunay triangulation of P if and only if the circumcircle of any triangle of $\mathcal{T}$ does not contain a point of P in its interior.*

Theorem 9.8 *Let P be a set of points in the plane. A triangulation $\mathcal{T}$ of P is legal if and only if $\mathcal{T}$ is a Delaunay triangulation of P.*

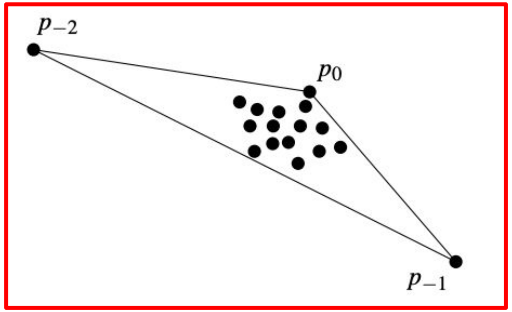
Theorem 9.9 *Let P be a set of points in the plane. Any angle-optimal triangulation of P is a Delaunay triangulation of P. Furthermore, any Delaunay triangulation of P maximizes the minimum angle over all triangulations of P.*

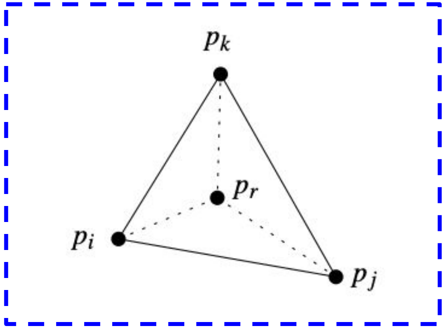
Computing the Delaunay Triangulation

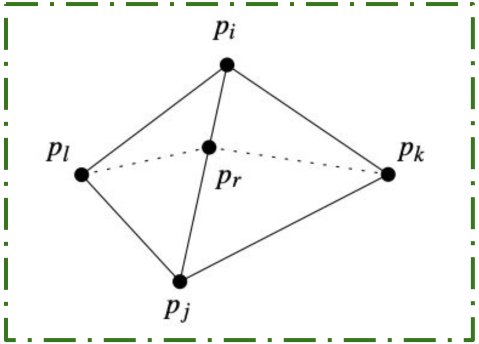
- Randomized, incremental approach

Algorithm DELAUNAYTRIANGULATION(P)
Input. A set P of $n+1$ points in the plane.
Output. A Delaunay triangulation of P.

Let p_0 be the lexicographically highest point of P, that is, the rightmost among the points with largest y-coordinate.
Let p_{-1} and p_{-2} be two points in $\mathbb{R}^2$ sufficiently far away and such that P is contained in the triangle $p_0 p_{-1} p_{-2}$.
Initialize $\mathcal{T}$ as the triangulation consisting of the single triangle $p_0 p_{-1} p_{-2}$.
Compute a random permutation $p_1, p_2, \ldots, p_n$ of $P \setminus \{p_0\}$.
for $r \leftarrow 1$ **to** n
  do (∗ Insert p_r into $\mathcal{T}$: ∗)
    Find a triangle $p_i p_j p_k \in \mathcal{T}$ containing p_r.
    if p_r lies in the interior of the triangle $p_i p_j p_k$
      then Add edges from p_r to the three vertices of $p_i p_j p_k$, thereby splitting $p_i p_j p_k$ into three triangles.
        LEGALIZEEDGE($p_r, \overline{p_i p_j}, \mathcal{T}$)
        LEGALIZEEDGE($p_r, \overline{p_j p_k}, \mathcal{T}$)
        LEGALIZEEDGE($p_r, \overline{p_k p_i}, \mathcal{T}$)
      else (∗ p_r lies on an edge of $p_i p_j p_k$, say the edge $\overline{p_i p_j}$ ∗)
        Add edges from p_r to p_k and to the third vertex p_l of the other triangle that is incident to $\overline{p_i p_j}$, thereby splitting the two triangles incident to $\overline{p_i p_j}$ into four triangles.
        LEGALIZEEDGE($p_r, \overline{p_i p_l}, \mathcal{T}$)
        LEGALIZEEDGE($p_r, \overline{p_l p_j}, \mathcal{T}$)
        LEGALIZEEDGE($p_r, \overline{p_j p_k}, \mathcal{T}$)
        LEGALIZEEDGE($p_r, \overline{p_k p_i}, \mathcal{T}$)
Discard p_{-1} and p_{-2} with all their incident edges from $\mathcal{T}$.
return $\mathcal{T}$







LEGALIZEEDGE(p_r, $\overline{p_i p_j}$, $\mathcal{T}$)

(* The point being inserted is p_r, and $\overline{p_i p_j}$ is the edge of $\mathcal{T}$ that may need to be flipped. *)

if $\overline{p_i p_j}$ is illegal

    then Let $p_i p_j p_k$ be the triangle adjacent to $p_r p_i p_j$ along $\overline{p_i p_j}$.

