Delaunay Triangulation/Graphs

Motivation: Terrains

Assume we have a 2-D manifold in the 3-D space, we want to represent it with only an efficient set of sample points of it.

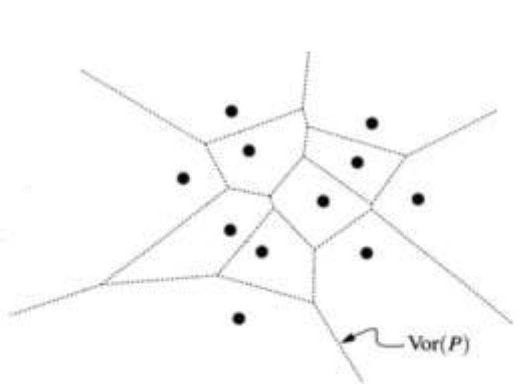
Let $A \subset \mathbb{R}^2$ be the set of data points, $f(p)$ is a height function for each $p \in A$.

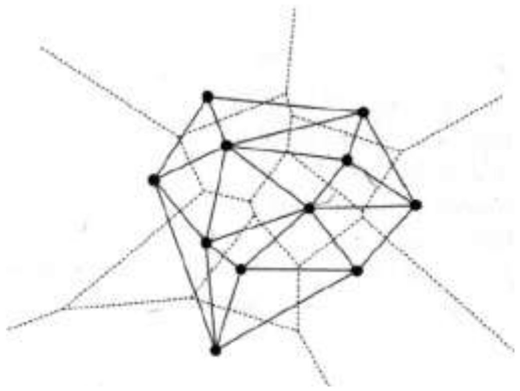
Using a simple step function, dividing the heights by the Voroni diagram, does not look natural.

A better solution: **triangulation** – and the Delaunay triangulation (the dual of the Voroni diagram) is the best approximation. The outer polygon must be a convex hull.

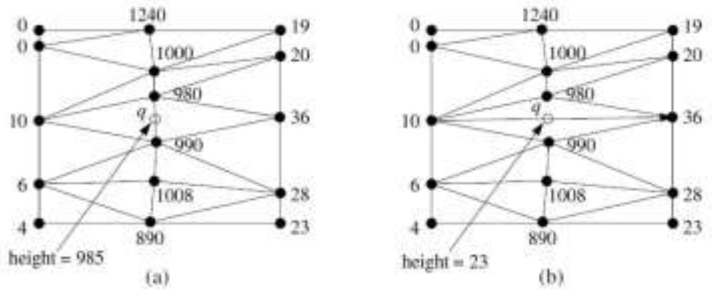
The **dual** of the Voroni diagram is connecting each edge-adjacent faces with an edge.

General position assumption: we assumed before that each Voroni vertex has a degree 3, therefore in the Delaunay triangulation each face will be a triangle.





Terrain problem, revisited



Any quadrilateral can be triangulated in 2 ways, and we want the smallest angle to be maximal, i.e. we don't want our triangles "skinny".

Delaunay triangulation

- Circumcircle property: for any triangle, drawing a circle through its vertices will have no interior points.
- Empty circle property: p, q are connected by an edge in the DT iff there's an empty circle passing through p, q.

  One direction is trivial, expanding an empty circle to include a third point and get the first property.

  If there's an empty circumcircle passing through p, q, then its center is a point on the edge of the VD between p, q.
- Closest pair property: closest pair of sites in P are neighbors in the DT.
- Let P be a set of n points, h of them in the convex hull, then:

- The DT has $2n-2-h$ triangles, $3n-3-h$ edges.

- In 3-D the DT will have $O(n) - O(n^2)$ tetrahedral.

Minimum spanning trees

<u>Theorem</u> (homework): the minimum spanning tree of a set of points P (in any dimension) is a subgraph of the DT – this is good since MST algorithms take $O\big((n+m)\lg n\big)$, and in a dense graph with $m = n^2$, we can first compute the DT in $O(n \lg n)$ and on that graph, which has $m = O(n)$, compute the MST.

