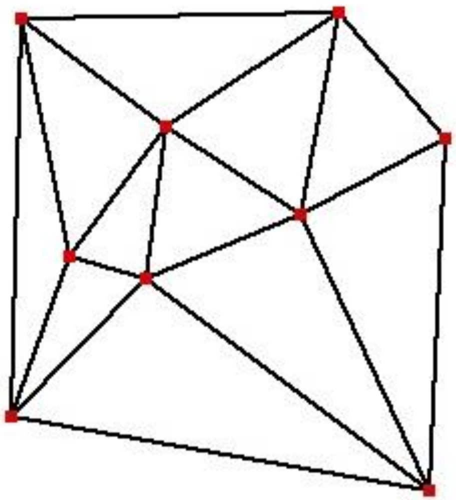
Computational Geometry



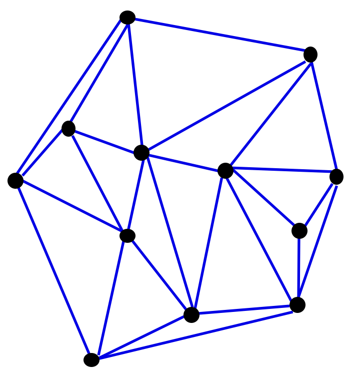
Delaunay Triangulations

Michael Goodrich

with slides from Carola Wenk and David Eppstein

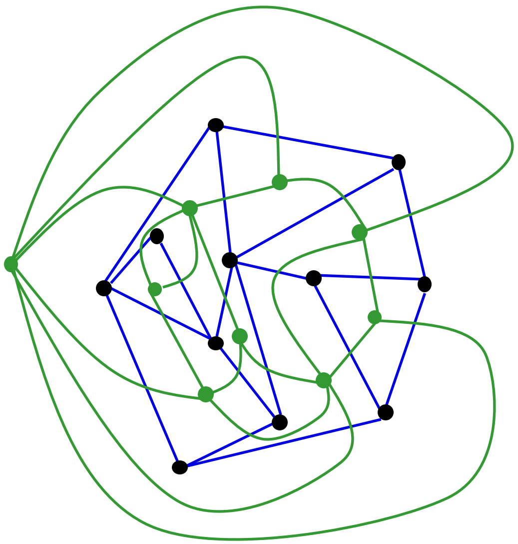
Triangulation

- Let $P = \{p_1, \ldots, p_n\} \subseteq R^2$ be a finite set of points in the plane.
- A **triangulation of *P*** is a simple, plane (i.e., planar embedded), connected graph $T=(P,E)$ such that
  - every edge in E is a line segment,
  - the outer face is bounded by edges of $\mathrm{CH}(P)$,
  - all inner faces are triangles.



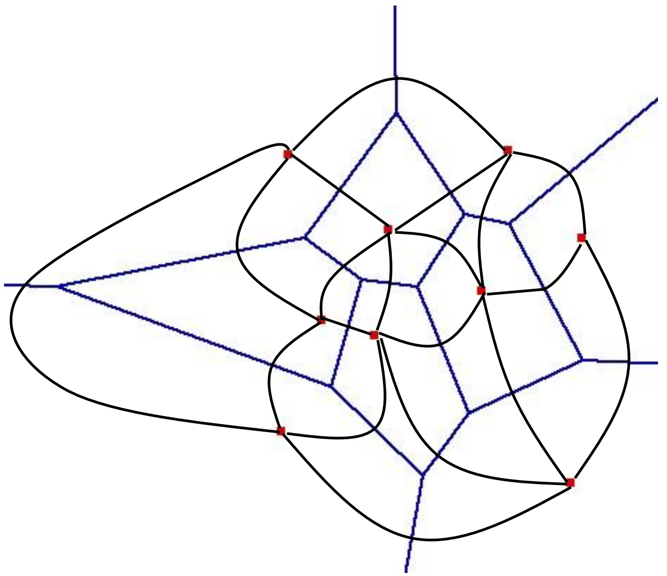
Dual Graph

- Let $G = (V, E)$ be a plane graph. The dual graph G^* has
  - a vertex for every face of G,
  - an edge for every edge of G, between the two faces incident to the original edge

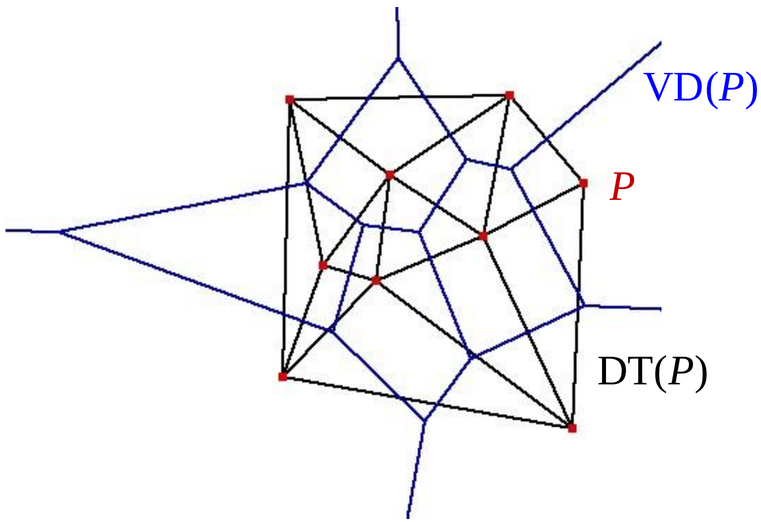


Delaunay Triangulation

- Let G be the plane graph for the Voronoi diagram VD(P) . Then the dual graph G^* is called the **Delaunay Triangulation DT(P)**.



