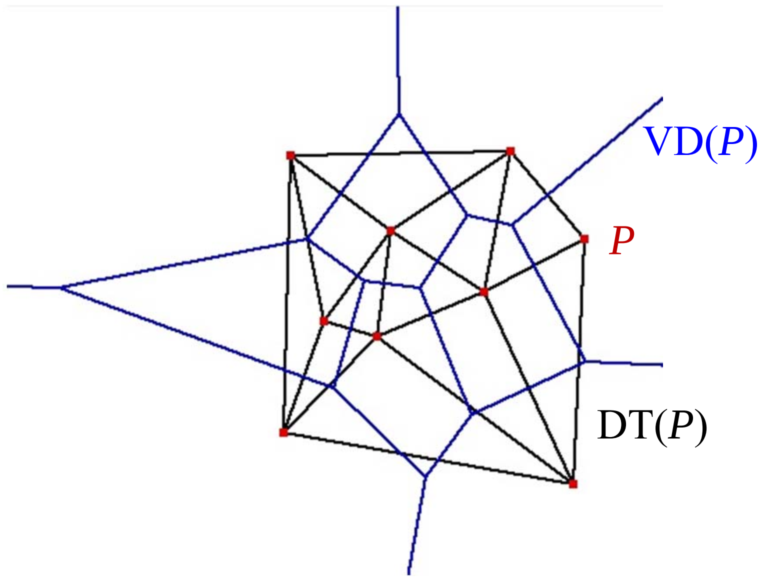
CMPS 6640/4040 Computational Geometry
Spring 2016



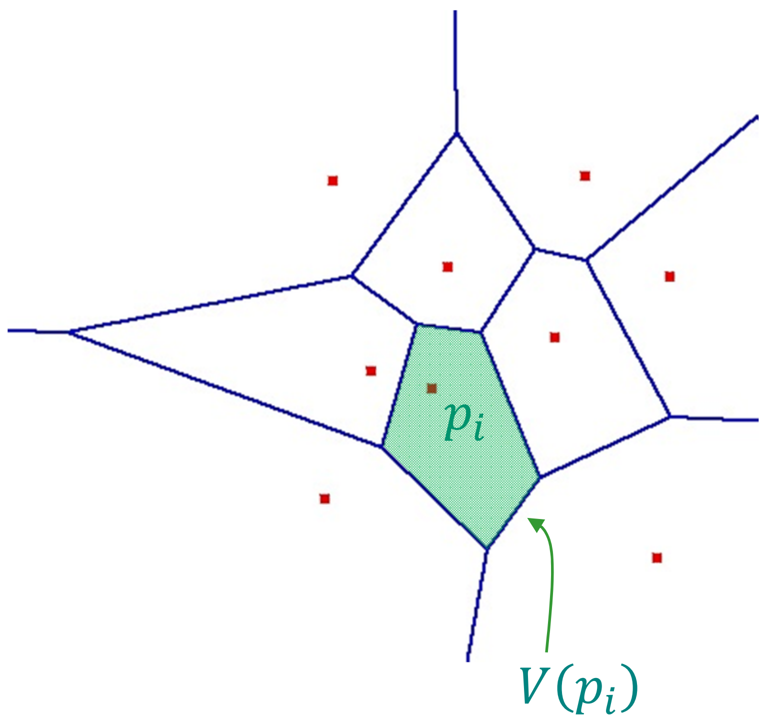
Voronoi Diagrams and Delaunay Triangulations

Carola Wenk

Based on:
Computational Geometry: Algorithms and Applications

Voronoi Diagram
(Dirichlet Tesselation)

- **Given:** A set of point sites $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^2$
- **Task:** Partition $\mathbb{R}^2$ into Voronoi cells
  $V(p_i) = \{q \in R^2 \mid d(p_i, q) < d(p_j, q)$ for all $j \neq i\}$



