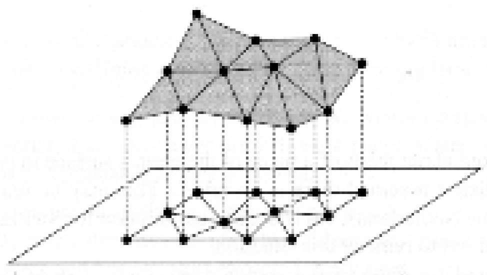
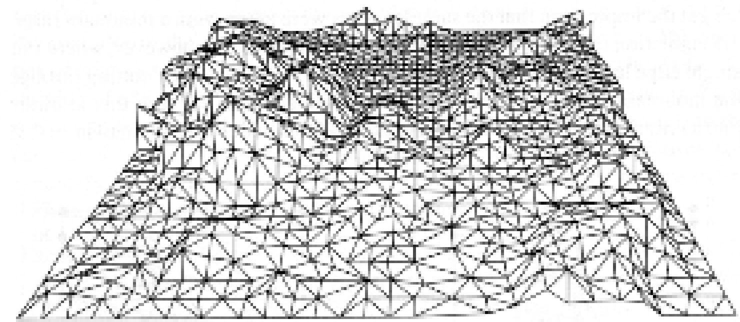
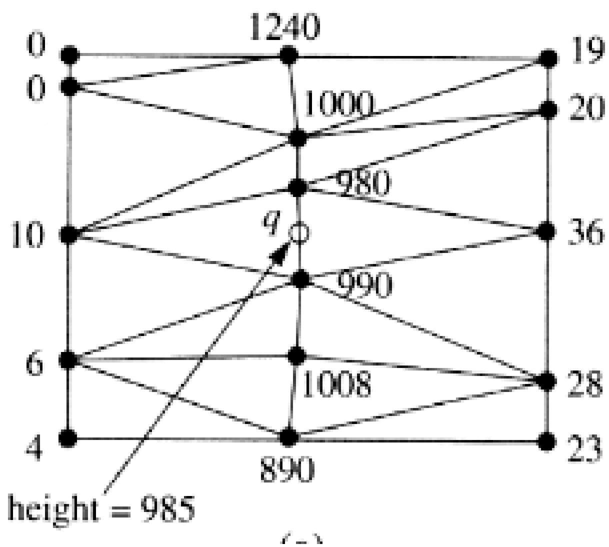
Delaunay Triangulations

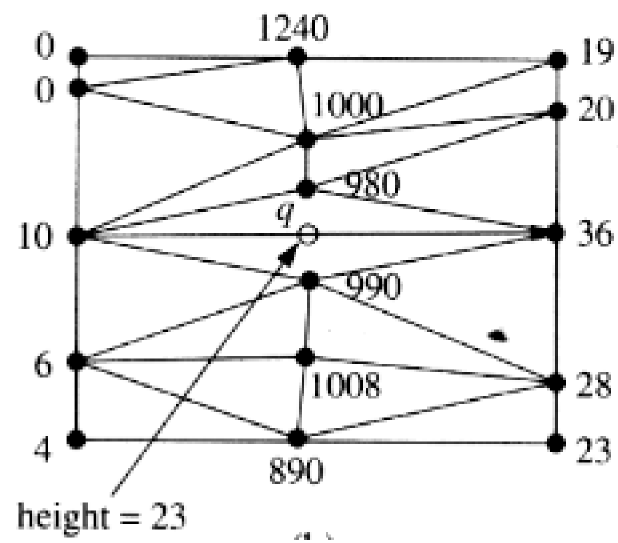
Pawel Winter

Motivation



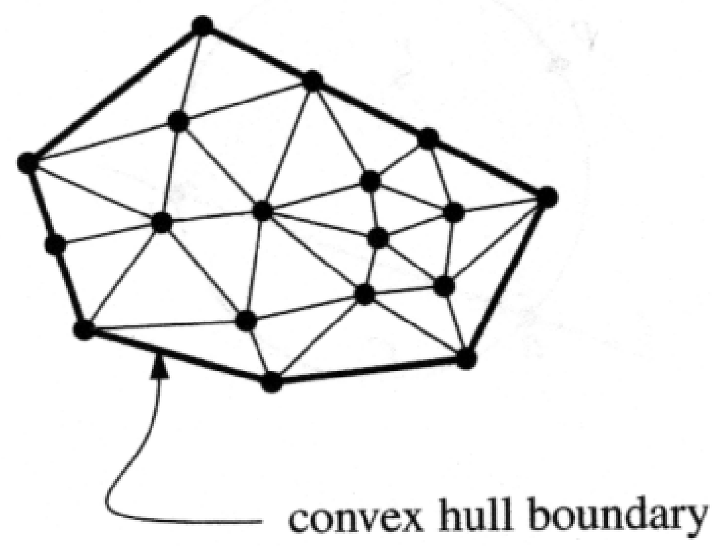



0
1240
19
0
1000
20
980
q
10
36
990
6
28
1008
4
23
890
height = 985


0
1240
19
0
1000
20
980
q
10
36
990
6
28
1008
4
23
890
height = 23

Triangulation is made of triangles

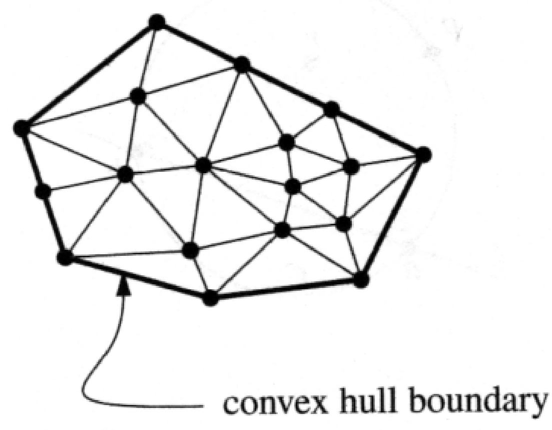
- Outer polygon must be convex hull
- Internal faces must be triangles, otherwise they could be triangulated further