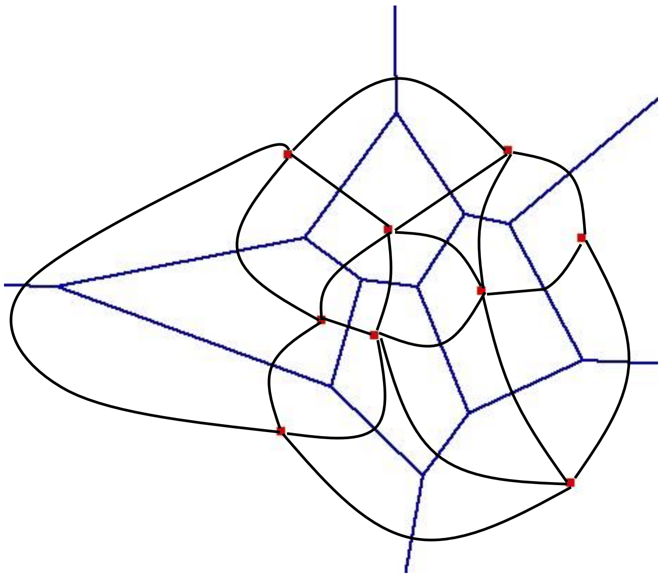
Canonical straight-line embedding for DT(P):



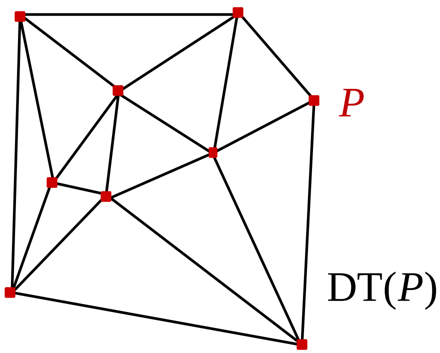
- If P is in general position (no three points on a line, no four points on a circle) then every inner face of DT(P) is indeed a triangle.
- DT(P) can be stored as an abstract graph, without geometric information. (No such obvious storing scheme for VD(P).)

Delaunay Triangulation

- Let *G* be the plane graph for the Voronoi diagram VD(*P*) . Then the dual graph *G** is called the **Delaunay Triangulation DT(*P*)**.



Canonical straight-line embedding for DT(P):

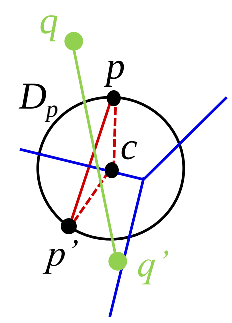


- If *P* is in general position (no three points on a line, no four points on a circle) then every inner face of DT(*P*) is indeed a triangle.
- DT(*P*) can be stored as an abstract graph, without geometric information. (No such obvious storing scheme for VD(*P*).)

Straight-Line Embedding

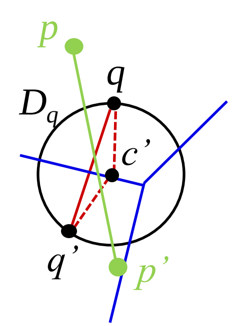
- **Lemma:** DT(P) is a plane graph, i.e., the straight-line edges do not intersect.
- **Proof:**
  - $\overline{pp'}$ is an edge of DT(P) $\Leftrightarrow$ There is an empty closed disk D_p with p and p' on its boundary, and its center c on the bisector.
  - Let $\overline{qq'}$ be another Delaunay edge that intersects $\overline{pp'}$

    $\Rightarrow$ q and q' lie outside of D_p, therefore $\overline{qq'}$ also intersects $\overline{pc}$ or $\overline{p'c}$
  - Symmetrically, $\overline{pp'}$ also intersects $\overline{qc'}$ or $\overline{q'c'}$



$\overline{pc} \subset V(p)$

$\overline{p'c} \subset V(p')$



$\overline{qc'} \subset V(q)$

$\overline{q'c'} \subset V(q')$

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

$\Rightarrow$ The edges do not lie in different Voronoi cells.

$\Rightarrow$ Contradiction $\square$

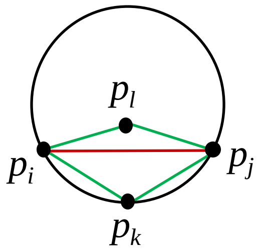
Characterization of DT(P)

- **Lemma**: Let $p,q,r \in P$ let Δ be the triangle they define. Then the following statements are equivalent:
  a) Δ belongs to DT(P)
  b) The circumcenter c of Δ is a vertex in VD(P)
  c) The circumcircle of Δ is empty (i.e., contains no other point of P)

  Proof sketch: All follow directly from the definition of DT(P) in VD(P). By definition of VD(P), we know that p,q,r are c's nearest neighbors.

