Delaunay Triangulations

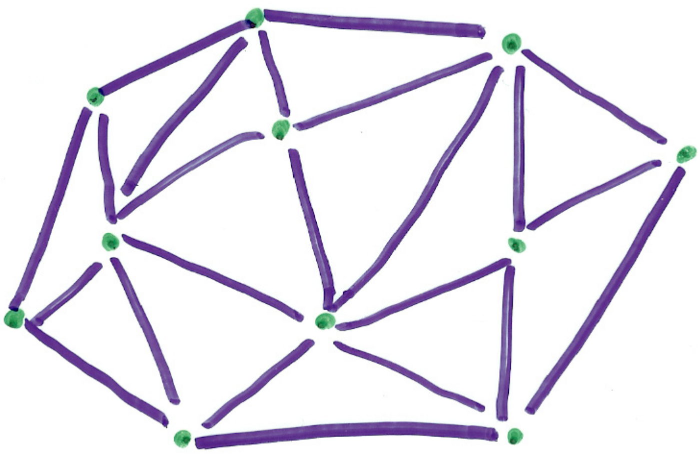
Let $P := \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$ be a finite set of points in the plane.

A <u>triangulation of P</u> is a $\underbrace{\text{simple}}_{}$ (no multiple edges), <u>plane</u> (planar embedded), connected graph $T = (P, E)$ such that

(i) every edge in E is a line segment

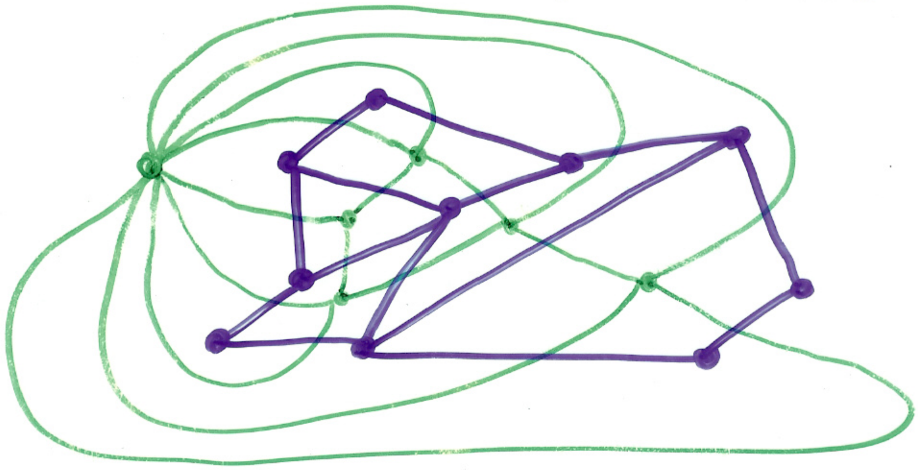
(ii) the outer face is bounded by edges of $CH(P)$