Triangulation Details

For P consisting of n points, all triangulations contain 2n-2-k triangles, 3n-3-k edges

- n = number of points in P
- k = number of points on convex hull of P



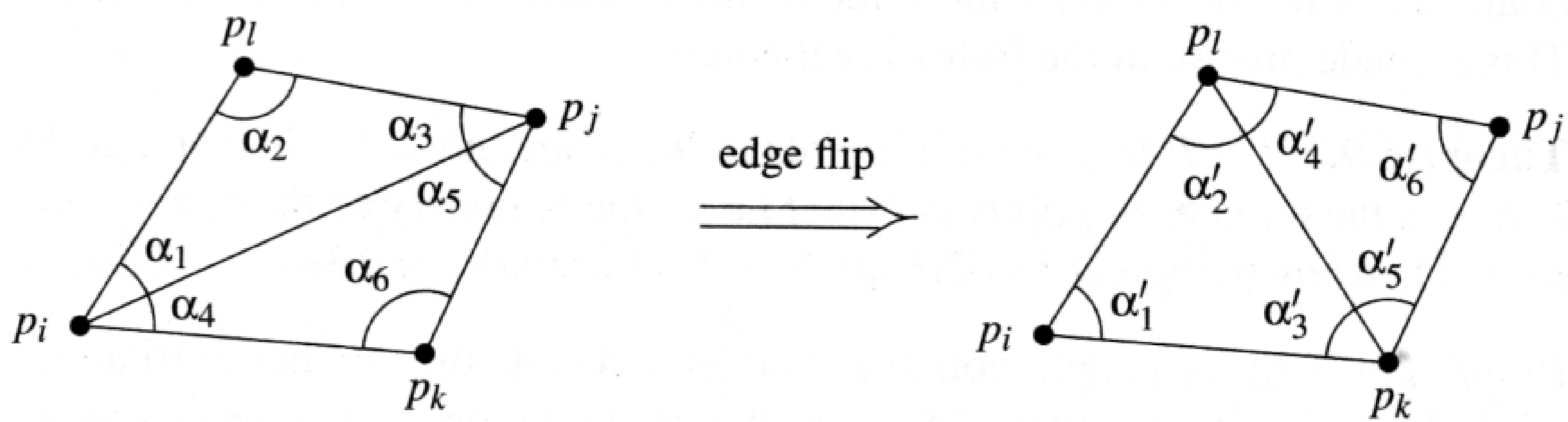
Angle Optimal Triangulations

- Create *angle vector* of the sorted angles of triangulation *T*, $(\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_{3m}) = A(T)$ with α_1 being the smallest angle
- $A(T)$ is larger than $A(T')$ iff there exists an i such that $\alpha_j = \alpha'_j$ for all $j < i$ and $\alpha_i > \alpha'_i$
- Best triangulation is a triangulation that is *angle optimal, i.e.*, has the largest angle vector. Maximizes minimum angle.

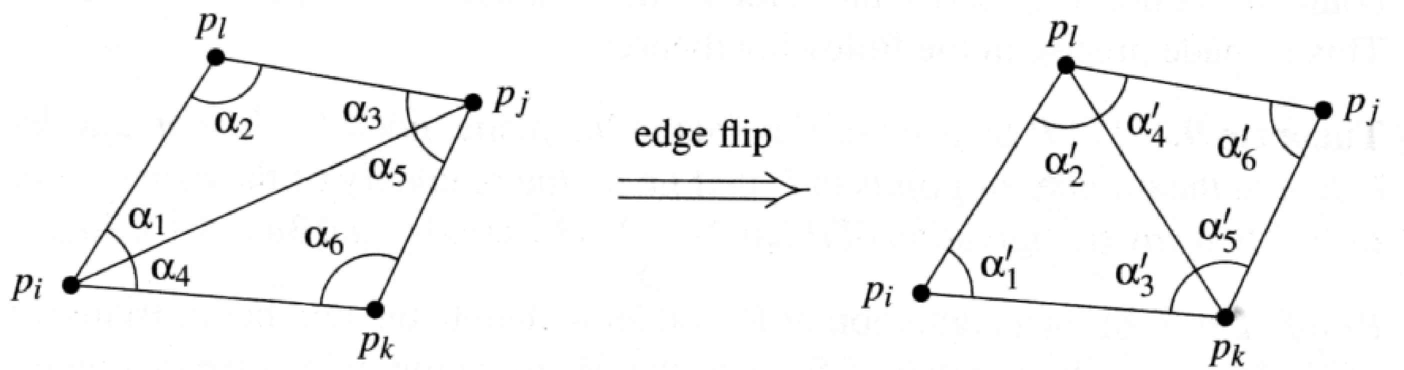
Angle Optimal Triangulations

Consider two adjacent triangles of T:

- If the two triangles form a convex quadrilateral, we could have an alternative triangulation by performing an *edge flip* on their shared edge.



Illegal Edges



- Edge *e* is illegal if:

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

- Only difference between *T* containing *e* and *T*' with *e* flipped are the six angles of the quadrilateral.