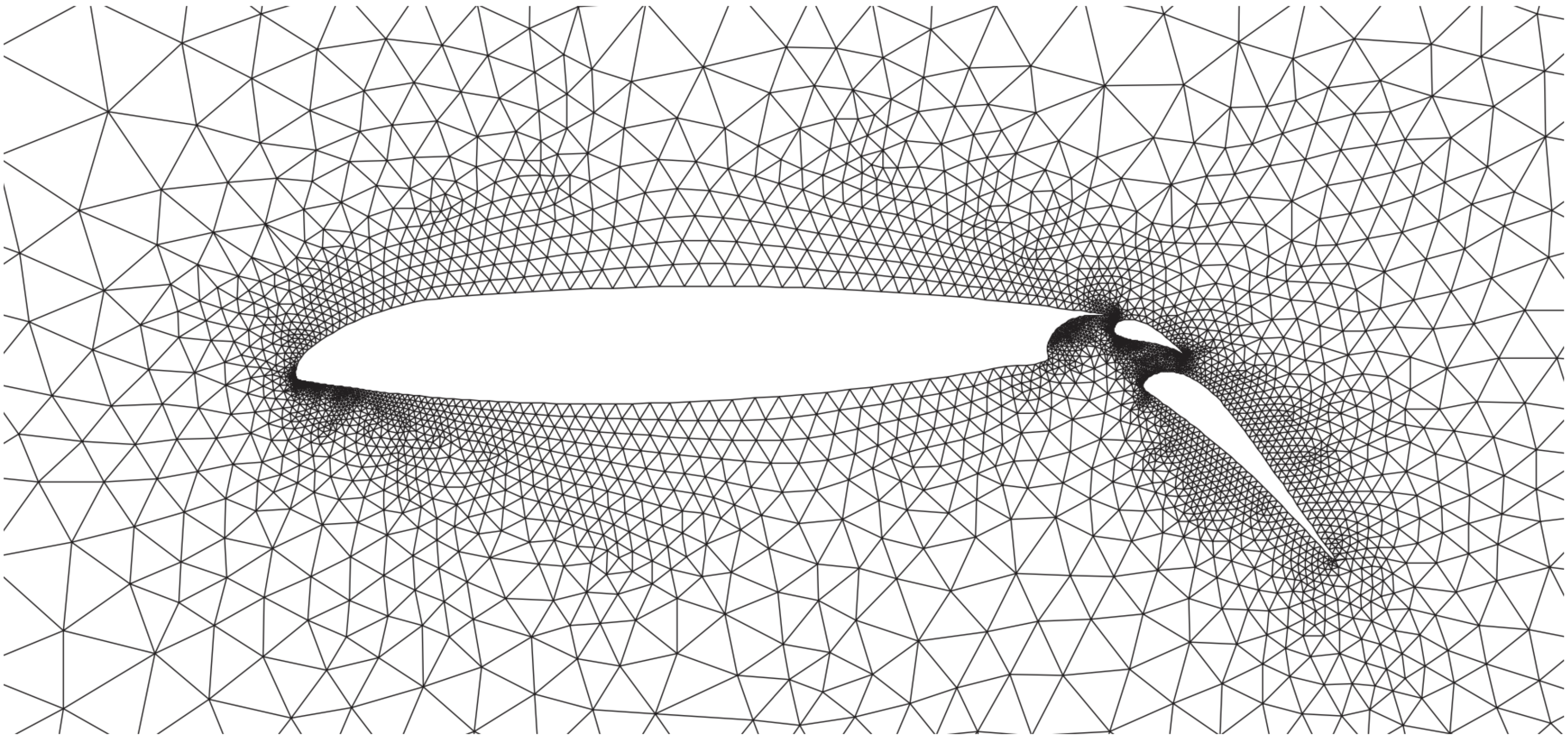
- **Characterization I**: Let T be a triangulation of P.
  Then T=DT(P) $\Leftrightarrow$ The circumcircle of any triangle in T is empty.

non-empty circumcircle



Application in scientific computing

Commonly used in finite element mesh generation to generate a mesh after some other method (e.g. quadtrees, week 3) has been used to place points for vertices

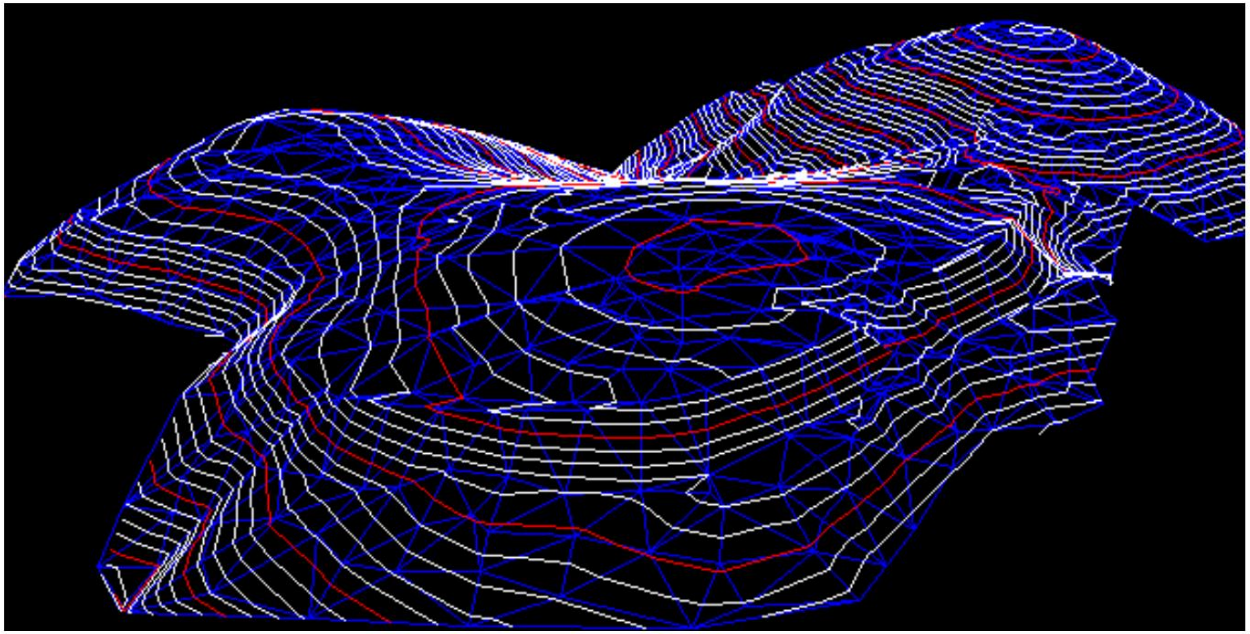


Application in geographic information systems

Triangulation of points with elevations $\Rightarrow$ surface in 3d

Given irregularly placed measurements of ground elevation, connect to form 3d model of ground surface