Main property

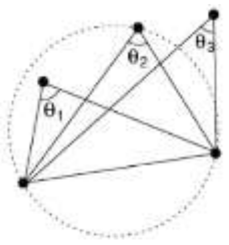
DT maximizes the minimum angle $\Rightarrow$ avoids skinny triangles.

For a triangulation T denote $a = \langle \alpha_1, \dots, \alpha_m \rangle$ the vector of angles in the triangulation where $\alpha_1 \le \dots \le \alpha_m$.

We say that $T_1 \ll T_2$ if at some i: $\alpha_i^1 < \alpha_i^2$ (and for all $j < i$: $\alpha_i^1 = \alpha_i^2$) – lexicographical comparison.

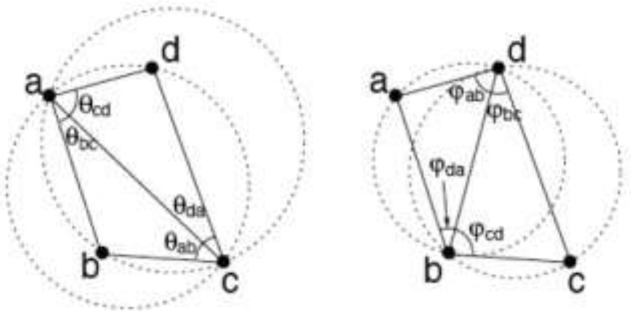
DT is a triangulation in which α_1 is maximal.

<u>Note</u>:



- $\theta_1 > \theta_2$, where θ_1 is an interior point and θ_2 is on the circle.
- $\theta_2 > \theta_3$, where θ_3 is an exterior point.
- The angle θ_2 is $\frac{1}{2}$ of the angle that sits on the same segment but sources at the center of the circle.

<u>Proof of main property</u>:



If bd violates the Delaunay condition, we flip it with ac – which changes the triangulation. This fixes the violation locally, i.e. the empty circle property. The angles properties above helps showing that $\min \varphi_i > \min \theta_i$.

That the algorithm for building DT: start with some triangulation, and then fix it repeatedly until no triangulation can be flipped.

Computing Legal Triangulation

- Compute a triangulation of input point P.
- Flip illegal edges until all are legal.
- <u>Problem</u>: repeating this process is exhaustive and slow.

But there is a simple randomized implementation with $O(n \lg n)$ expected time.

In-circle test:

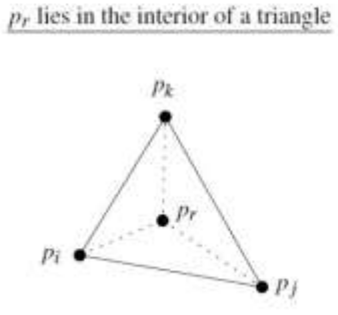
Testing whether a site d lies in the interior of the circle of the triangle Δabc: calculate the determinant of:

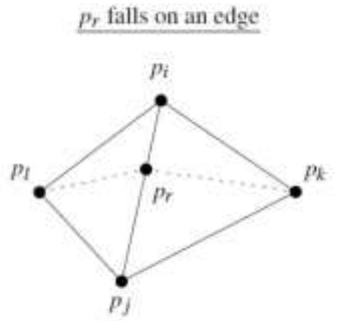
$\begin{bmatrix} a_x & a_y & a_x^2+a_y^2 & 1 \\ b_x & b_y & b_x^2+b_y^2 & 1 \\ c_x & c_y & c_x^2+c_y^2 & 1 \\ d_x & d_y & d_x^2+d_y^2 & 1 \end{bmatrix}$ – this is constant of course.

Algorithm overview

1. Start with a "big enough" bounding triangle that contains all points P (by adding 2 points "far enough" and the topmost $p \in P$).
2. Randomly choose $p_r \in P$.
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 until all points are added to T.

Cases of triangle subdivision:





Before adding p_r, $p_ip_jp_k$ was a valid triangle, since its circle had no other points in it.

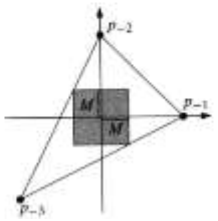
In the first case, p_r is in the circle of $p_ip_kp_j$, that circle had no $p \in P$ in it before. If we shrink the circle to have p_r, p_k on its circumference, we get that its circle is contained in the original larger circle, so it has no $p \in P$ in it too – so it is legal. We do have to check the outer sides of the edges, such as p_ip_k.