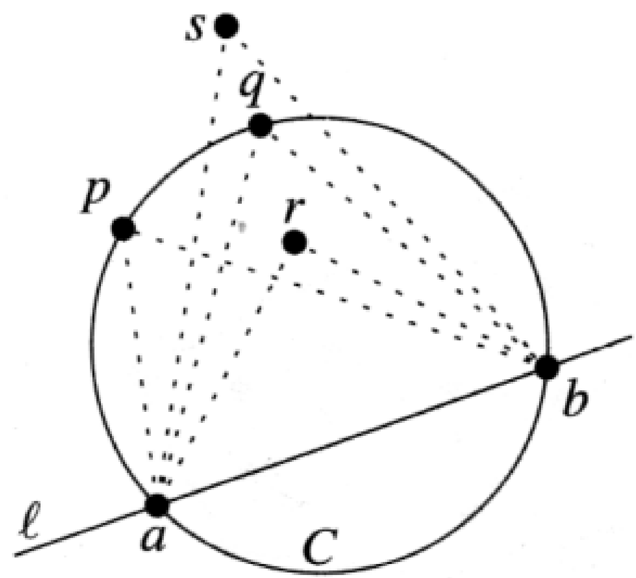
Illegal Triangulations

- If triangulation *T* contains an illegal edge e, we can make A(*T*) larger by flipping e.
- A triangulation *T* is an *illegal triangulation* if it contains an illegal edge.

Thales's Theorem

- We can use *Thales's Theorem* to test if an edge is legal without calculating angles

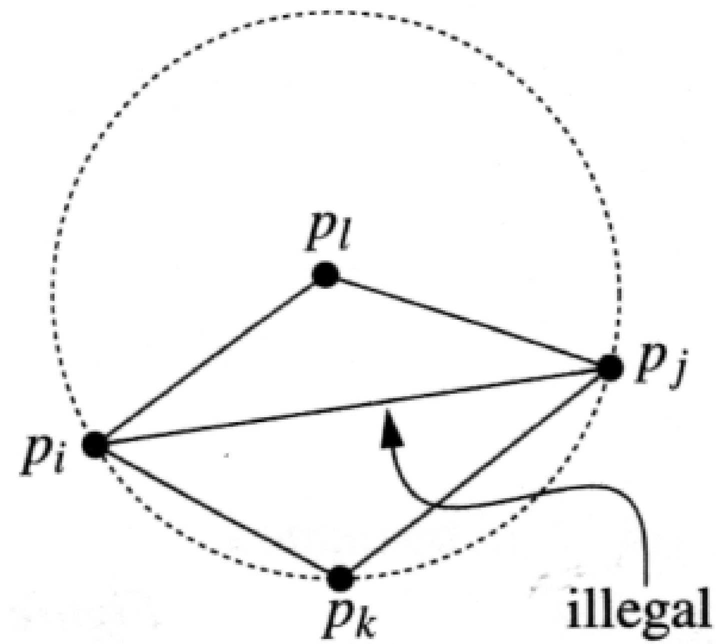
Let C be a circle, l a line intersecting C in points a and b and p, q, r, and s points lying on the same side of l. Suppose that p and q lie on C, that r lies inside C, and that s lies outside C. Then:



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

Testing for Illegal Edges

- If p_i, p_j, p_k, p_l form a convex quadrilateral and do not lie on a common circle, exactly one of p_ip_j and p_kp_l is an illegal edge.



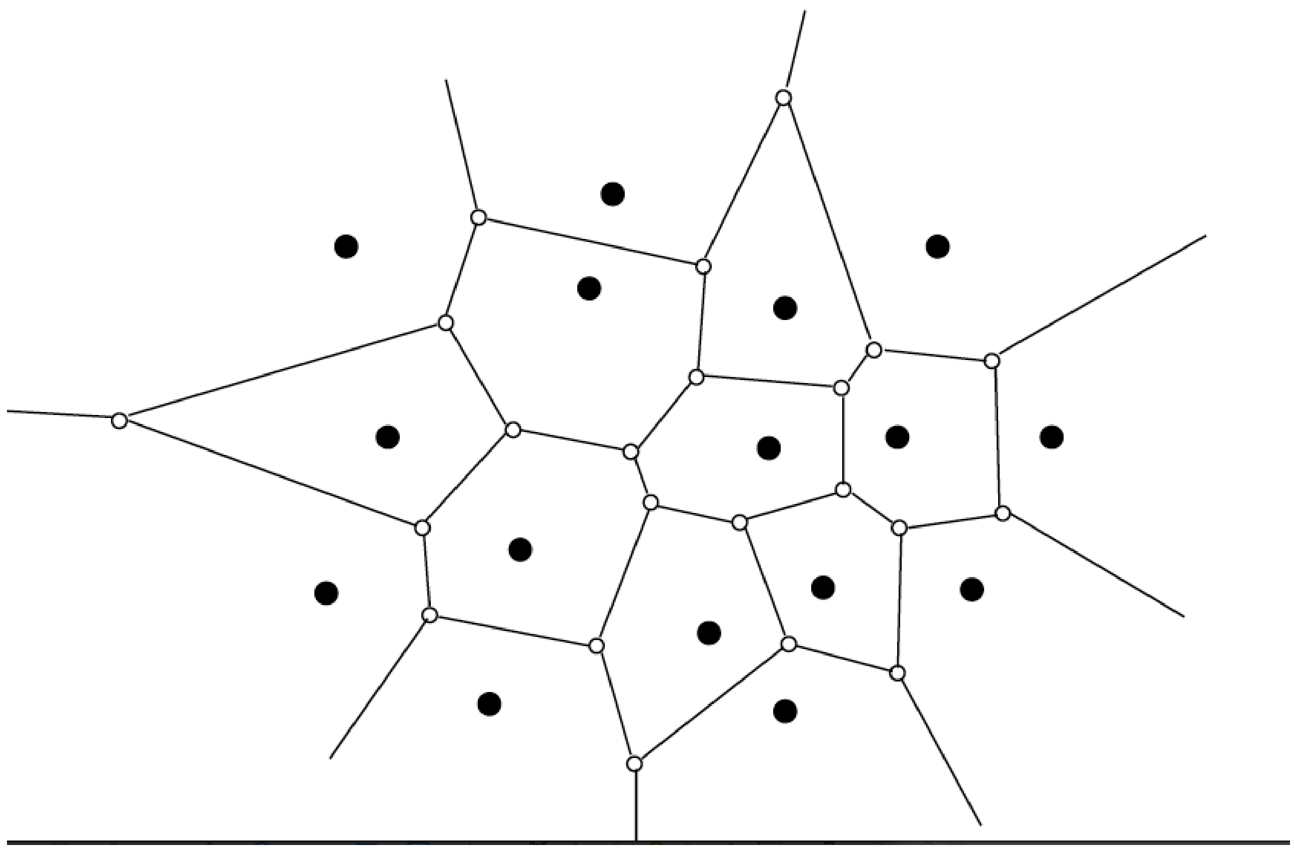
- The edge p_ip_j is illegal iff p_l lies inside circle C through p_i, p_j and p_k.