Called a "triangulated irregular net"



Application in face recognition

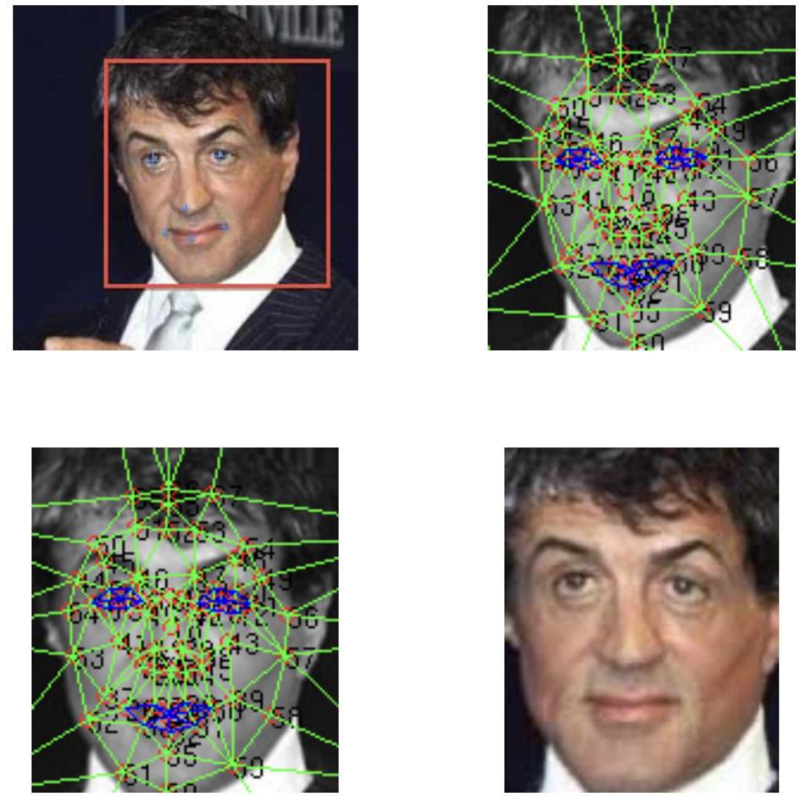
"DeepFace": used by Facebook to recognize people in photos from 2014 to 2021 (stopped for legal reasons, not technical problems)

Uses six points ($2 \times$ eyes, nose, $3 \times$ mouth) to fit 3d generic model

Map 67 "fiducial marks" on 3d model back to 2d image; Delaunay triangulate

Linearly map each triangle to warp to symmetric "frontal" appearance

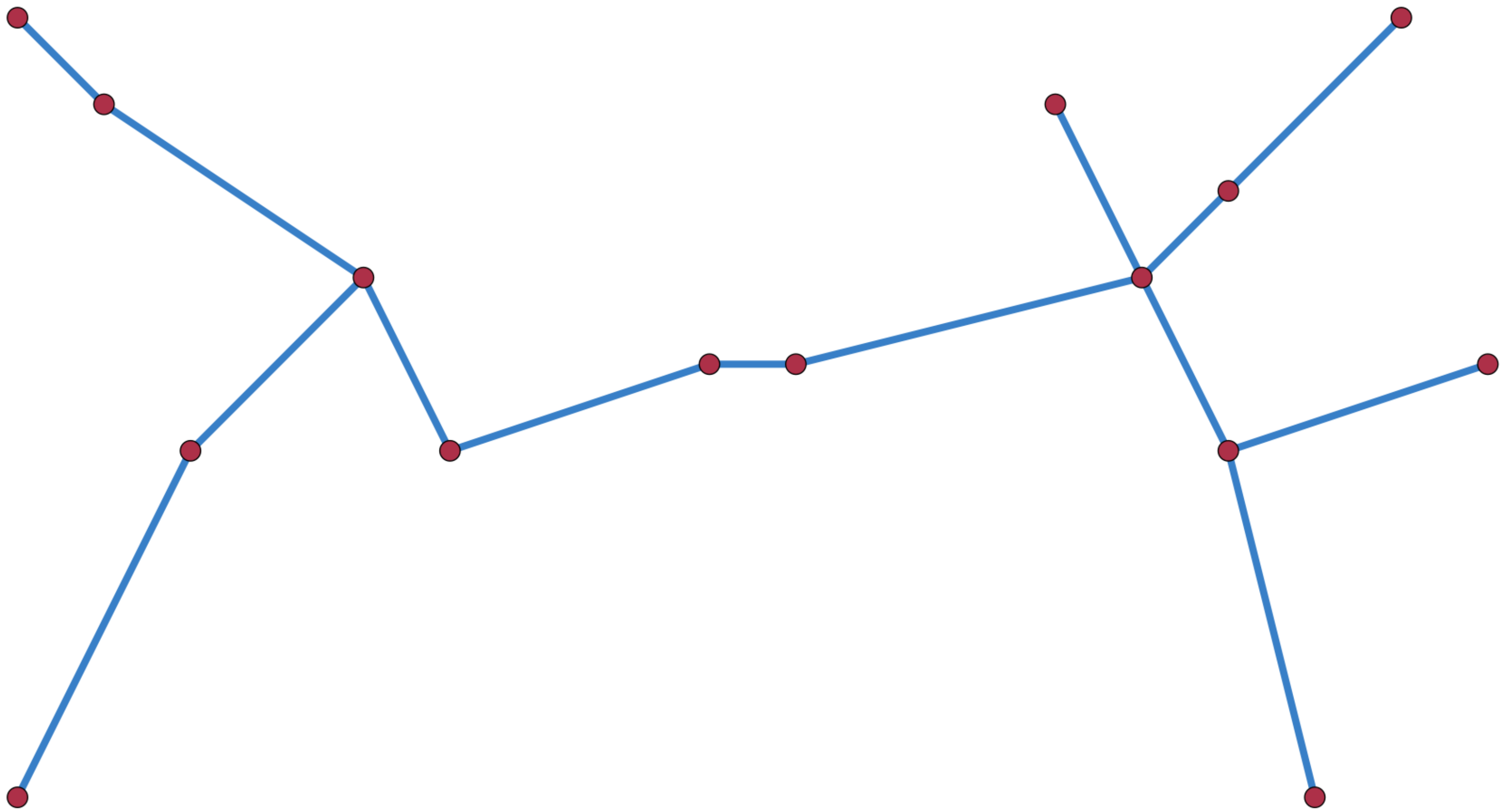
Result is passed to a deep neural net



[Taigman et al. 2014]