        (* Flip $\overline{p_i p_j}$: *) Replace $\overline{p_i p_j}$ with $\overline{p_r p_k}$.

        LEGALIZEEDGE(p_r, $\overline{p_i p_k}$, $\mathcal{T}$)

        LEGALIZEEDGE(p_r, $\overline{p_k p_j}$, $\mathcal{T}$)

Correctness

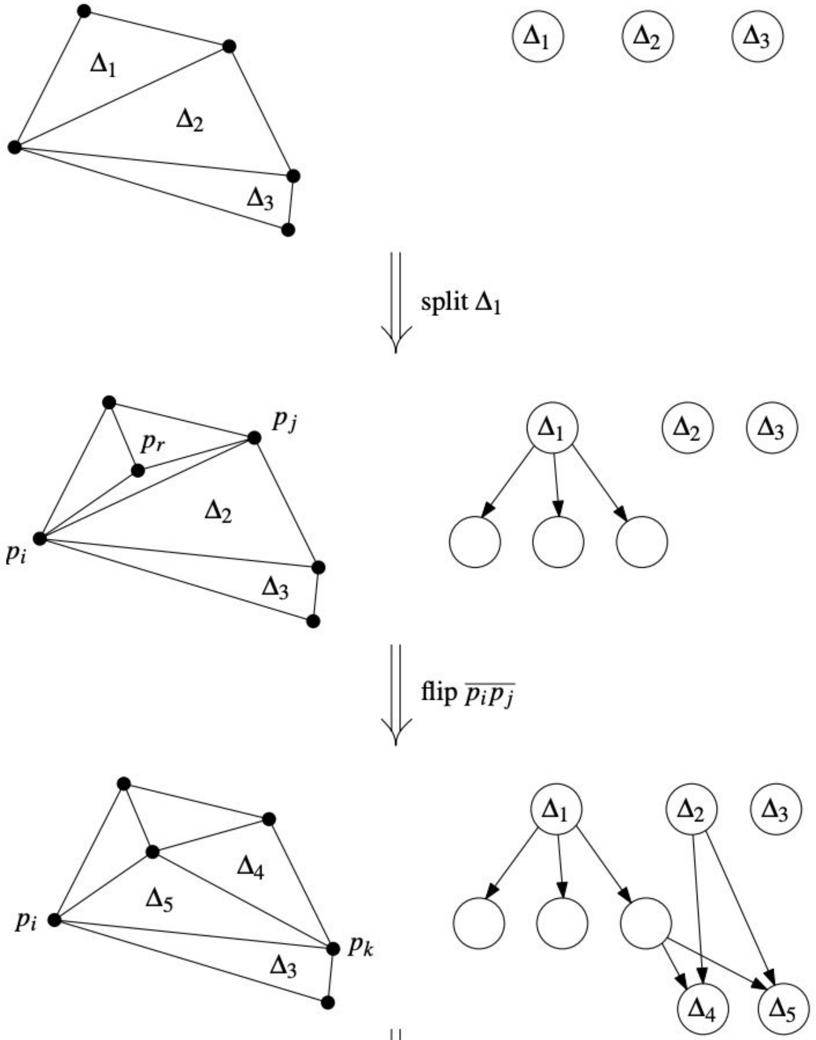
Lemma 9.10 *Every new edge created in* DELAUNAYTRIANGULATION *or in* LEGALIZEEDGE *during the insertion of* p_r *is an edge of the Delaunay graph of* $\{p_{-2}, p_{-1}, p_0, \ldots, p_r\}$.