(iii) all inner faces are triangles

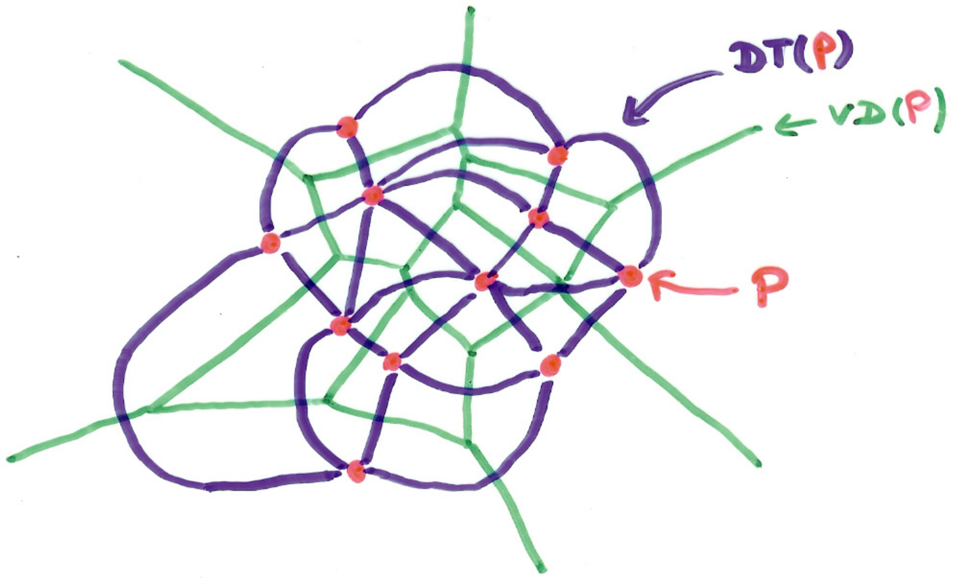


Let G be a plane graph. The <u>dual graph G^*</u> has

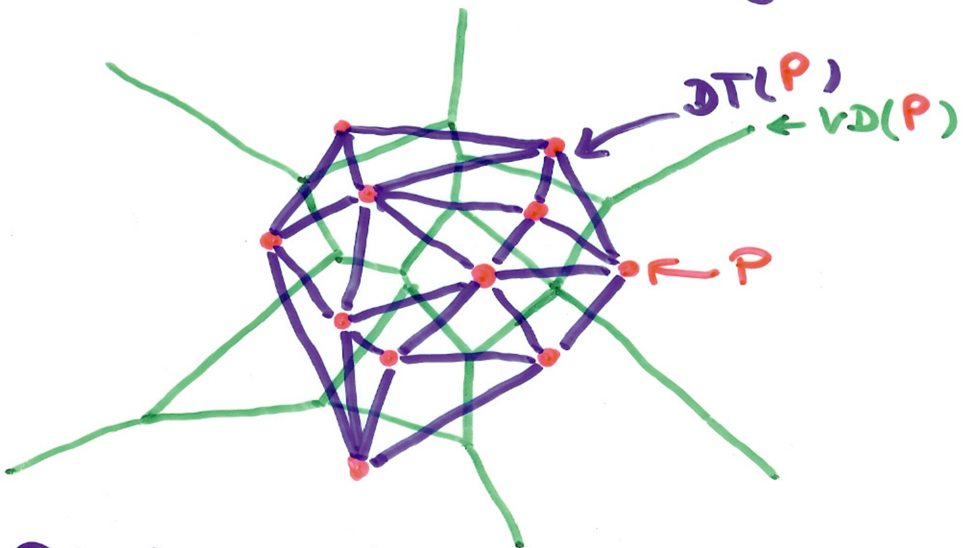
- a vertex for every face of G
- an edge for every edge of G
  (between the two faces incident to the original edge)



Let G be the plane graph for the Voronoi diagram $VD(P)$. Then the <u>dual graph</u> G^* is called the <u>Delaunay triangulation</u> $DT(P)$.



Canonical straight-line embedding for $DT(P)$:



- If P is in general position (no three points on a line, no four points on a circle) then every inner face of $DT(P)$ is indeed a triangle.
- $DT(P)$ can be stored as an abstract graph, without geometric information. (No such obvious storing scheme for $VD(P)$)

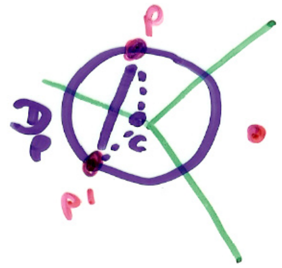
Can the straight line segments of DT(P) intersect? No.

Theorem: DT(P) is a plane graph.

Proof:

- $\overline{pp'}$ is an edge of DT(P)

  $\Leftrightarrow$ There is an empty closed disk D_p with p, p' on its boundary, and its center c on the bisector

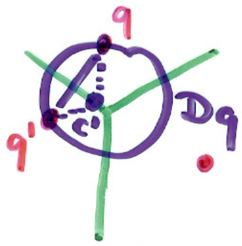


$\overline{pc}$ is fully contained in the Voronoi cell of p

- Let $\overline{q,q'}$ be another Delaunay edge that intersects $\overline{pp'}$
- q, q' lie outside of D_p

  $\rightsquigarrow$ $\overline{q,q'}$ also intersects $\overline{pc}$ or $\overline{p'c}$

- Similarly:



  $\rightsquigarrow$ $\overline{p,p'}$ also intersects $\overline{qc'}$ or $\overline{q'c'}$

- $\Rightarrow$ ($\overline{pc}$ or $\overline{p'c}$) and ($\overline{qc'}$ or $\overline{q'c'}$) intersect

  $\Rightarrow$ edges not in different Voronoi cells

  ↯ contradiction

Characterizations of the DT(P) I

Lemma: For $p, q, r \in P$ let $\Delta(p,q,r)$ be the triangle they define. Then the following statements are equi