Computing Legal Triangulations

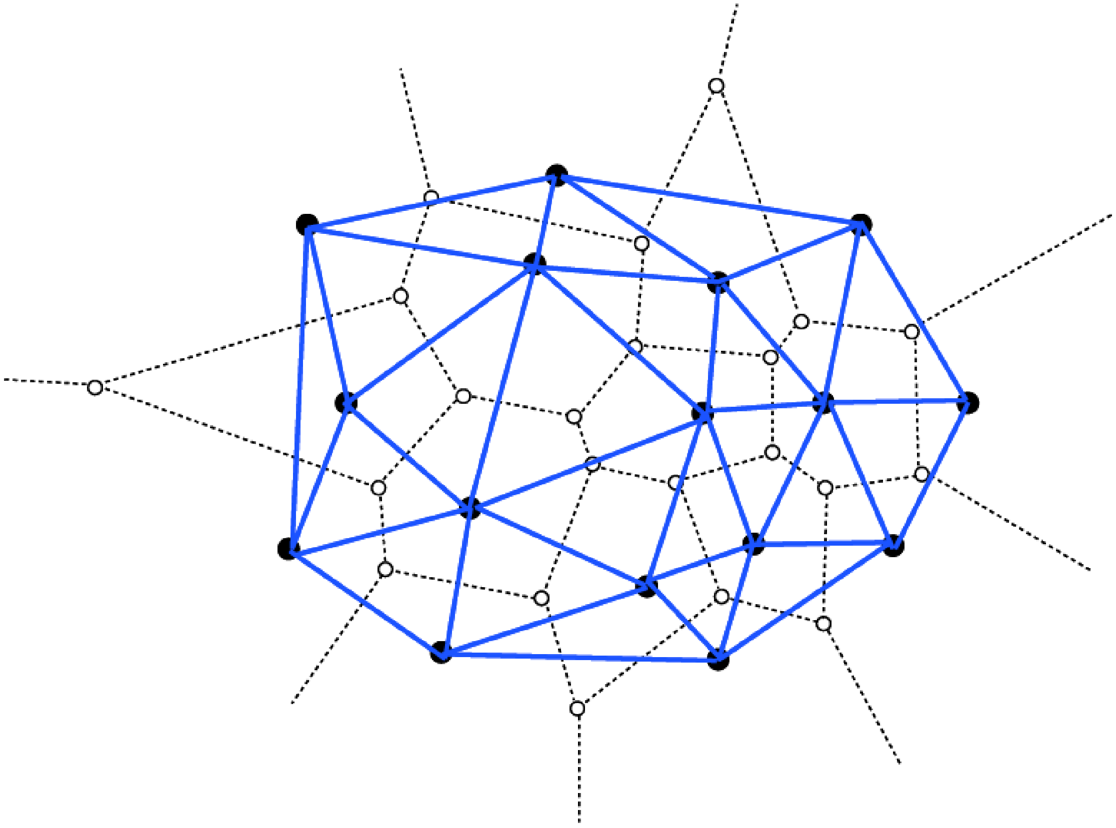
1. Compute a triangulation of input points P.
2. Flip illegal edges of this triangulation until all edges are legal.

- Algorithm terminates because there is a finite number of triangulations.
- *Too slow to be interesting…*

