TRIANGULATIONS

Hanan Samet

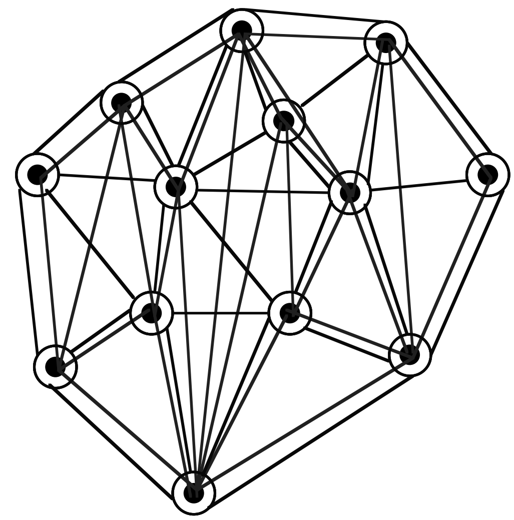
Computer Science Department and
Center for Automation Research and
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742
e-mail: hjs@umiacs.umd.edu

Copyright © 1997 Hanan Samet

These notes may not be reproduced by any means (mechanical or electronic or any other) without the express written permission of Hanan Samet

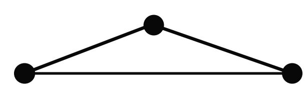
TRIANGULATION

1. Find a complete set of minimal clusters of locations in a scattered data set
2. Sample triangulation

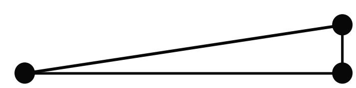


   - bad triangulation
   - good triangulation

3. Thin triangles are often undesirable (e.g., terrain modeling)
   - involve one or two very small angles which result from connecting vertices using a distance that is much longer than that which can be obtained through the use of more equiangular angles
     a. two very small interior angles, or



     b. one very small angle and two angles close to 90°

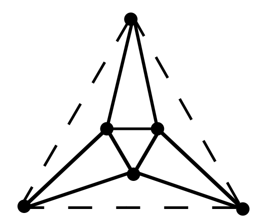


   - elevation values at the vertices of the triangles are used to interpolate (e.g., a weighted average) a surface at a point *p* in the plane
     a. use the vertices of the triangle containing *p*
     b. the longer and thinner the triangles, the further away are the vertices of the triangles serving as the basis of the triangulation and the higher the likelihood of error

Copyright © 1998 by Hanan Samet

TRIANGULATION POINT DATA SETS

- Types
  1. arbitrary point set
     - edges of convex hull are in all of the triangulations since triangulations are maximal and thus these edges cannot be ruled out by any other edge
     - generally complex
  2. vertices of a polygon
     - defined to only triangulate the interior of the polygon
     - edges of the convex hull of the vertices are not necessarily in the triangulation as they may lie outside of the original polygon - e.g., a star shaped polygon



     - can triangulate vertex set of any simple (i.e., not self-intersecting) polygon with n vertices in $O(n)$ time (Chazelle)
       1. quality of triangulation is unknown
       2. too complicated to be implemented
- A data set with N data points has $N \cdot (N-1)/2$ possible edges
  1. triangulation is a maximal subset of these edges chosen so that no edge crosses any other edge
  2. if convex hull of the points has M vertices
     - number of edges = $3 \cdot N - M - 3$
     - number of faces (i.e., triangles) = $2 \cdot N - M - 2$
     - derive by induction

Copyright © 1998 by Hanan Samet

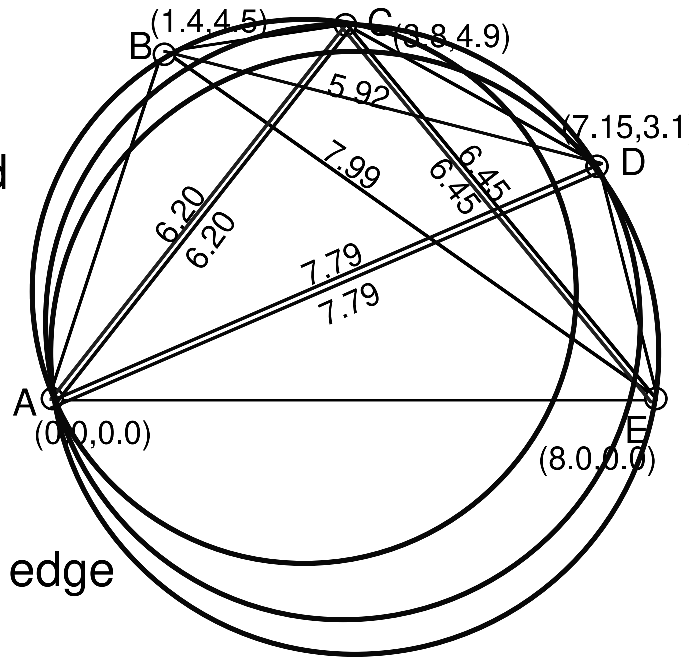
INTUITIVE TRIANGULATION PROCEDURES

- Start with an arbitrary triangulation
  1. finite set of points (triangle vertices)
  2. form a finite collection of line segments (triangle edges)
  3. form a collection of triangles (point or edge triples)
  4. record triangle adjacencies (edge-triangle triples)
     - can also be implicit
- Optimize by iterating with a local rule to select more equi–angular triangles
  1. select shorter of two intersecting quadrilateral diagonals
  2. reduce the minimal distance to a triangle vertex
  3. select triangle with minimum altitude
  4. select triangle with closest circumcenter
  5. select triangle with least rate of steepest ascent
- Disadvantages
  1. nonuniqueness
  2. elongated triangles
  3. need for several iterations
  4. preprocessing requirements