Euclidean minimum spanning tree

Connect given points by a tree of minimum total edge length



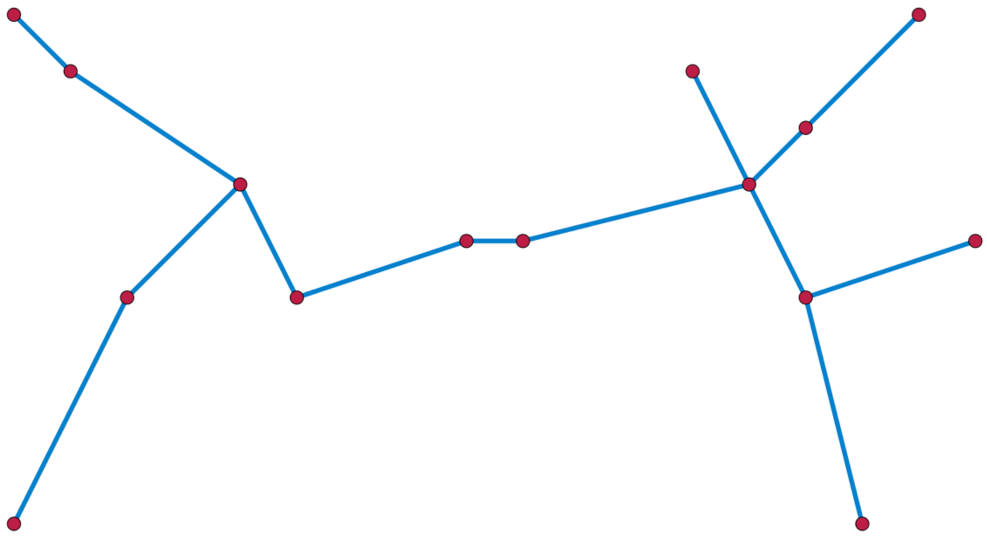
Applications of minimum spanning tree

Original application: making physical connections between geographic locations (power grid) with low construction cost

Clustering: delete longest edge $\Rightarrow$ two clusters as far from each other as possible

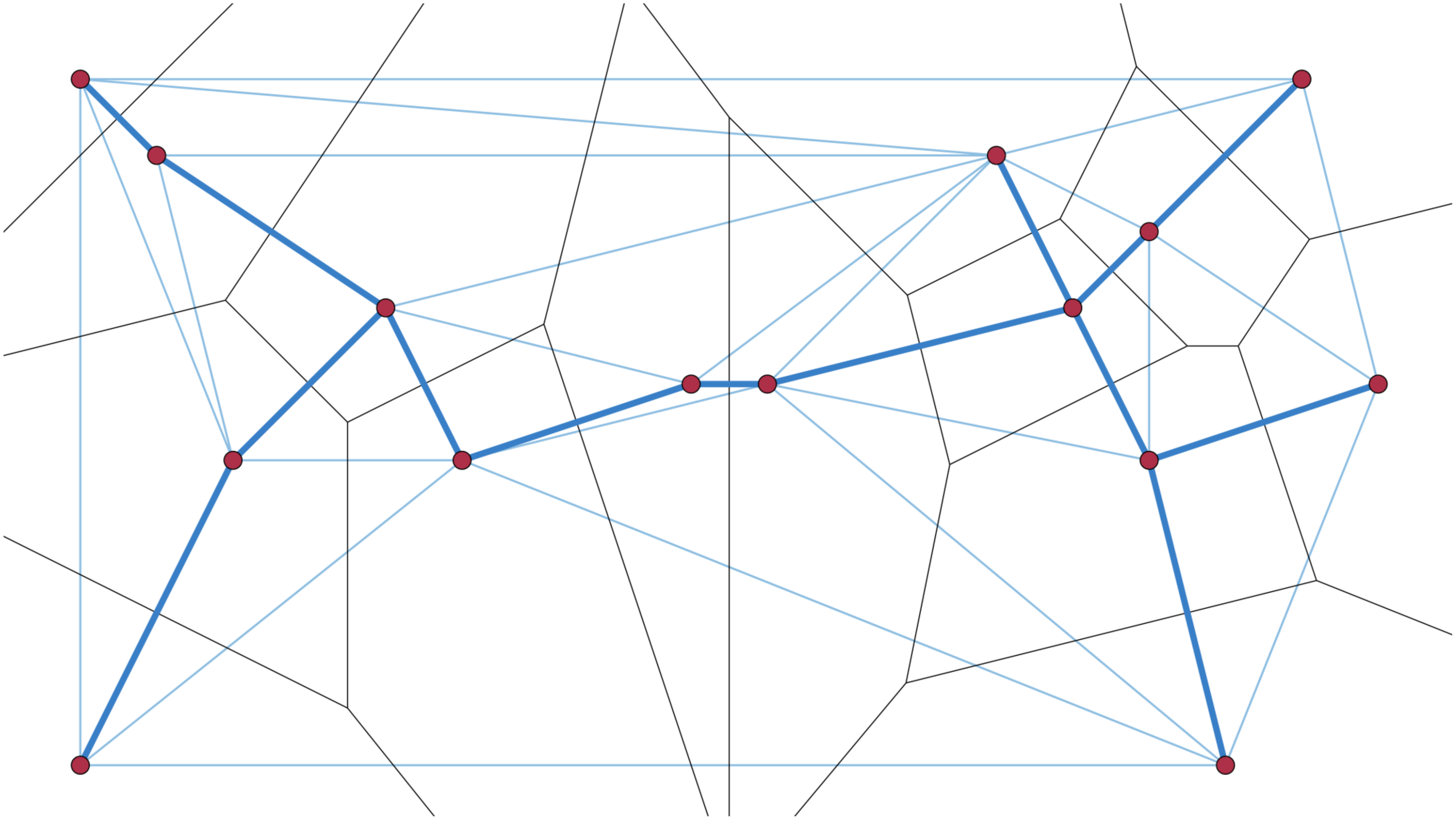
Generating a one-dimensional approximation to the shape of a cloud of points