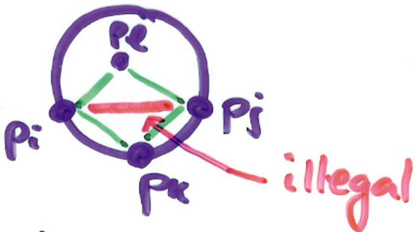
(i) $\Delta(p,q,r)$ belongs to $DT(P)$

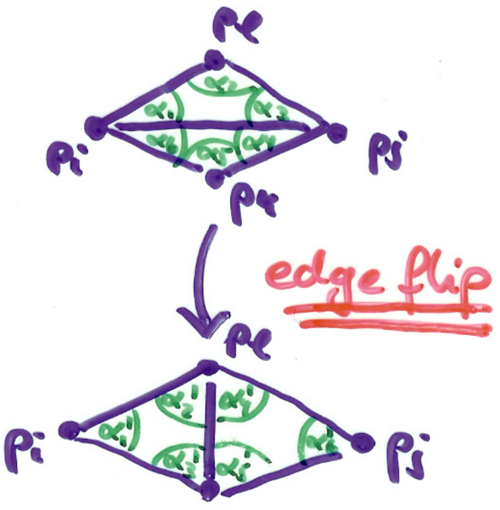
$\Leftrightarrow$ (ii) The circumcenter of $\Delta(p,q,r)$ is a vertex in $VD(P)$

$\Leftrightarrow$ (iii) The circumcircle of $\Delta(p,q,r)$ is empty (i.e., contains no other point of P)

Characterization I: Let T be a triangulation of P. Then $T = DT(P) \Leftrightarrow$ The circumcircle of any triangle in T is empty

Let $p_i, p_j, p_k, p_\ell \in P$. Then $\overline{p_i p_j}$ is a illegal edge $:\Leftrightarrow$ p_ℓ lies in the interior of the circle through p_i, p_j, p_k



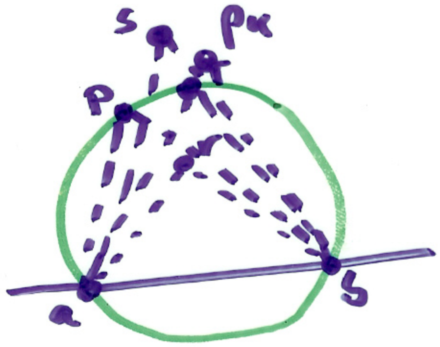


Equivalently $\overline{p_i p_j}$ is illegal

$\Leftrightarrow \min_{1 \le i \le 6} \alpha_i < \min_{1 \le i \le 6} \alpha_i'$

Theorem (Thales): Let a, b, p, q be four points on a circle, r inside and s outside the circle, and all p, q, r, s to the same side of the line through a, b. Then

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



Characterizations of the DT(P) II

A triangulation is called legal if it does not contain an illegal edge.

Characterization 2: Let T be a triangulation of P.
Then $T = DT(P) \Leftrightarrow T$ is legal

Algorithm Legal Triangulation (T):

Input: A triangulation T of a point set P
Output: A legal triangulation of P

while T contains an illegal edge $\overline{p_i p_j}$ **do**
  // Flip $\overline{p_i p_j}$
  Let $\Delta(p_i p_j p_k)$ and $\Delta(p_i, p_j, p_l)$ be the two adjacent triangles in T.
  Remove $\overline{p_i p_j}$ from T, add $\overline{p_k p_l}$ to T
return T

Analysis:

- In every iteration of the loop the angle vector of T (all angles in T sorted by increasing value) increases
- With this one can show that a flipped edge never appears again

$\rightsquigarrow O(n^2)$ edges $\rightsquigarrow O(n^2)$ time

Characterizations of the DT(P) III

- Let T be a triangulation of P, and let $\alpha_1, \alpha_2, \dots, \alpha_{3m}$ the angles of the m triangles in T, sorted by increasing value. Then $A(T) := (\alpha_1, \dots, \alpha_{3m})$ is called the angle vector of T.
- A triangulation T is called angle optimal iff $A(T) > A(T')$ for any other triangulation T' of the same point set
- Let T' be a triangulation ~~and~~ containing an illegal edge, and let T'' be the triangulation resulting from flipping this edge.

$$\Rightarrow A(T'') > A(T')$$

- T is angle optimal $\Rightarrow$ T is legal $\Rightarrow T = DT(P)$

Characterization 3: Let T be a triangulation of P. Then $T = DT(P) \Leftrightarrow T$ is angle optimal

(If P is not in general position then any triangulation obtained by triangulating the faces maximizes the minimum angle.)

