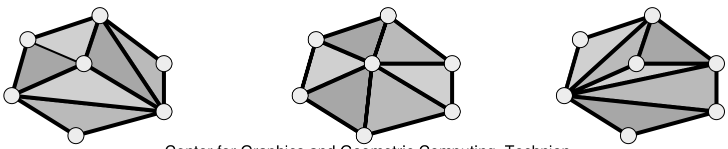
Computational Geometry

Chapter 9

Delaunay Triangulation

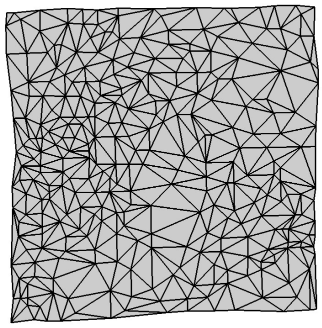
Triangulations

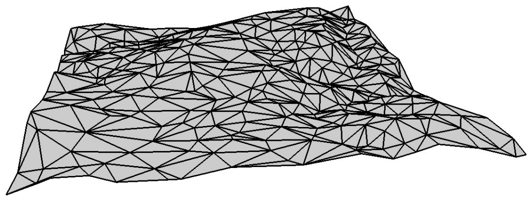
- A *triangulation* of set *S* of points in the plane is a *partition* of the convex hull of the set into triangles whose vertices are the points, and do not contain other points. (Why is there always a triangulation?!)
- An alternative definition: A maximal collection of line-segments inside CH(*S*) whose endpoints are points of *S*. (These segments form the triangles.)
- There are an exponential number of triangulations of a point set. Best known bound: O(43^n) [Sharir and Welzl, 2006].



Motivation

- Assume a height value is associated with each point.
- A triangulation of the points defines a *piecewise-linear* surface of triangular patches.

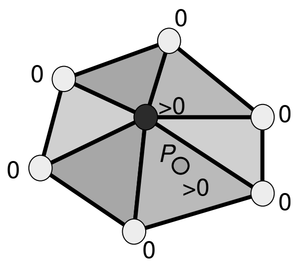


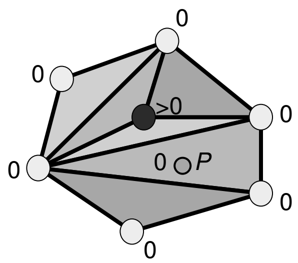


2D 3D

Piecewise-Linear Interpolation

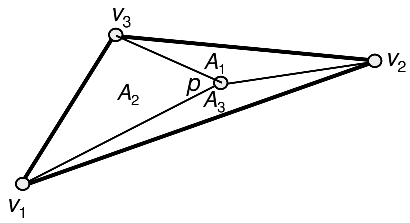
- The height of a point *P* inside a triangle is determined by the height of the triangle vertices, and the location of *P*.
- The result depends on the triangulation.





Barycentric Coordinates

- Any point inside a triangle can be expressed *uniquely* as a *convex* combination of the triangle vertices:



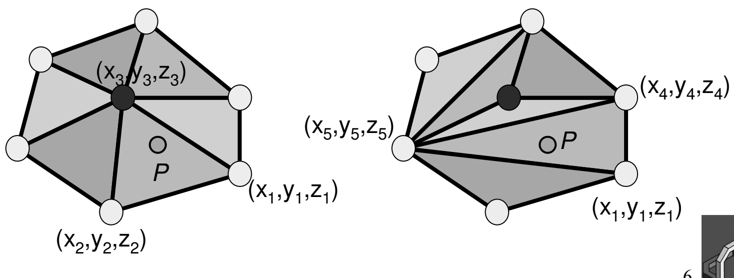
$$p = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$$

$$\alpha_i = \frac{A_i}{A_1 + A_2 + A_3} \quad \text{for } 1 \le i \le 3$$

$$\alpha_i \ge 0, \quad \alpha_1 + \alpha_2 + \alpha_3 = 1$$

(back to) Piecewise-Linear Interpolation

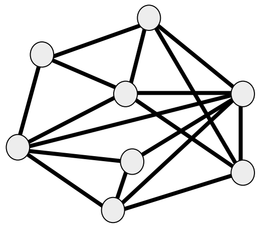
$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 = x_p$$
$$\alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3 = y_p$$
$$\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 - z_p = 0$$
$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$



An O(n^3)-Time Triangulation Algorithm

- Repeat
  - Select two sites.
  - If the edge connecting them does not intersect previously kept edges, keep it.