Implementation Details

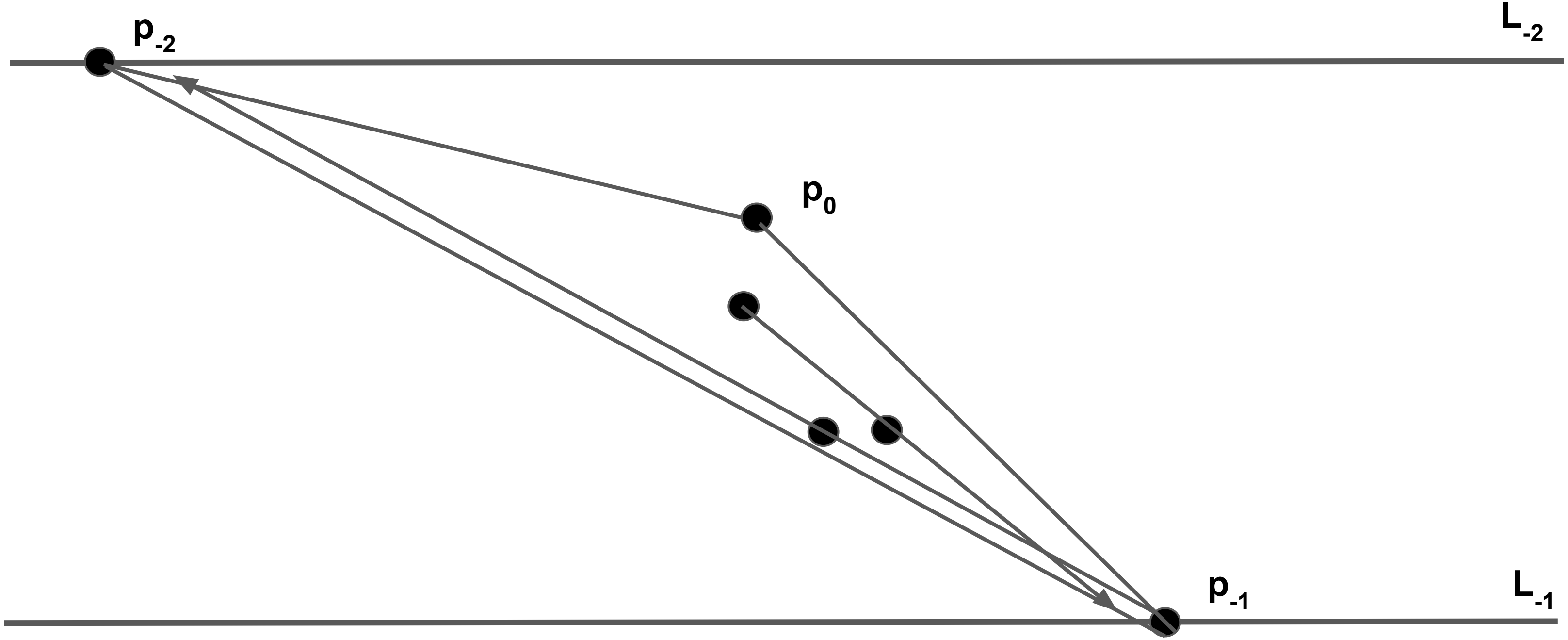
- Finding the triangle that contains a point
- Dummy nodes p_{-1} and p_{-2}