Voronoi Diagrams

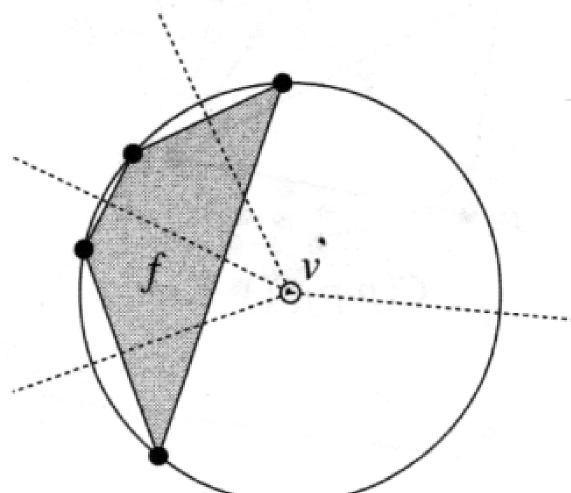


Delaunay Graph



Delaunay Triangulations

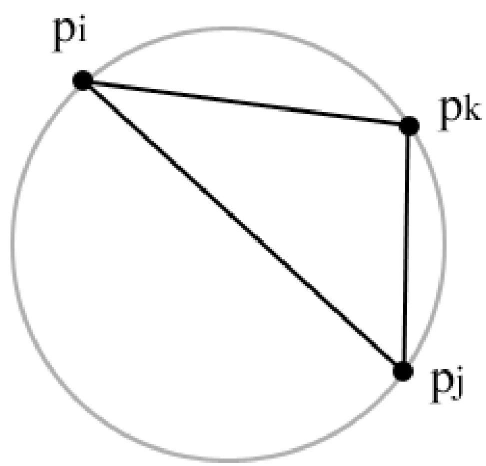
- Delaunay graph is the dual of Voronoi diagram
- Some sets of more than 3 points of Delaunay graph may lie on the same circle.
- These points form empty convex polygons, which can be triangulated.
- *Delaunay triangulation* is a triangulation obtained by adding 0 or more edges to the Delaunay graph.



Properties of Delaunay Triangles

From the properties of Voronoi Diagrams…

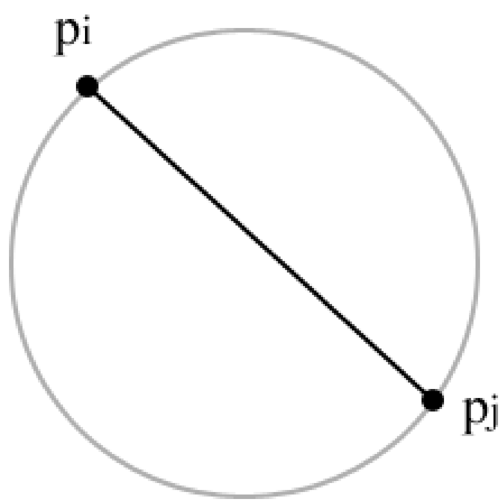
- Three points p_i, p_j, p_k ιν *P* are vertices of the same face of the DT(*P*) iff the circle through p_i, p_j, p_k contains no point of *P* on its interior.



Properties of Delaunay Triangles

From the properties of Voronoi Diagrams…

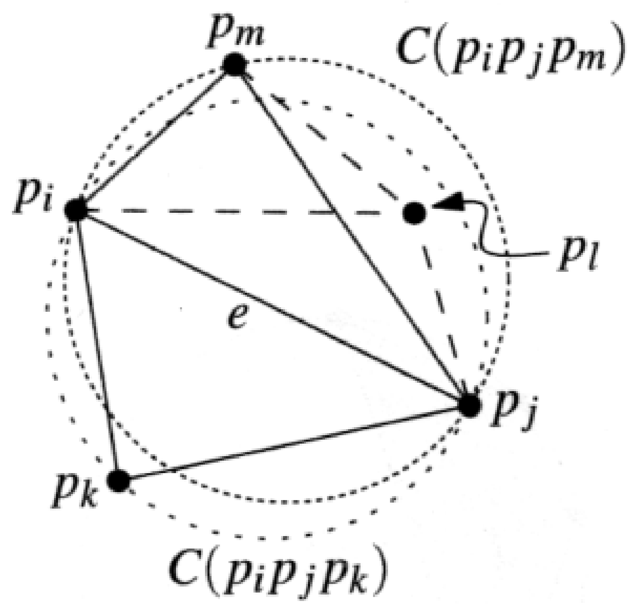
- Two points p_i, p_j ιν P form an edge of DT(P) iff there is a closed disc C that contains p_i and p_j on its boundary and does not contain any other point of P.



Legal Triangulations, revisited

A triangulation *T* of *P* is legal iff *T* is a DT(P).

- DT ==> Legal: Empty circle property and Thales's Theorem implies that all DT are legal
- Legal ==> DT: See Theorem 9.8



DT and Angle Optimal

- The angle optimal triangulation is legal and therefore it is a DT.
- If *P* is in general position, DT is unique. Hence there is only one legal triangulation and it is angle optimal.
- If multiple DT exist, not all of them are angle optimal.
- However, by Thales's Theorem, the minimum angle of each of the DT is the same.
- Thus, all DT maximize the minimum angle.

How do we compute DT(P)?

- We could compute Vor(P), and then dualize into DT(P).
- Instead, we will compute DT(P) using a randomized incremental method.

Algorithm Overview

1. Initialize triangulation T with a "big enough" helper bounding triangle that contains all points P.
2. Randomly choose a point p_r from P.
3. Find the triangle Δ that p_r lies in.
4. Subdivide Δ into smaller triangles that have p_r as a vertex.
5. Flip edges until all edges are legal.
6. Repeat steps 2-5 until all points have been added to T.