  Until all faces are triangles.

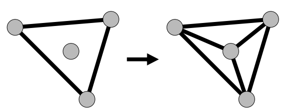


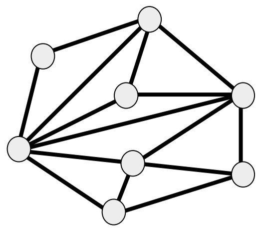
- Question: Why O(n^3) time?
- Question: Why is the algorithm guaranteed to stop before running out of edges?
- Answer: Because every nontriangular face has a diagonal that was not processed yet. (Why?!)

Center for Graphics and Geometric Computing, Technion 7

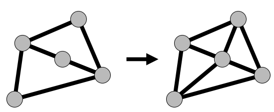
An O($n \log n$)-Time Triangulation Algorithm

- Construct the convex hull of the points, and connect one arbitrary vertex to all others.
- Insert the other sites one after the other…
- Two possibilities:
  - Point inside a triangle: One triangle becomes three.





  - Point on an edge: Two triangles become four.



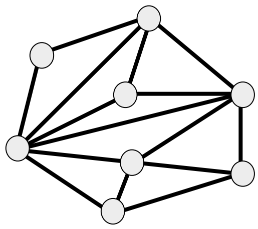
Question: Why O($n \log n$) time?

Center for Graphics and Geometric Computing, Technion 8

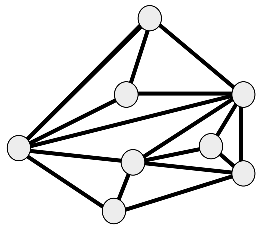
Number of Triangles

- The number of triangles t in a triangulation of n points depends on the number of vertices h on the convex hull: $t = (h-2) + 2(n-h) = 2n-h-2$.

$n = 8$



$h = 6 \rightarrow t = 8$

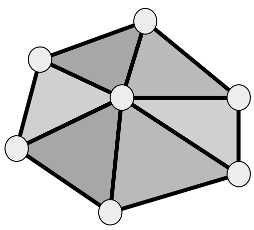


$h = 5 \rightarrow t = 9$

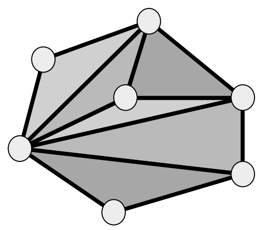
Quality Triangulations

- Consider a triangulation T.
- Let $\alpha(T) = (\alpha_1, \alpha_2, \ldots, \alpha_{3t})$ be the vector of angles in the triangulation T sorted in increasing order.
- A triangulation T_1 is "better" than T_2 if $\alpha(T_1) > \alpha(T_2)$ (compared lexicographically).
- The **Delaunay triangulation** is the "best" (avoiding, as much as possible, long skinny triangles).

Good:



Bad:



Thales's Theorem

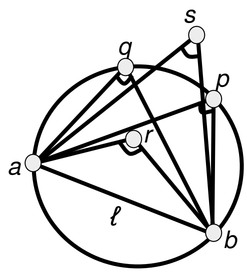
- Theorem:

  Let *C* be a circle, and *ℓ* a line intersecting *C* at the points *a* and *b*. Let *p*, *q*, *r*, and *s* be points lying on the same side of *ℓ*, where *p* and *q* are on *C*, *r* inside *C*, and *s* outside *C*. Then:

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

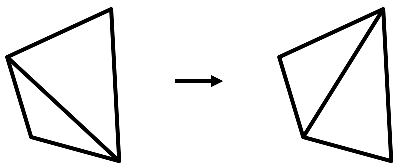
- Proof omitted.

  (Thales proved the theorem directly; one can deduce it from the sine theorem.)



Improving a Triangulation

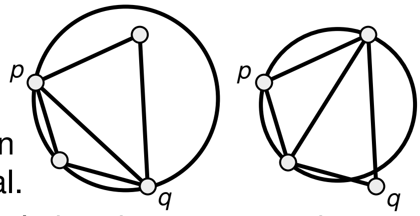
- In any convex quadrangle, an *edge flip* is possible. (Why? Why isn't it possible in a concave triangle?)
- Claim: If this flip *improves* the triangulation locally, it also improves the global triangulation. (Why?)