Approximating traveling salesperson tour
(tree traversal order gives tour of length $\leq 2 \times$ optimal length)



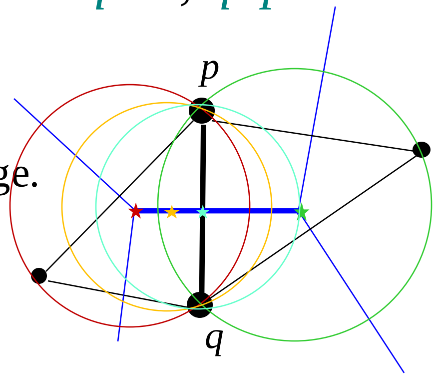
Minimum spanning tree property

Every edge of the minimum spanning tree is in the DT

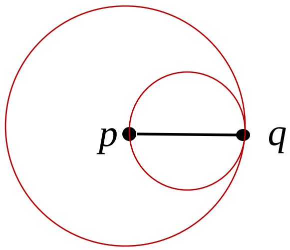


All Nearest Neighbors

- All nearest neighbors: Find for each $p \in P$ its nearest neighbor $q \in P$; $q \neq p$.
  - **Empty circle property:** $p,q \in P$ are connected by an edge in DT(P) $\Leftrightarrow$ there exists an empty circle passing through p and q.
    Proof: "$\Rightarrow$": For the Delaunay edge pq there must be a Voronoi edge. Center a circle through p and q at any point on the Voronoi edge, this circle must be empty.
    "$\Leftarrow$": If there is an empty circle through p and q, then its center c has to lie on the Voronoi edge because it is equidistant to p and q and there is no site closer to c.



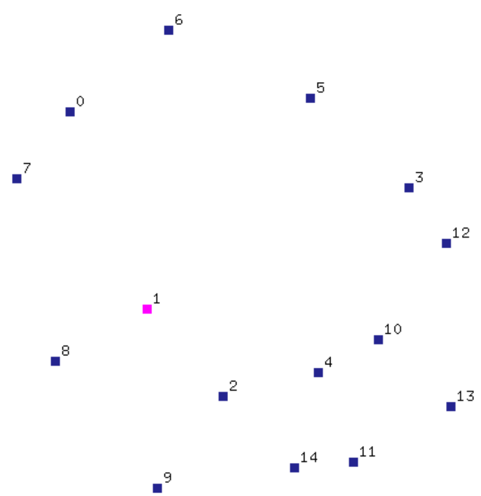
  - **Claim:** In DT(P), every $p \in P$ is adjacent to its nearest neighbors.
    Proof: Let $q \in P$ be a nearest neighbor adjacent to p in DT(P). Then the circle centered at p with q on its boundary has to be empty, so the circle with diameter pq is empty and pq is a Delaunay edge.

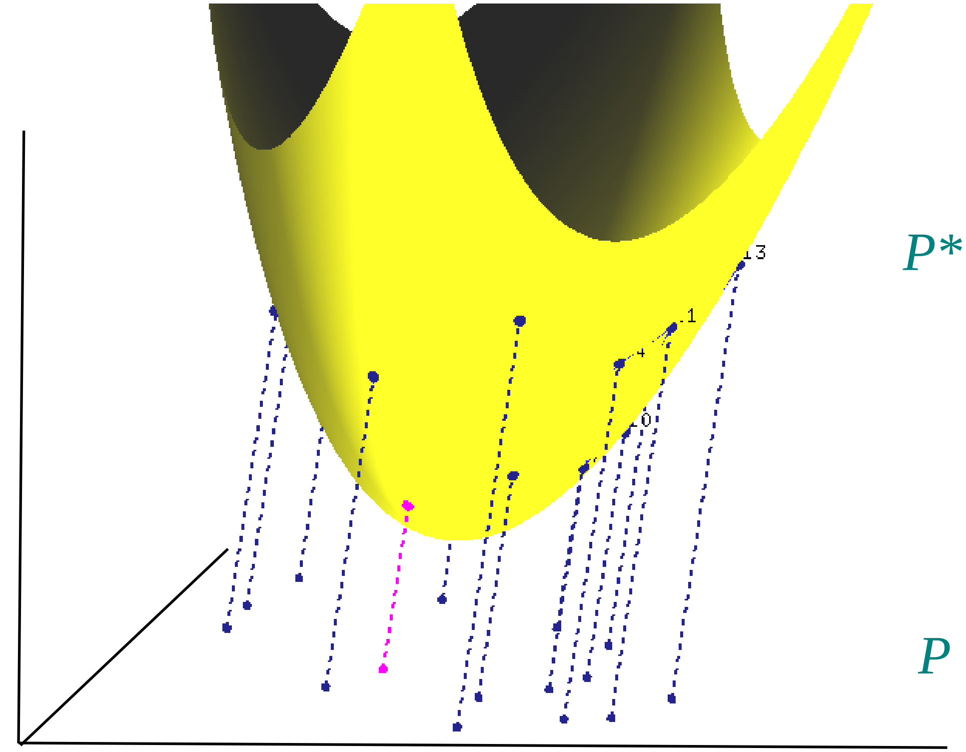


  - **Algorithm:** Find all nearest neighbors in O(n) time: Check for each $p \in P$ all points connected to p with a Delaunay edge.