Finding Triangle

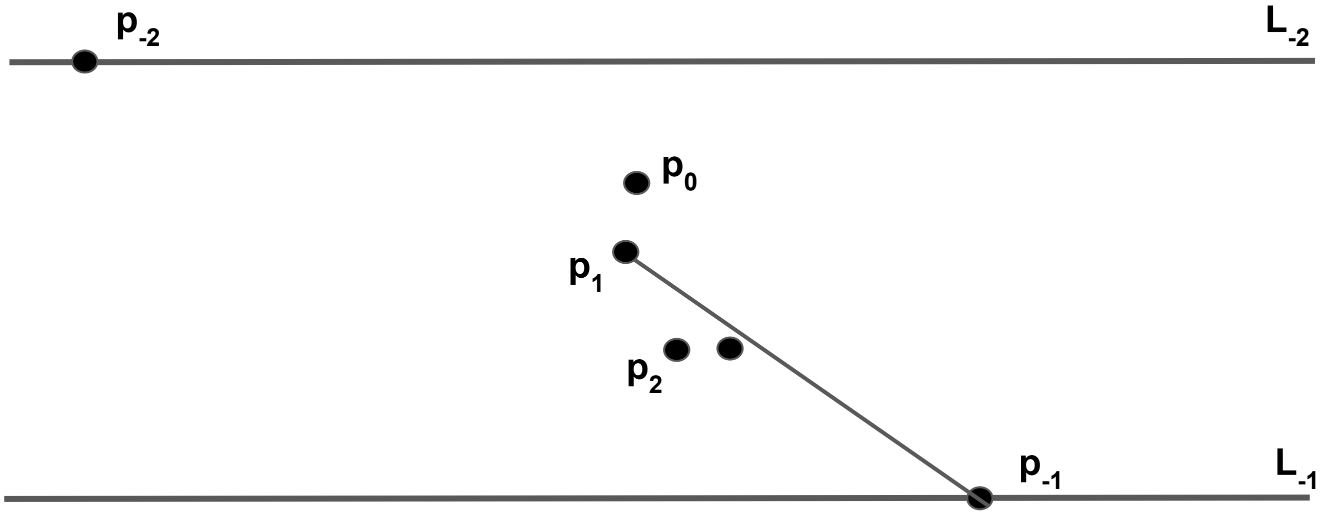
- Search tree



Dummy Nodes

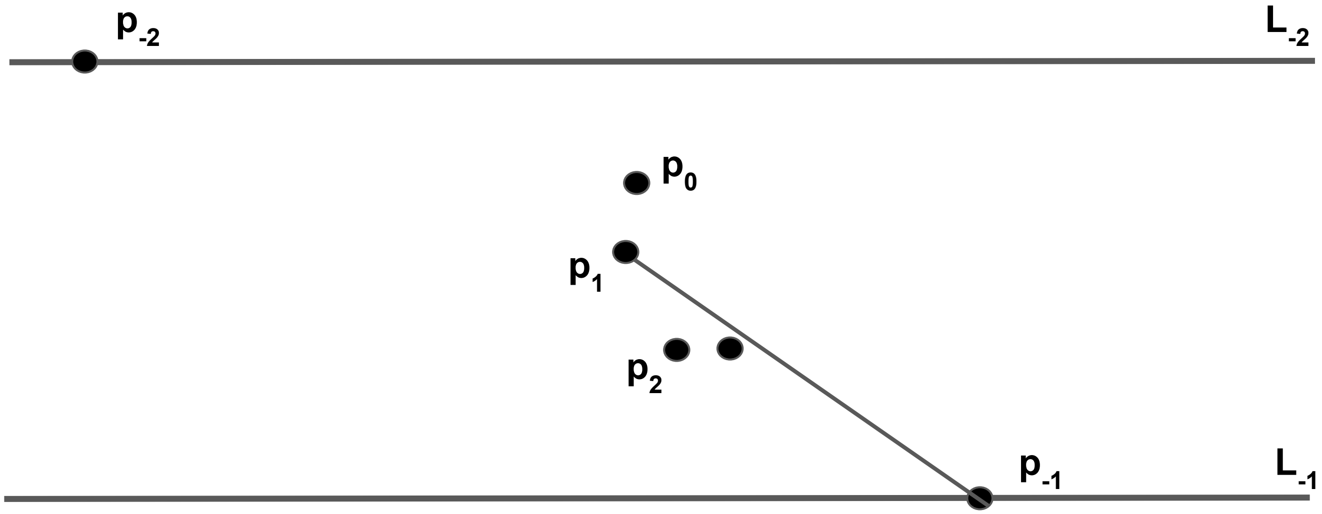


Dummy Nodes



p_i left of $p_j p_{-1} \Leftrightarrow i > j$

Dummy Nodes

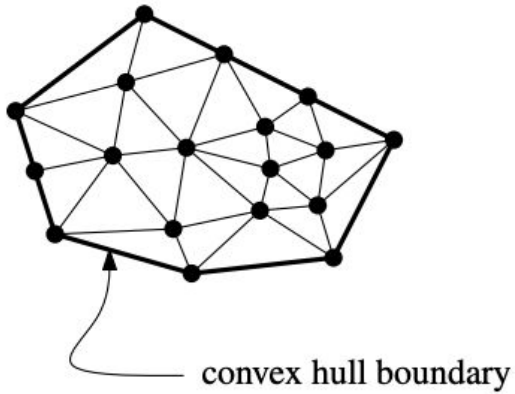


$p_i p_j$ illegal $\Leftrightarrow \min(k, l) < \min(i, j)$

Analysis

Theorem 9.12 *The Delaunay triangulation of a set P of n points in the plane can be computed in $O(n \log n)$ expected time, using $O(n)$ expected storage.*

Theorem 9.1 *Let P be a set of n points in the plane, not all collinear, and let k denote the number of points in P that lie on the boundary of the convex hull of P. Then any triangulation of P has $2n-2-k$ triangles and $3n-3-k$ edges.*



Analysis

Lemma 9.11 *The expected number of triangles created by algorithm* DELAUNAYTRIANGULATION *is at most* $9n+1$.