- If an edge flip improves the triangulation (locally and hence globally), the original edge is called *illegal*.

Illegal Edges

- **Lemma:** An edge *pq* is illegal iff any of its opposite vertices is inside the circle defined by the other three vertices.
- **Proof:** By Thales's theorem.
- Moreover, a convex quadrangle in general position has exactly one legal diagonal.



- **Theorem:** A Delaunay triangulation does not contain illegal edges. (Otherwise it can be improved locally.)
- **Corollary:** A triangle is Delaunay iff the circle through its vertices is empty of other sites.
- **Observation:** The Delaunay triangulation is not unique if more than three sites are cocircular.

An $\Theta(n^4)$-Time Delaunay Triangulation

- For all triples of sites:
  - If the circle through the triple of sites does not contain any other sites, keep the triangle whose vertices are the triple.
- Complexity: $\Theta(n^3)$ triples, $\Theta(n)$ work on each triple; Total: $\Theta(n^4)$ time.

  (Space complexity: $\Theta(n)$.)

The In-Circle Test

Theorem: If *a*,*b*,*c*,*d* form a CCW convex polygon, then *d* lies in the circle determined by *a*, *b*, and *c* iff:

$$\det \begin{pmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{pmatrix} > 0$$

Proof:

We prove that equality holds if the points are cocircular.

There exists a center *q* and radius *r* such that:

$$(a_x - q_x)^2 + (a_y - q_y)^2 = r^2$$

Similarly for *b*, *c*, *d*.

In vector notation:

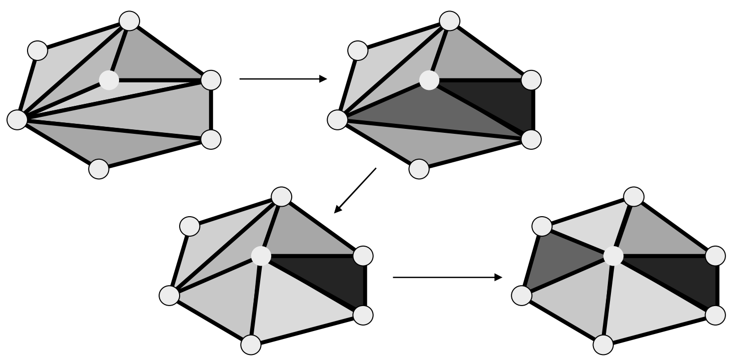
$$\begin{pmatrix} a_x^2 + a_y^2 \\ b_x^2 + b_y^2 \\ c_x^2 + c_y^2 \\ d_x^2 + d_y^2 \end{pmatrix} - 2q_x \begin{pmatrix} a_x \\ b_x \\ c_x \\ d_x \end{pmatrix} - 2q_y \begin{pmatrix} a_y \\ b_y \\ c_y \\ d_y \end{pmatrix} + (q_x^2 + q_y^2 - r^2) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 0$$

So these four vectors are linearly dependent, and hence their determinant vanishes.

Corollary: $d \in \circ(a,b,c)$ iff $b \in \circ(a,c,d)$ iff $c \in \circ(b,a,d)$ iff $a \in \circ(b,c,d)$.

Naive Delaunay Algorithm

- ❑ Start with an arbitrary triangulation.
- ❑ Flip any illegal edge until no more exist.



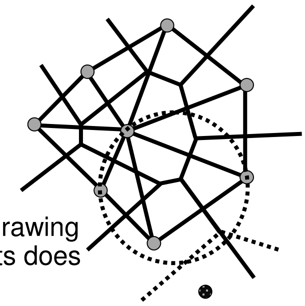
Naive Delaunay Algorithm (cont.)

- Question: Why does the algorithm terminate?
- Answer: Because every flip increases the vector angle, and there are finitely-many such vectors.
- However, this algorithm is in practice very slow.

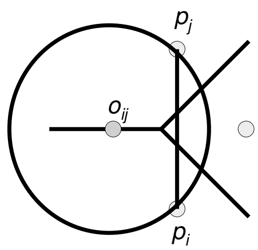
- Question: Why does the algorithm converge to the optimum triangulation?
- Answer: Because there are no local maxima (proof deferred).