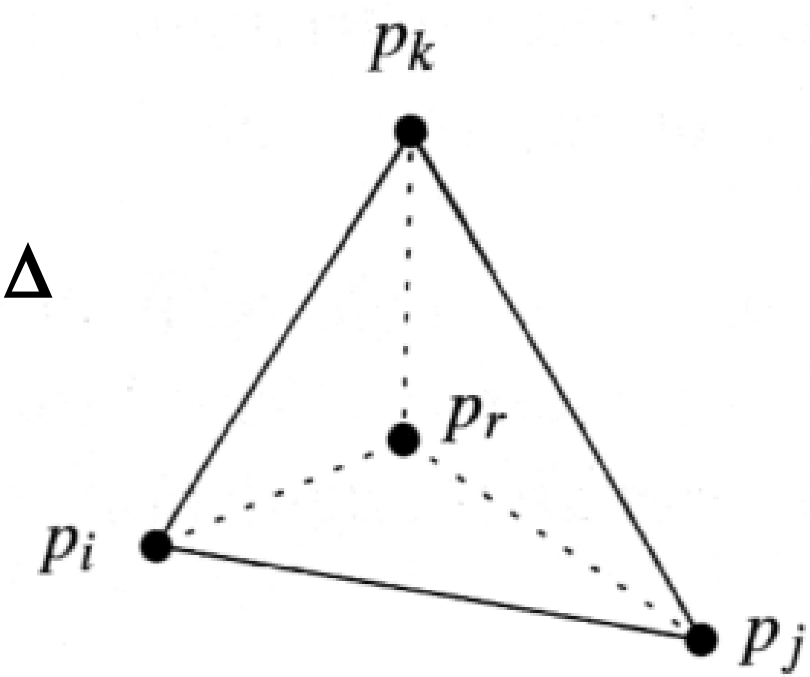
Let's skip steps 1, 2, and 3 for now…

Triangle Subdivision: Case 1 of 2

Assuming we have already found the triangle Δ that p_r lives in, subdivide Δ into smaller triangles that have p_r as a vertex.

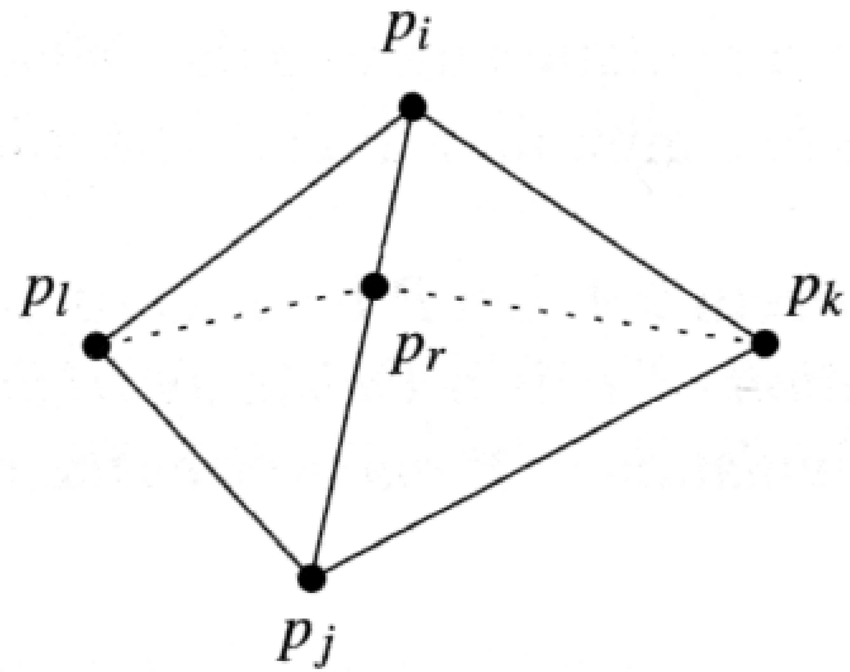
Two possible cases:

1) p_r lies in the interior of Δ



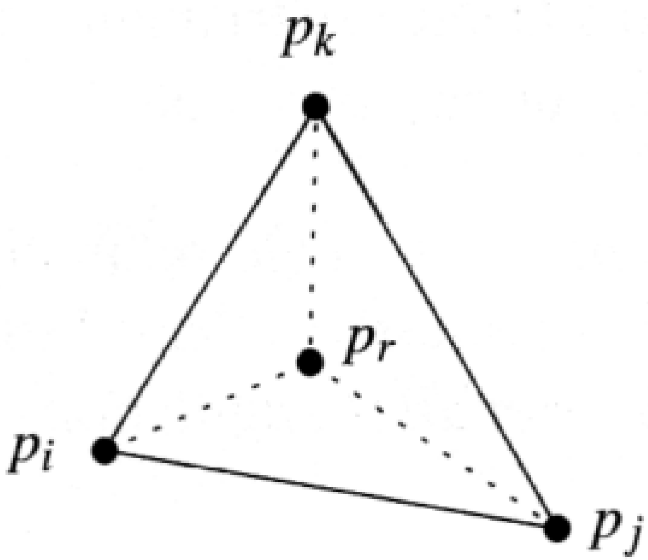
Triangle Subdivision: Case 2 of 2

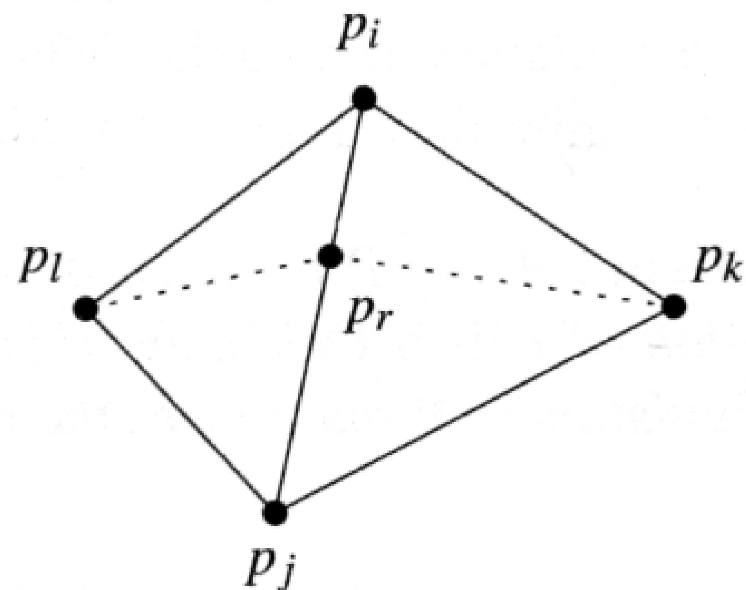
2) p_r falls on an edge between two adjacent triangles



Which edges are illegal?