Applications of Voronoi Diagrams

- Nearest neighbor queries:
  - Sites are post offices, restaurants, gas stations
  - For a given query point, locate the nearest point site in $O(\log n)$ time
    $\rightarrow$ point location

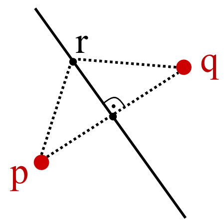
- Closest pair computation (collision detection):
  - Naïve $O(n^2)$ algorithm; sweep line algorithm in $O(n \log n)$ time
  - Each site and the closest site to it share a Voronoi edge
    $\rightarrow$ Check all Voronoi edges (in $O(n)$ time)

- Facility location: Build a new gas station (site) where it has minimal interference with other gas stations
  - Find largest empty disk and locate new gas station at center
  - If center is restricted to lie within $CH(P)$ then the center has to be on a Voronoi edge

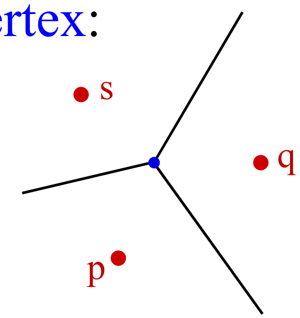
Bisectors

- Voronoi edges are portions of bisectors
- For two points p, q, the bisector $b(p,q)$ is defined as

$$b(p,q) = \{r \in \mathbb{R}^2 \mid d(p,r) = d(q,r)\}$$



- Voronoi vertex:

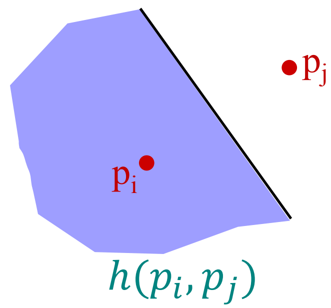


Voronoi cell

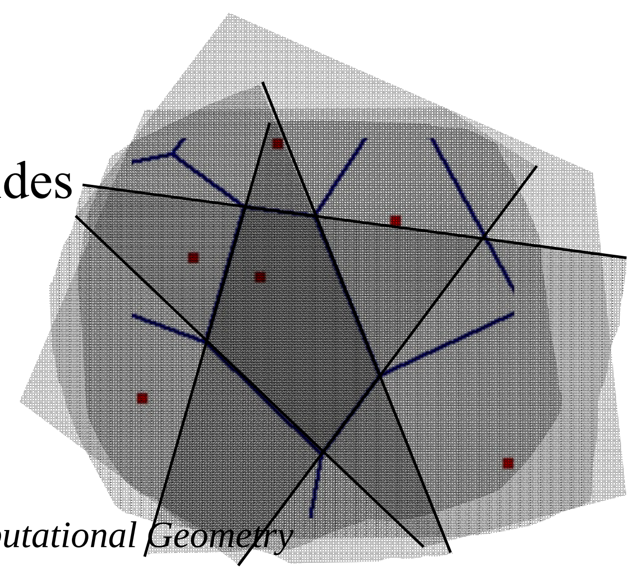
- Each Voronoi cell $V(p_i)$ is convex and

$$V(p_i) = \bigcap_{\substack{p_j \in P \\ j \neq i}} h(p_i, p_j) \;,$$

where $h(p_i, p_j)$ is the halfspace that is defined by bisector $b(p_i, p_j)$ and that contains p_i