Delaunay Triangulation by Duality

- Draw the Delaunay graph (the dual graph of the Voronoi diagram) by connecting each pair of neighboring sites in the Voronoi diagram.
- If no four points are cocircular, then the Delaunay graph is triangulated,
- General position assumption: There are no four cocircular points.
- We need to prove:
  - Correctness of this duality. That is, that drawing the Delaunay graph with straight segments does not cause any segment intersection.
  - That this triangulation indeed maximizes the angle vector (and, hence, it is the Delaunay triangulation).
- **Corollary:** The Delaunay triangulation (DT) of *n* points can be computed in O(*n* log *n*) time.

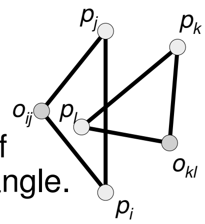


Proof of Planarity of Delaunay Triangulation

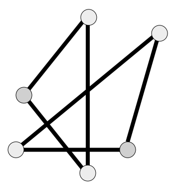


- Let S be a set of sites, and let DT(S) be the dual graph of VD(S).
- Let p_ip_j be an edge of DT(S). It is so because cells of p_i and p_j in VD(S) are neighbors in VD(S). Hence, there exists an **empty** circle passing through p_i,p_j and whose center o_{ij} is on their bisector (the edge of VD(S) separating between the cells of p_i and p_j).
- Assume for contradiction that p_ip_j intersects another edge p_kp_l in DT(S).
- Observe the possible interactions between the triangles $\Delta o_{ij}p_ip_j$ and $\Delta o_{kl}p_kp_l$… (next slide)

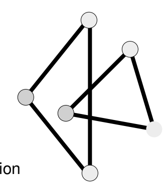
Planarity Proof (cont.)



- Case A (one triangle contains a yellow vertex of the other triangle):
  Impossible, since the circumscribing circle of the first triangle is empty, hence also the triangle.



- Case B (no triangle contains a vertex of the other triangle):
  Cannot avoid an intersection of a pair of white edges, which is impossible, because the white edges are fully contained in **disjoint** Voronoi cells.



- Case C (one triangle contains a green vertex of the other triangle) is possible.
  Question: Why isn't it a contradiction?

Delaunay Triangulation: Main Property

- **Theorem**:
  Let S be a set of points in the plane. Then,
  (i) $p_i, p_j, p_k \in S$ are vertices of a triangle (face) of DT(S)
  $\leftrightarrow$ The circle passing through p_i, p_j, p_k is empty;
  (ii) $\overline{p_i, p_j}$ (for $p_i, p_j \in S$) is an edge of DT(S)
  $\leftrightarrow$ There exists an empty circle passing through p_i, p_j.
- **Proof**: Dualize the Voronoi-diagram theorem.
- Corollary:
  A triangulation T(S) is DT(S)
  $\leftrightarrow$ Every circumscribing circle of a triangle $\Delta \in$ T(S) is empty.

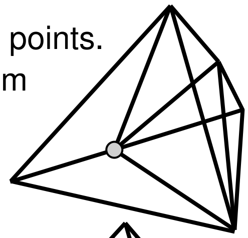
Wrapping Up

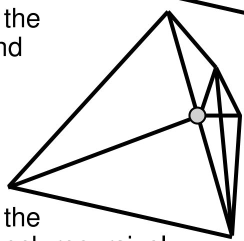
- **Theorem**:
  Let S be a set of points in the plane, and let T(S) be a triangulation of S. Then,
  T(S) = DT(S) $\leftrightarrow$ T(S) is legal.
- **Proof**: Follows from the definitions of a legal edge and triangulation. (Exercise!)
- **Corollary**: DT(S) maximizes the vector angle.
- Since DT(S) is unique, there is only one legal triangulation, and thus, there are no local maxima in the edge-flip algorithm. That is, it converges to DT(S).