Applications of the Delaunay Triangulation

1) Terrain modelling: Model a scanned surface (terrain: graph of continuous piecewise linear function) by triangulation of the sample points.

$\Rightarrow$ Angle optimal triangulations give intuitively better approximations since they avoid long skinny triangles

2) All nearest neighbors: Find for each $p \in P$ its nearest neighbor $q \in P$; $q \neq p$.

- Every $p \in P$ is adjacent to its nearest neighbor $q \in P$ in $DT(P)$

(Assume not, then 

containing another point r, which then is closer. ↯)

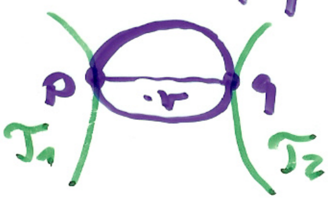
- $O(n)$ algorithm which simply checks all adjacent edges for each point in P

3) Minimal spanning tree: Construct a Euclidean MST for P.

- The edges of every MST of P are a subset of the edges of $DT(P)$

Proof: Assume $\overline{pq}$ is in an MST $\mathcal{T}$, but $\overline{pq} \notin DT(P)$

$\rightarrow$ Circle with diameter $|\overline{pq}|$



Euclidean distance

$\rightarrow d(p,r) < d(p,q)$ and $d(q,r) < d(p,q)$

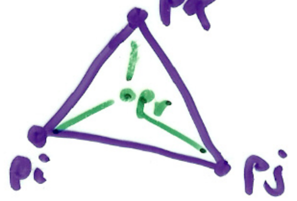
$\rightarrow$ Remove $\overline{pq}$ from $\mathcal{T}_1$
- If $r \in \mathcal{T}_1$, insert $\overline{rq}$
- If $r \in \mathcal{T}_2$, insert $\overline{pr}$

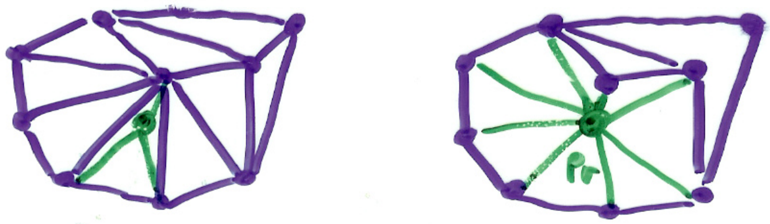
$\rightarrow$ New MST of smaller total length ↯

Randomized incremental Construction of DT(P)

- Start with a large triangle $\Delta(p_{-3}, p_{-2}, p_{-1})$ containing P

- Insert points of P incrementally
  - Find the containing triangle
  - Add new edges
  - Flip all illegal edges until every edge is legal

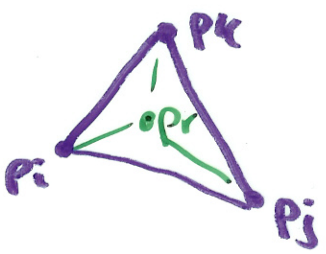


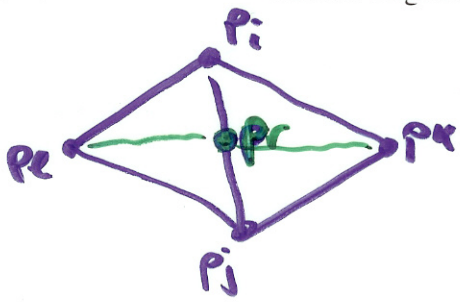


- – An edge can become illegal only if one of its incident triangles changes
  - Check only edges of new triangles
  - Every new edge created is incident to p_r
  - Every old edge is legal (if p_r is on one of the incident triangles, the edge would have been flipped if it were illegal)
  - Every new edge is legal (since it has been created from flipping an illegal edge; it can be seen that this yields a legal edge)