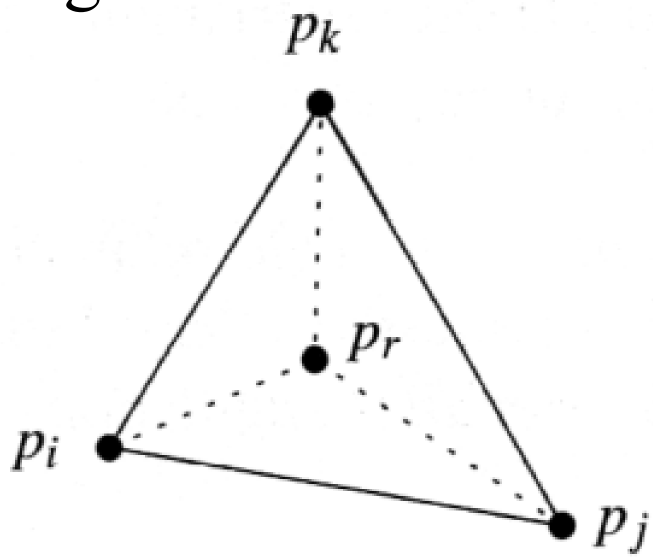
- Before subdivision, all edges are legal.
- After new edges is added, some of the edges may become illegal.
- Added edges are legal.

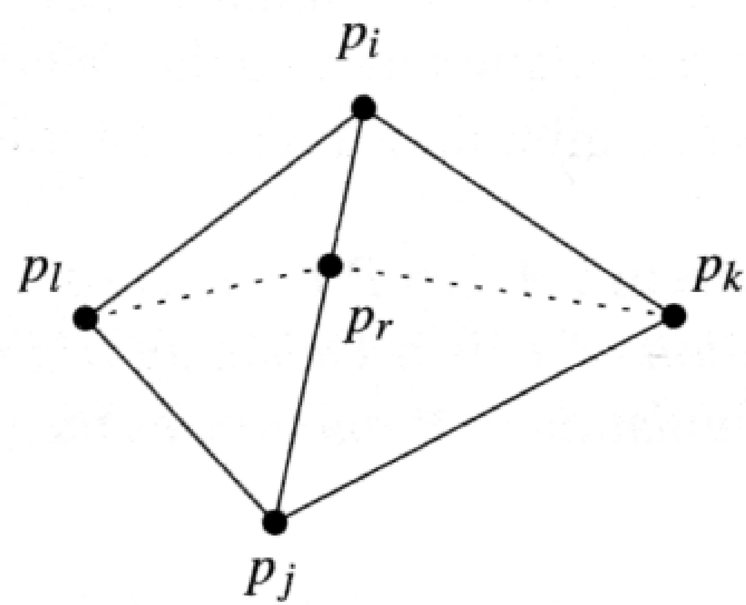




Outer Edges May Be Illegal

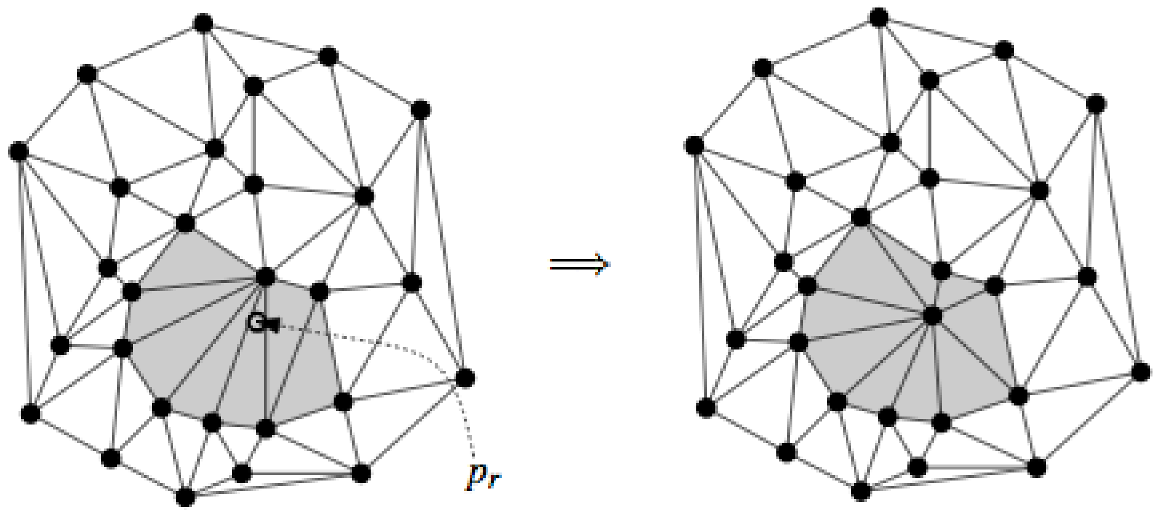
- An edge can become illegal only if one of its incident triangles changed.
- Outer edges of the incident triangles $\{p_jp_k, p_ip_k, p_kp_j\}$ or $\{p_ip_l, p_lp_j, p_jp_k, p_kp_i\}$ may have become illegal.





Flip Illegal Edges

- Flip the outer edges if they are illegal.
- Note that flipped edges are all incident to the added point p_r.