An O(n log n)-Time Delaunay Algorithm

A **Randomized** incremental algorithm:

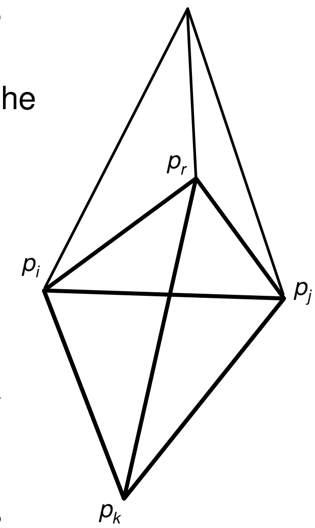
- Form a bounding triangle Δ_0 enclosing all points.
- Add the points one after another in random order and update the triangulation.
- If the point is inside an existing triangle:
  - Connect the point to the triangle vertices.
  - Check if a flip can be performed on any of the three triangle edges. If so, flip the edge and check recursively the neighboring edges (opposite to the new point).
- If the site is on an existing edge:
  - Replace the edge with four new edges.
  - Check if a flip can be performed on any of the opposite edges. If so, flip the edge and check recursively the neighboring edges (opposite to the new point).



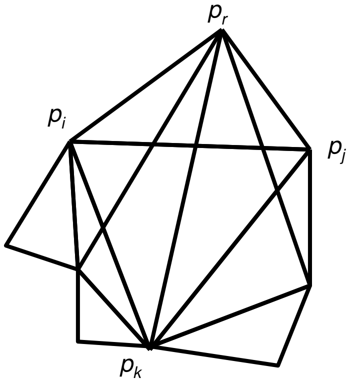


Flipping Edges

- A new point p_r was added, causing the creation of the edges p_ip_r and p_jp_r.
- The legality of the edge p_ip_j (with opposite vertex) p_k is checked.
- If p_ip_j is illegal, perform a flip, and recursively check edges p_ip_k and p_jp_k, the new edges opposite to p_r.
- Notice that the recursive call for p_ip_k cannot eliminate the edge $p_r\,p_k$.
- **Note:** All edge flips replace edges opposite to the new vertex by edges adjacent to it!

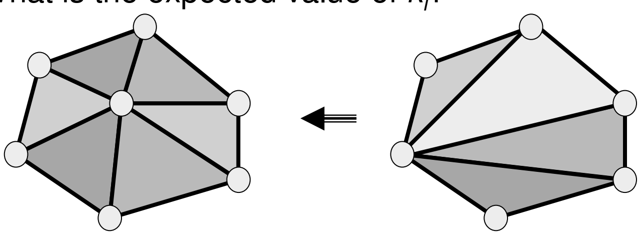


Flipping Edges: Example



Number of Triangles

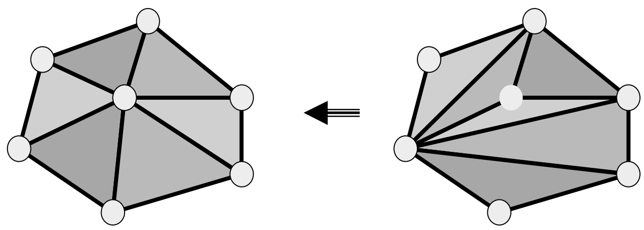
- **Theorem:** The expected number of triangles created in the course of the algorithm (some of which also disappear) is at most $9n+1$.
- **Proof:**
  During the insertion of point p_i, k_i new edges are created: 3 new initial edges, and k_i-3 due to flips. Hence, the number of new triangles is at most $3+2(k_i-3) = 2k_i-3$.
  (A point on an edge results in $2k_i-4$ triangles.)
- What is the expected value of k_i?



Number of Triangles (cont.)

- Recall that the Voronoi diagram has at most $3N - 6$ edges, where N is the number of vertices.
- The number of edges in a graph and its dual are identical.
- Taking into account the initial triangle Δ_0, after inserting i points, there are at most $3(i+3) - 6 = 3i + 3$ edges.

  Three of them belong to Δ_0, so we are left with at most $3i$ internal edges that are adjacent to the input points.

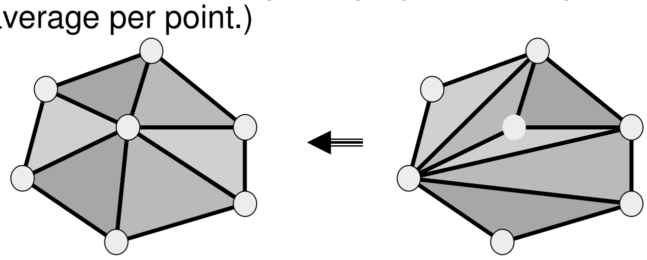


Center for Graphics and Geometric Computing, Technion

27

Number of Triangles (cont.)