⇒ A Voronoi cell has at most $n-1$ sides



Voronoi Diagram

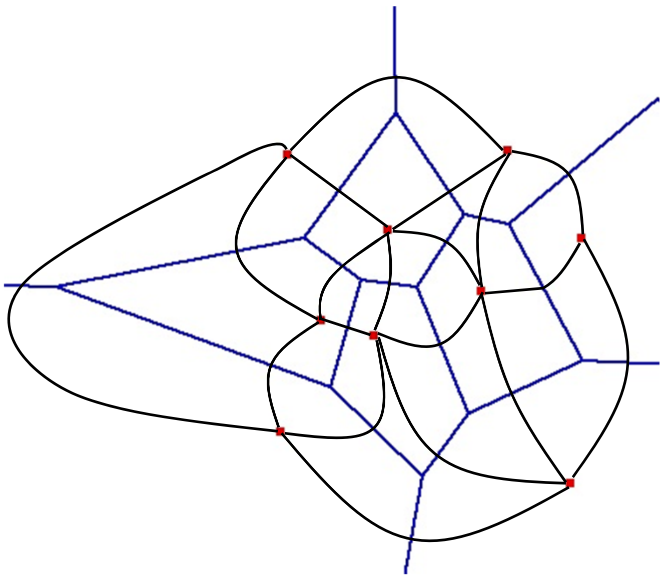
- For $P = \{p_1, \ldots, p_n\} \subseteq R^2$, let the **Voronoi diagram** $VD(P)$ be the planar subdivision induced by all Voronoi cells $VD(p_i)$ for all $i \in \{1, \ldots, n\}$.
  $\Rightarrow$ The Voronoi diagram is a planar embedded graph with vertices, edges (possibly infinite), and faces (possibly infinite)

- **Theorem:** Let $P = \{p_1, \ldots, p_n\} \subseteq R^2$. Let n_v be the number of vertices in $VD(P)$ and let n_e be the number of edges in $VD(P)$. Then