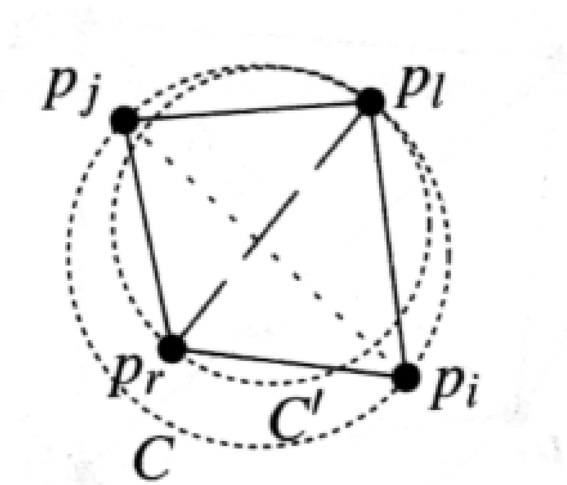


New Flipped Edges are Legal

Consider **a** new edge $p_r p_l$ caused by a flip.

Before adding $p_r p_l$,

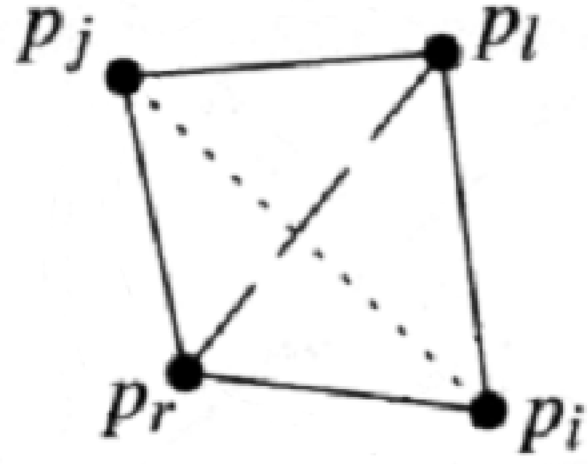
- p_l was part of some triangle $p_i p_j p_l$
- Circumcircle *C* of p_i, p_j, and p_l did not contain any other points of *P* in its interior.
- If we shrink *C*, we can find a circle *C'* that passes through $p_r p_l$
- C' contains no points in its interior.
- Therefore, $p_r p_l$ is legal.



Recursive Flipping

- However, after the edges have been flipped, outer edges of the new triangles may now be illegal.
- So we need to recursively flip edges…

LegalizeEdge



p_r = point being inserted

p_ip_j = edge that may need to be flipped

LEGALIZEEDGE(p_r, p_ip_j, T)

- **if** p_ip_j is illegal
- **then** Let $p_ip_jp_l$ be the triangle adjacent to $p_rp_ip_j$ along p_ip_j
- Replace p_ip_j with p_rp_l
- LEGALIZEEDGE(p_r, p_ip_l, T)
- LEGALIZEEDGE(p_r, p_lp_j, T)

Bounding Triangle

Let $\{p_{-2}, p_{-1}, p_0\}$ be the vertices of the bounding triangle:

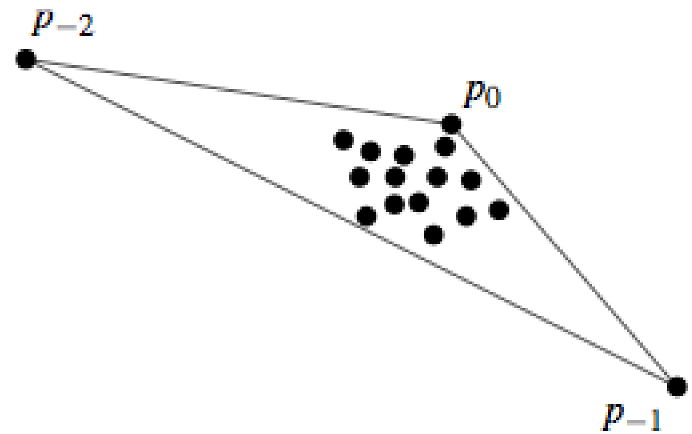
p_0 is the top point of given *n* points.

p_{-1} is below given *n* points sufficiently far to the right.

p_{-2} is above given *n* points sufficiently far to the left.

This triangle

- contains all points of P in its interior.
- will not destroy edges between points in P: consecutive vertices of the convex hull together with a big point have an empty circumscribing circle.

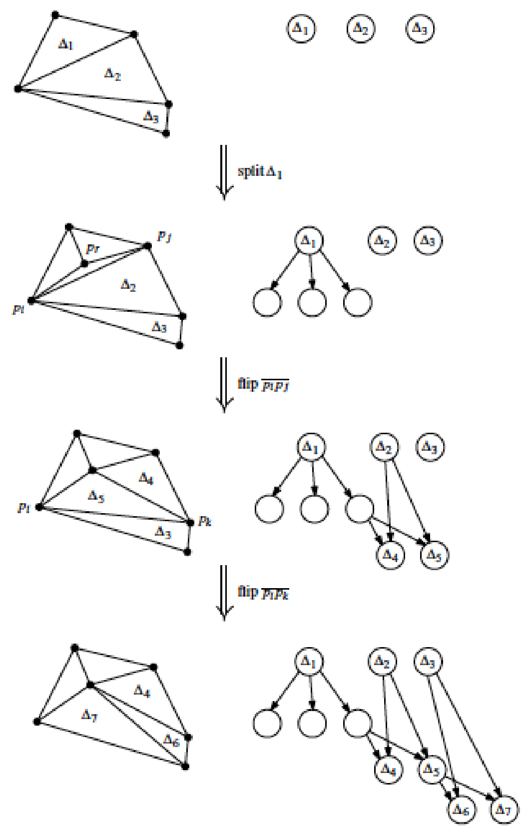


Triangle Location Step

3. Find the triangle T that p_r lies in.

- Take an approach similar to Point Location approach.
- Maintain a point location structure D, a directed acyclic graph.

Triangle Location



Time and Space Complexity

O(nlogn) expected time.

O(n) expected space.

See Section 9.4