DT and 3D CH

Theorem: Let $P=\{p_1,\ldots,p_n\}$ with $p_i=(a_i, b_i, 0)$. Let $p^*_i=(a_i, b_i, a^2_i+ b^2_i)$ be the vertical projection of each point p_i onto the paraboloid $z=x^2+ y^2$. Then DT(P) is the orthogonal projection onto the plane $z=0$ of the lower convex hull of $P^*=\{p^*_1,\ldots,p^*_n\}$.

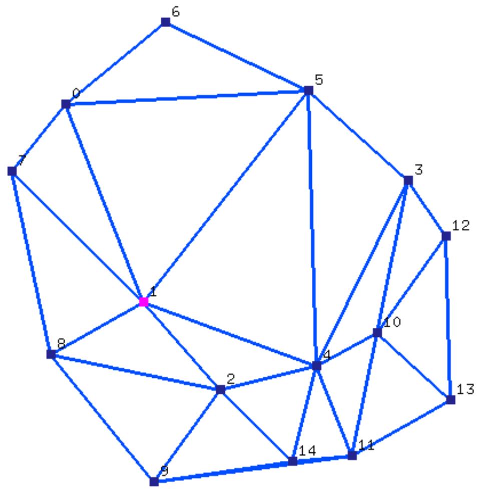


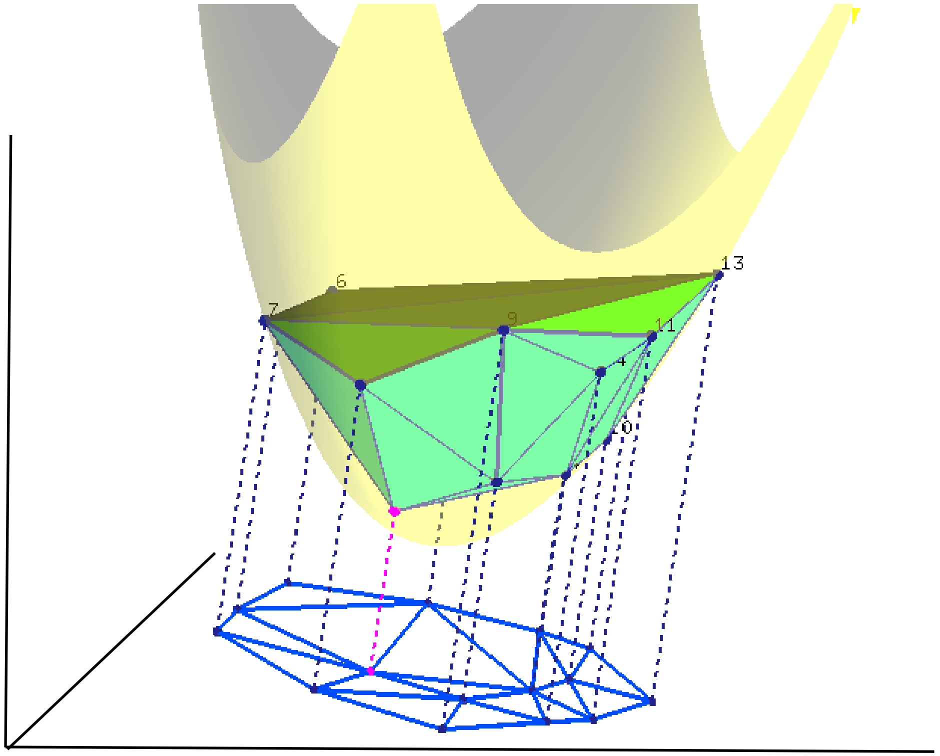


Pictures generated with Hull2VD tool available at http://www.cs.mtu.edu/~shene/NSF-2/DM2-BETA

DT and 3D CH

Theorem: Let $P=\{p_1,\ldots,p_n\}$ with $p_i=(a_i, b_i,0)$. Let $p'_i=(a_i, b_i, a^2_i+ b^2_i)$ be the vertical projection of each point p_i onto the paraboloid $z=x^2+ y^2$. Then $DT(P)$ is the orthogonal projection onto the plane $z=0$ of the lower convex hull of $P'=\{p'_1,\ldots,p'_n\}$.