- Storage for search tree is O(n)

Analysis

- Every visited triangle in tree has been destroyed
- Can be charged to Delaunay Triangle

$$\sum_{\Delta} \mathrm{card}(K(\Delta))$$

Lemma 9.13 *If P is a point set in general position, then*

$$\sum_{\Delta} \operatorname{card}(K(\Delta)) = O(n \log n),$$

where the summation is over all Delaunay triangles Δ created by the algorithm.

$$\sum_{r=1}^{n} \Bigg(\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \operatorname{card}(K(\Delta)) \Bigg).$$

$$\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \operatorname{card}(K(\Delta)) = \sum_{q \in P \setminus P_r} k(P_r, q, p_r).$$

$$\mathrm{E}\big[k(P_r, q, p_r)\big] \leqslant \frac{3k(P_r, q)}{r}.$$

$$\mathrm{E}\big[\sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} \operatorname{card}(K(\Delta))\big] \leqslant \frac{3}{r} \sum_{q \in P \setminus P_r} k(P_r, q).$$

$$\mathrm{E}\Big[\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \operatorname{card}(K(\Delta))\Big] \leqslant \frac{3}{r}\sum_{q\in P\setminus P_r} k(P_r,q).$$

$$\mathrm{E}\big[k(P_r,p_{r+1})\big] = \frac{1}{n-r}\sum_{q\in P\setminus P_r} k(P_r,q).$$

$$\mathrm{E}\Big[\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \operatorname{card}(K(\Delta))\Big] \leqslant 3\left(\frac{n-r}{r}\right)\mathrm{E}\big[k(P_r,p_{r+1})\big].$$

$$\mathrm{E}\Big[\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \operatorname{card}(K(\Delta))\Big] \leqslant 3\left(\frac{n-r}{r}\right)\mathrm{E}\big[\operatorname{card}(\mathcal{T}_r\setminus\mathcal{T}_{r+1})\big].$$

$$\mathrm{E}\Big[\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \operatorname{card}(K(\Delta))\Big] \leqslant 3\left(\frac{n-r}{r}\right)\Big(\mathrm{E}\big[\operatorname{card}(\mathcal{T}_{r+1}\setminus\mathcal{T}_r)\big]-2\Big).$$

$$\mathrm{E}\Big[\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \mathrm{card}(K(\Delta))\Big] \leqslant 3\left(\frac{n-r}{r}\right)\Big(\mathrm{E}\big[\mathrm{card}(\mathcal{T}_{r+1}\setminus\mathcal{T}_r)\big]-2\Big).$$

$$\mathrm{E}\Big[\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \mathrm{card}(K(\Delta))\Big] \leqslant 12\left(\frac{n-r}{r}\right).$$

$$\sum_{r=1}^{n}\left(\sum_{\Delta\in\mathcal{T}_r\setminus\mathcal{T}_{r-1}} \mathrm{card}(K(\Delta))\right). \quad = O(n\log n)$$

Framework

Theorem 9.14 *Let (X,Π,D,K) be a configuration space, and let $\mathcal{T}$ and X_r be defined as above. Then the expected number of configurations in $\mathcal{T}(X_r) \setminus \mathcal{T}(X_{r-1})$ is at most*

$$\frac{d}{r} E[\text{card}(\mathcal{T}(X_r))],$$

where d is the maximum degree of the configuration space.

Framework

Theorem 9.15 *Let (X, Π, D, K) be a configuration space, and let $\mathcal{T}$ and X_r be defined as above. Then the expected value of*

$$\sum_{\Delta} \text{card}(K(\Delta)),$$

where the summation is over all configurations Δ appearing in at least one $\mathcal{T}(X_r)$ with $1 \leqslant r \leqslant n$, is at most

$$\sum_{r=1}^{n} d^2 \left(\frac{n-r}{r}\right)\left(\frac{\mathrm{E}\left[\text{card}(\mathcal{T}(X_r))\right]}{r}\right),$$

where d is the maximum degree of the configuration space.

History

- Descartes 1644
- Georgy Feodosievych Voronoy 1908
- Boris Delone 1934

Discussion

- Why maximize the smallest angle instead of minimize the largest angle?
- How necessary is it to compute the exact Delaunay triangulation?