So in the first case we have 3 edges to check; in the second case we have to check the outer edges as well.

The check continues from that point on, and we need to bound the number of fixes we need to perform.

The moment we get to an edge that the condition is not violated, we stop propagating the check beyond it – since the condition holds for that part.

Bounding triangle:

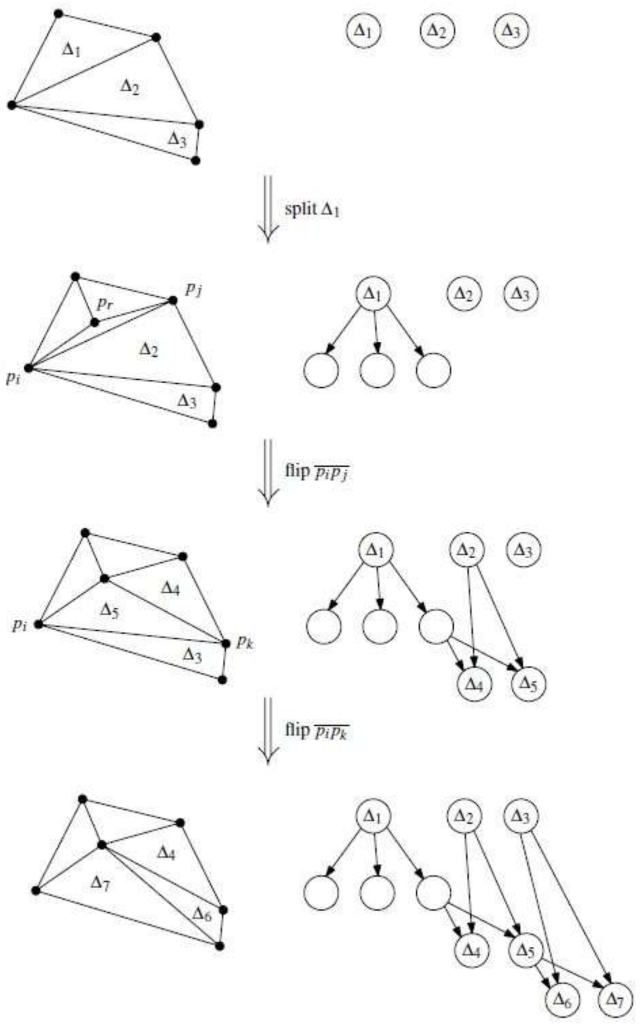


we set $p_{-1} = (3M, 0), p_{-2} = (0, 3M), p_{-3} = (-3M, -3M)$

Triangle location step

We maintain a data structure D similar to trapezoidal map to find efficiently the triangle in which p_r resides.

The leaves of D are the triangles of the current triangulation. We start with the single leaf of $\Delta p_{-1}p_{-2}p_{-3}$.



<u>How do we search the data structure</u>?

The structure basically holds a history of triangulations. For a given p_r:

- Start with the root.
- Look for the child that contains p_r.
- Repeat until reached a leaf.

Running time

Notations:

- $P_r = \{p_1, \dots, p_r\}$ – the first r points processed.
- $\Omega = \{p_{-1}, p_{-2}, p_{-3}\}$ – the points of the bounding triangle.
- $\mathcal{DG}_r = \mathcal{DG}(\Omega \cup P_r)$ – the Delaunay graph of iteration r.

Lemma 9.11: the expected number of triangles created by the Delaunay-Triangulation procedure is $9n + 1$ – that is the total number of fixes, and the size of D.

Backward analysis

- Let p_r be a random element of P_r.
- $\mathcal{DG}_r$ has $3(r + 3) - 6$ edges – from Euler's formula – 3 vertices per triangle, minus 6 of the boundary.
- Total degree of $P_r \leq 2[3(r + 3) - 9] = 6r$

$\Rightarrow E[degree\ of\ random\ vertex] = 6$

This results with the expected number of triangles created in step r: complete analysis in the slides.