  $n_v \leq 2n - 5$, and

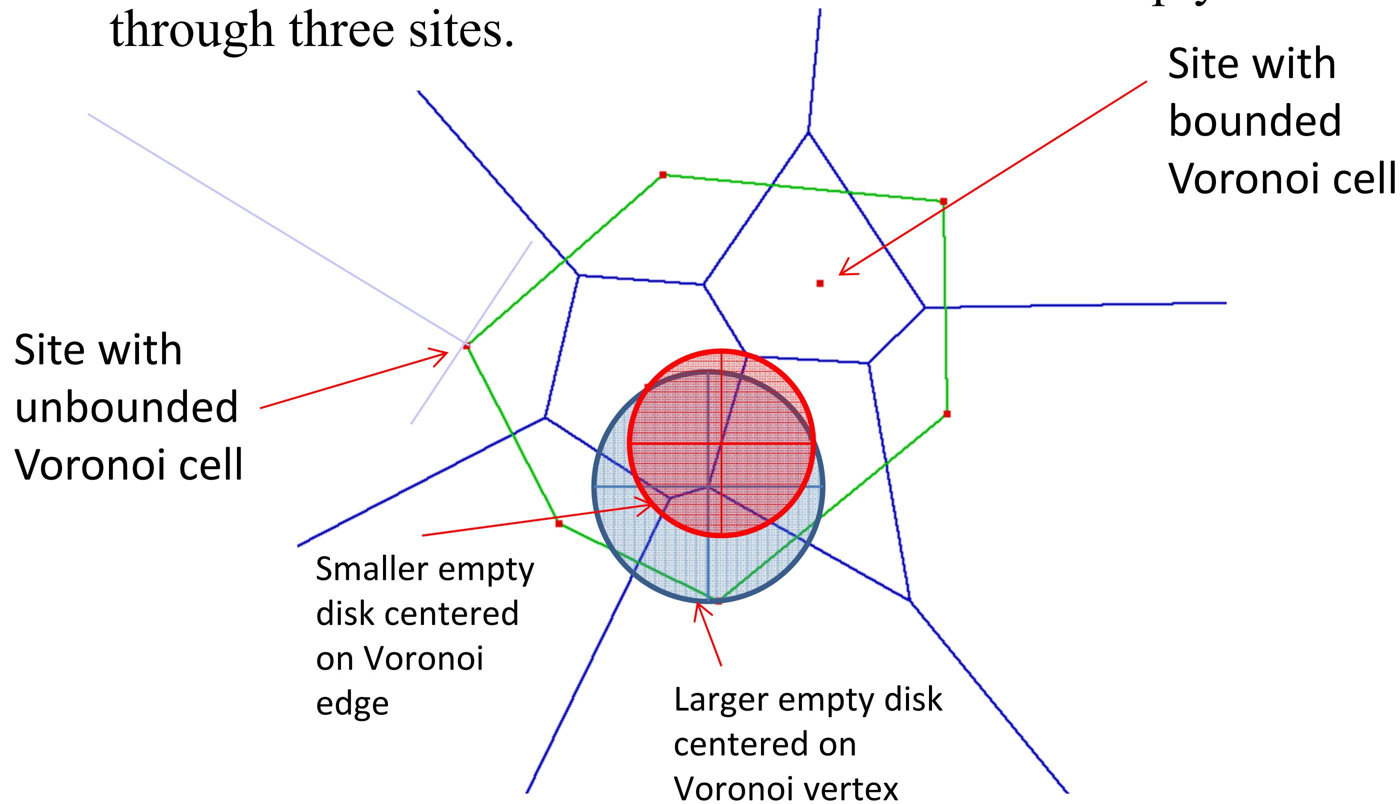
  $n_e \leq 3n - 6$

Proof idea: Use Euler's formula for the dual graph.



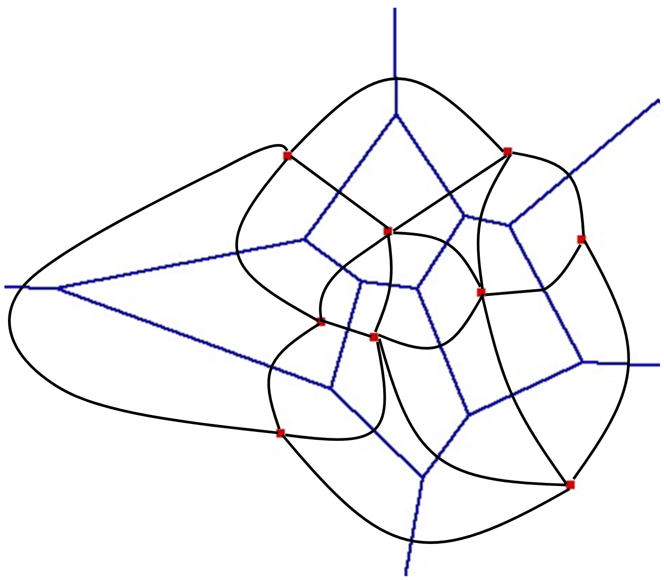
Properties

1. A Voronoi cell $V(p_i)$ is unbounded iff p_i is on the convex hull of the sites.
2. Each point on an edge of the VD is equidistant from its two nearest neighbors p_i and p_j.
3. v is a Voronoi vertex iff it is the center of an empty circle that passes through three sites.

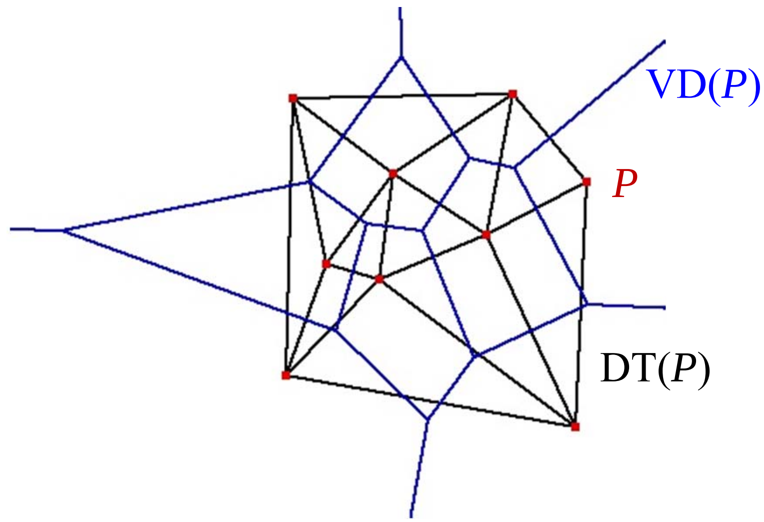


Delaunay Triangulation

- 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):