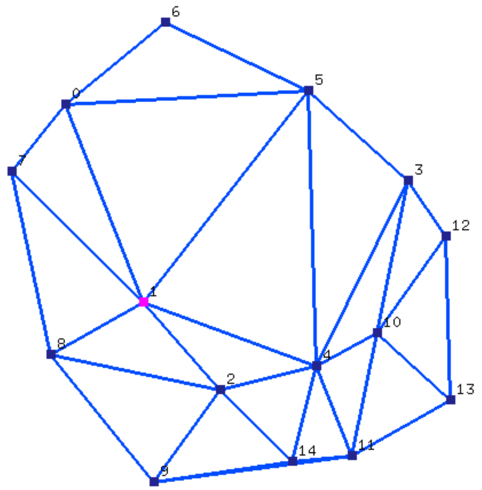


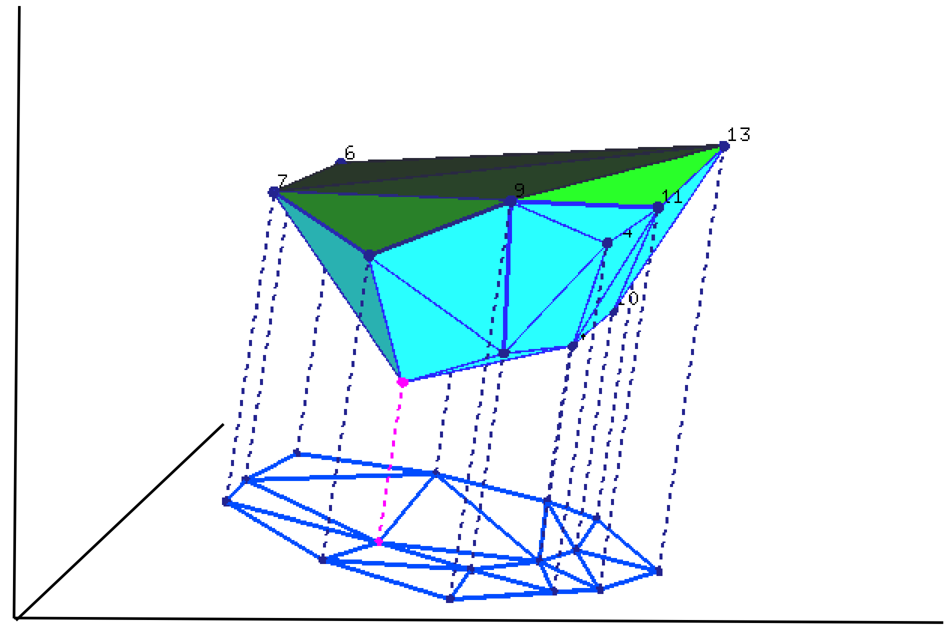


Pictures generated with Hull2VD tool available at http://www.cs.mtu.edu/~shene/NSF-2/DM2-BETA

DT and 3D CH

Theorem: Let $P=\{p_1,\ldots,p_n\}$ with $p_i=(a_i, b_i,0)$. Let $p'_i=(a_i, b_i, a^2_i+ b^2_i)$ be the vertical projection of each point p_i onto the paraboloid $z=x^2+ y^2$. Then $\mathrm{DT}(P)$ is the orthogonal projection onto the plane $z=0$ of the lower convex hull of $P'=\{p'_1,\ldots,p'_n\}$.





Pictures generated with Hull2VD tool available at http://www.cs.mtu.edu/~shene/NSF-2/DM2-BETA

DT and 3D CH

Theorem: Let $P=\{p_1,\ldots,p_n\}$ with $p_i=(a_i, b_i,0)$. Let $p'_i=(a_i, b_i, a^2_i+ b^2_i)$ be the vertical projection of each point p_i onto the paraboloid $z=x^2+ y^2$. Then DT(P) is the orthogonal projection onto the plane $z=0$ of the lower convex hull of $P'=\{p'_1,\ldots,p'_n\}$.

p'_i, p'_j, p'_k form a (triangular) face of LCH(P')

$\Leftrightarrow$

The plane through p'_i, p'_j, p'_k leaves all remaining points of P above it

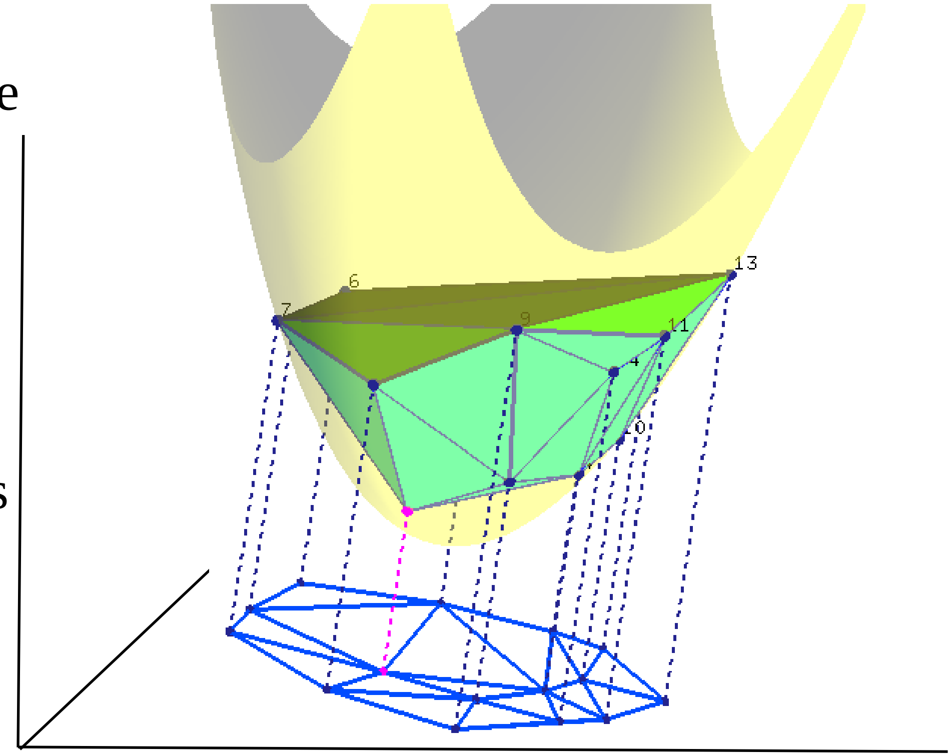
property of unit paraboloid

$\Leftrightarrow$

The circle through p_i, p_j, p_k leaves all remaining points of P in its exterior

$\Leftrightarrow$

p_i, p_j, p_k form a triangle of DT(P)



Slide adapted from slides by Vera Sacristan.