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

Let p_{-1}, p_{-2}, and p_{-3} be a suitable set of three points such that P is contained in the triangle $p_{-1}p_{-2}p_{-3}$.
Initialize $\mathcal{T}$ as the triangulation consisting of the single triangle $p_{-1}p_{-2}p_{-3}$.
Compute a random permutation $p_1, p_2, \ldots, p_n$ of P.
for $r \leftarrow 1$ **to** n
    do (* Insert p_r into $\mathcal{T}$: *)
        Find a triangle $p_ip_jp_k \in \mathcal{T}$ containing p_r.
        if p_r lies in the interior of the triangle $p_ip_jp_k$
          then Add edges from p_r to the three vertices of $p_ip_jp_k$, thereby splitting $p_ip_jp_k$ into three triangles.
                LEGALIZEEDGE($p_r, \overline{p_ip_j}, \mathcal{T}$)
                LEGALIZEEDGE($p_r, \overline{p_jp_k}, \mathcal{T}$)
                LEGALIZEEDGE($p_r, \overline{p_kp_i}, \mathcal{T}$)
          else (* p_r lies on an edge of $p_ip_jp_k$, say the edge $\overline{p_ip_j}$ *)
                Add edges from p_r to p_k and to the third vertex p_l of the other triangle that is incident to $\overline{p_ip_j}$, thereby splitting the two triangles incident to $\overline{p_ip_j}$ into four triangles.
                LEGALIZEEDGE($p_r, \overline{p_ip_l}, \mathcal{T}$)
                LEGALIZEEDGE($p_r, \overline{p_lp_j}, \mathcal{T}$)
                LEGALIZEEDGE($p_r, \overline{p_jp_k}, \mathcal{T}$)
                LEGALIZEEDGE($p_r, \overline{p_kp_i}, \mathcal{T}$)
Discard p_{-1}, p_{-2}, and p_{-3} with all their incident edges from $\mathcal{T}$.
return $\mathcal{T}$





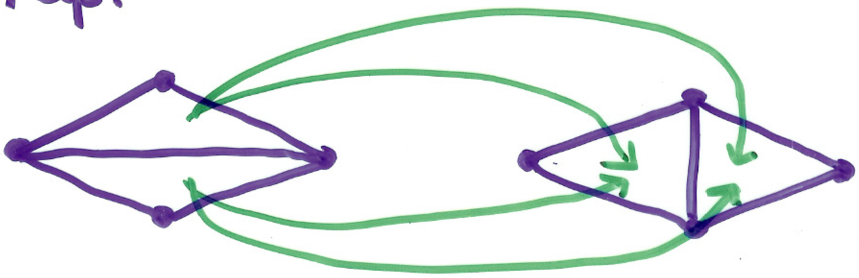
LEGALIZEEDGE($p_r, \overline{p_ip_j}, \mathcal{T}$)
(* The point being inserted is p_r, and $\overline{p_ip_j}$ is the edge of $\mathcal{T}$ that may need to be flipped. *)
if $\overline{p_ip_j}$ is illegal
    then Let $p_ip_jp_k$ be the triangle adjacent to $p_rp_ip_j$ along $\overline{p_ip_j}$.
        (* Flip $\overline{p_ip_j}$: *) Replace $\overline{p_ip_j}$ with $\overline{p_rp_k}$.
        LEGALIZEEDGE($p_r, \overline{p_ip_k}, \mathcal{T}$)
        LEGALIZEEDGE($p_r, \overline{p_kp_j}, \mathcal{T}$)

History of constructed triangles:

- Division of a triangle:

Store pointers from the old triangle to the three new triangles

- Flip:



Store pointers from both old triangles to both new triangles

- History allows easy localization of the triangle containing a new point by simply following the pointers

Analysis sketch (details in book):

- Without point location, the insertion of p_r in triangulation $T_r = DT(\{p_1, \dots, p_r\})$ takes $O(\deg(p_r, T_r))$ time
- Backwards analysis:
  - $E(\deg(p_r, T_r))$ is constant
    $\rightsquigarrow \sum_{r=1}^{n} E(\deg(p_r, T_r)) = O(n)$
  - $F_r :=$ # triangles visited in history to locate p_r
    $E(F_r) = O(\log r)$
    $\rightsquigarrow \sum_{r=1}^{n} E(F_r) = O(\sum_{r=1}^{n} \log r) = O(n \log n)$

Theorem: The randomized incremental construction of $DT(P)$, $|P| = n$, has an expected runtime of $O(n \log n)$.