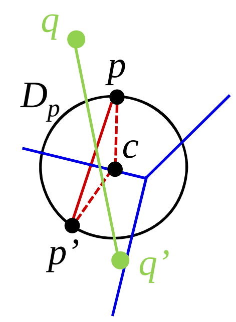


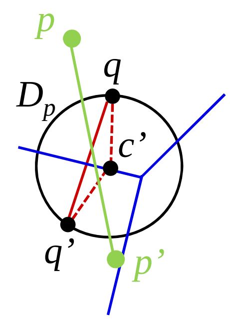
- If P is in general position (no three points on a line, no four points on a circle) then every inner face of DT(P) is indeed a triangle.
- DT(P) can be stored as an abstract graph, without geometric information. (No such obvious storing scheme for VD(P).)

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 $\overline{pp'}$. (i.e., p, p', q, q' are distinct)

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





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

$\Rightarrow$ The edges are not in different Voronoi cells

$\Rightarrow$ Contradiction

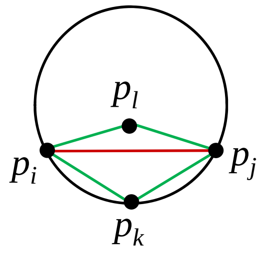
□

Characterization I of DT(P)

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

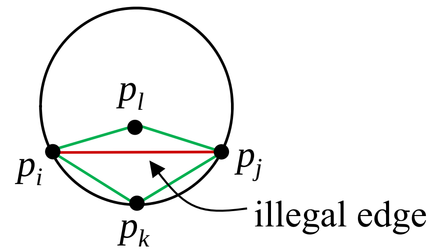
- **Characterization I**: Let T be a triangulation of P.
  Then T=DT(P) $\Leftrightarrow$ The circumcircle of any triangle in T is empty.

non-empty circumcircle

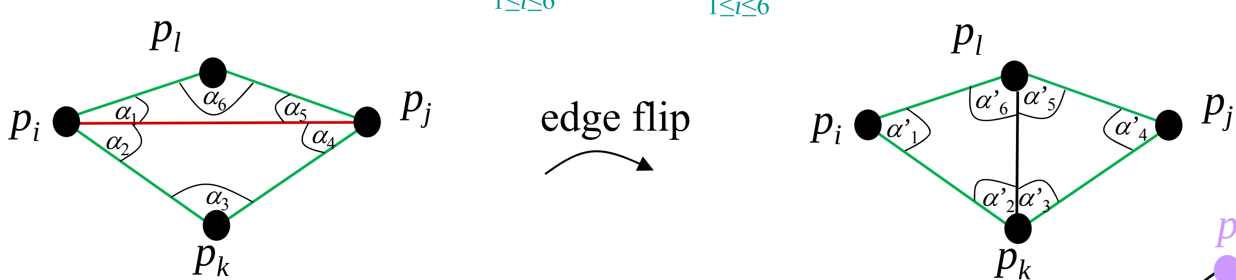


Illegal Edges

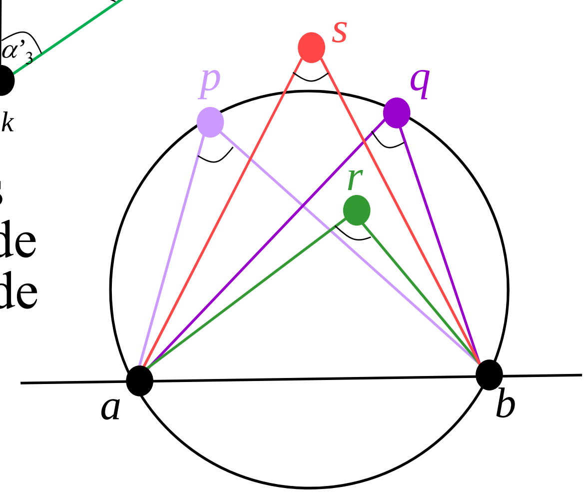
- **Definition:** Let $p_i, p_j, p_k, p_l \in P$.
  Then $\overline{p_i\, p_j}$ is an **illegal edge** $\Leftrightarrow$ p_l lies in the interior of the circle through p_i, p_j, p_k .