- The sum of all vertex degrees is thus at most $2 \cdot 3i = 6i$.
- On the average, the degree of each vertex is only 6 ! But this is exactly the number of new edges!
- Hence, the expected number of triangles created in the ith step is at most $E(2k_i - 3) = 2\,E(k_i) - 3 = 9$.
- Therefore, the expected number of triangles created (and possibly destroyed) for n points is $9n + 1$. (One initial bounding triangle plus 9 triangles on average per point.)



Center for Graphics and Geometric Computing, Technion

28

Algorithm Complexity

- Point location for every point: O(log *n*) time (not shown).
- Flips: Θ(n) expected time in total (for all steps).
- Total expected time: O(*n* log *n*).
- Space: $\Theta(n)$.

Relatives of the Delaunay Triangulation

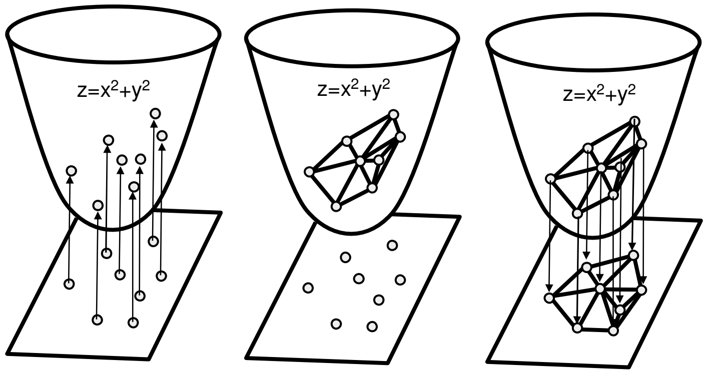
- **Euclidean Minimum Spanning Tree (EMST):**
  A tree of minimum length connecting all the sites.
- **Relative Neighborhood Graph (RNG):**
  Two sites *p*, *q* are connected if

$$d(p,q) \leq \min_{r \in P, r \neq p,q} \max(d(p,r), d(q,r))$$

- **Gabriel Graph (GG):**
  Two sites *p*, *q* are connected if the circle whose *diameter* is *pq* is empty of other sites.

- **Theorem**: EMST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ DT .

Delaunay Triangulation and Convex Hulls



Project the 2D point set onto the 3D paraboloid → Compute the 3D lower convex hull → Project the 3D facets Back to the plane.

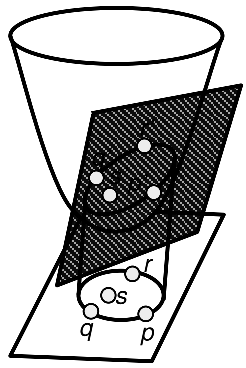
The 2D triangulation is Delaunay!

Proof of Lift-Up

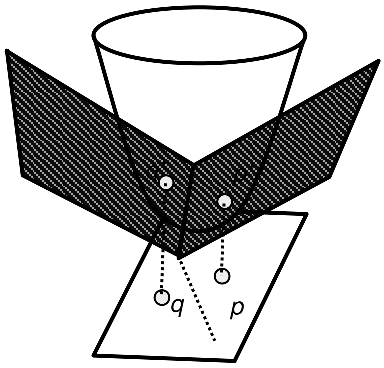
- The intersection of a plane with the paraboloid is an ellipse whose projection to the plane is a circle.
- *s* lies within the circumcircle of *p*,*q*,*r* iff *s*' lies on the lower side of the plane passing through *p*',*q*',*r*'.

↓

- *p*, *q*, *r* ∈ *S* form a Delaunay triangle iff *p*', *q*', *r*' form a face of the convex hull of *S*'.



More about Lifting Up



- Given a set S of points in the plane, associate with each point $p=(a,b)\in S$ the plane $z = 2ax+2by-(a^2+b^2)$, which is tangent to the paraboloid at p', the vertical projection of p onto the paraboloid.
- VD(S) is the vertical projection onto the XY plane of the boundary of the convex polyhedron that is the intersection of the halfspaces above these planes.