- **Lemma:** Let $p_i, p_j, p_k, p_l \in P$.
  Then $p_i\, p_j$ 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, and let r be inside and let s be outside of the circle, such that p,q,r,s lie on the same side of the line through a, b.
  Then $\angle a,s,b < \angle a,q,b = \angle a,p,b < \angle a,r,b$



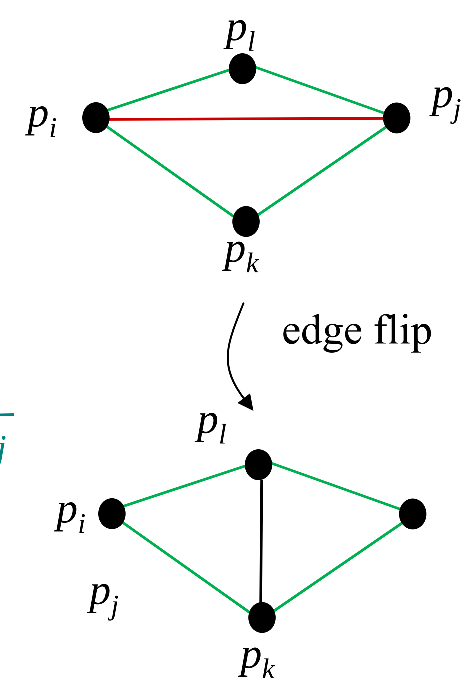
Characterization II of DT(P)

- **Definition:** A triangulation is called legal if it does not contain any illegal edges.
- **Characterization II**: 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 p_i, p_j, p_k, p_l be the quadrilateral containing $\overline{p_i p_j}$
      Remove $\overline{p_i p_j}$ and add $\overline{p_k p_l}$
  return T



Runtime 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
- There are $O(n^2)$ edges, therefore the runtime is $O(n^2)$

Characterization III of DT(P)

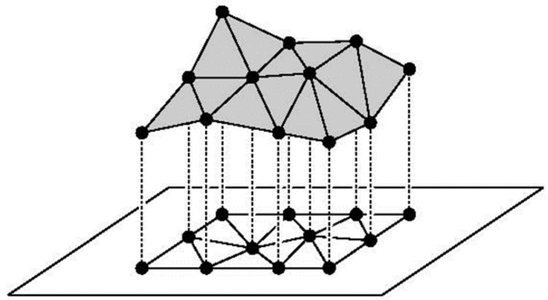
- **Definition:** Let T be a triangulation of P and let $\alpha_1, \alpha_2, \ldots, \alpha_{3m}$ be the angles of the m triangles in T sorted by increasing value. Then $A(T)=(\alpha_1,\alpha_2,\ldots,\alpha_{3m})$ is called the angle vector of T.
- **Definition:** A triangulation T is called **angle optimal** $\Leftrightarrow A(T) > A(T')$ for any other triangulation of the same point set P.

- Let T' be a triangulation that contains an illegal edge, and let T'' be the resulting triangulation after flipping this edge. Then $A(T'') > A(T')$.
- T is angle optimal $\Rightarrow$ T is legal $\Rightarrow$ T=DT(P)

- **Characterization III**: 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 DT



- Terrain modeling:
  - Model a scanned terrain surface by interpolating the height using a piecewise linear function over $\mathbb{R}^2$.
  - Angle-optimal triangulations give better approximations / interpolations since they avoid skinny triangles