Copyright © 1998 by Hanan Samet

EXAMPLE POLYGON TRIANGULATIONS

They work for arbitrary point sets as well



Optimal triangulation (MWT)

- choose triangles so the total length of edges is minimized
- AC and CE
- takes $O(N^3)$ time for a polygon and much higher for more complex point sets

Greedy triangulation

- exclude an edge if a shorter edge intersects it
- process edges in order of increasing length
- choose BD first (it is the shortest) followed by AD because it is the shortest of the remaining nonintersecting edges
- $O(N^2 \log(N^2)) = O(N^2 \log N)$ time as need to sort edges

Delaunay triangulation

- no point lies inside the circumcircle of any other triangle
- choose edges AC and AD because only triangles ABC, ACD, and ADE have empty circumcircles
- $O(N \log N)$ time

Plane-sweep triangulation

- sort vertices in increasing order of x coordinate values
- sweep an infinite line from left to right and draw edges from each encountered vertex to all visible vertices to its left that do not intersect an existing edge
- $O(N \log N)$ time as need to sort vertices, sweep ($O(N)$), and compute visibility at each vertex ($O(\log N)$)

Copyright © 1998 by Hanan Samet

USE OF TRIANGULATIONS IN ELEVATION MODELS

- Assume that elevations of a subset of points in the plane are known
- Use interpolation on triangle faces to determine the elevation of a point that is not a triangle vertex

Ex: given point $q = (q_x, q_y)$, determine q_z

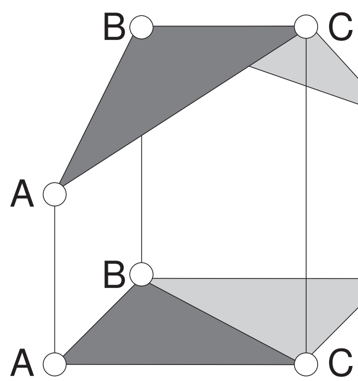
1. locate triangle t (t_1, t_2, t_3) containing q
   - Kirkpatrick's K-structure
   - Edelsbrunner, Guibas, and Stolfi's layered dag
   - store triangulation in a PMR quadtree
     a. locate e—the nearest edge to q
     b. starting at e, walk around q to extract the enclosing triangle
2. $q = a_1 \cdot t_1 + a_2 \cdot t_2 + a_3 \cdot t_3$
   - $a_1 + a_2 + a_3 = 1$ and a_1, a_2, a_3 are non-negative
   - unique $a_1 + a_2 + a_3 = 1$ and a_1, a_2, a_3 are known as *vertex* (or *barycentric*) coordinates
   - we know q_x and q_y
     a. $q_x = a_1 \cdot t_{1x} + a_2 \cdot t_{2x} + a_3 \cdot t_{3x}$
     b. $q_y = a_1 \cdot t_{1y} + a_2 \cdot t_{2y} + a_3 \cdot t_{3y}$
3. solve for a_i and substitute into $q_z = a_1 \cdot t_{1z} + a_2 \cdot t_{2z} + a_3 \cdot t_{3z}$

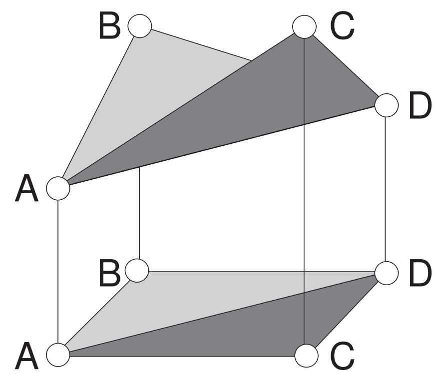
Copyright © 1998 by Hanan Samet

AMBIGUITY IN CHOOSING TRIANGULATIONS

- Problems in using triangulations for modeling surfaces

  1. any four non-coplanar measurements can be viewed as vertices of a tetrahedron in three-dimensional space
  2. projection of the tetrahedron in two-dimensional space yields a nonplanar triangulation
     - only retain one of the diagonals of the quadrilateral
     - the retained diagonal determines whether the surface is convex or concave





Copyright © 1998 by Hanan Samet

OPTIMAL TRIANGULATION (MWT)

Def: minimal sum of edge lengths

- Assign triangles to yield the shortest possible total length of edges in a completed network
- Also known as minimum weight triangulation (MWT)
- Several distinct triangulations may have the same total length
- An angular criterion may yield a triangulation where the proportions of the triangles are better for interpolation
- Takes $O(N^3)$ time for a simple polygon (i.e., not self-intersecting) and much higher for more complex point sets
  1. there is an exponential number of possible triangulations for a convex polygon (a Catalan number)
  2. impractical to try them all and take the minimum
  3. use dynamic programming

Copyright © 1998 by Hanan Samet

MINIMUM WEIGHT POLYGON TRIANGULATION ALGORITHM

- For each of $O(N^2)$ possible edges, assume it is in the MWT triangulation and determine the two vertices out of $O(N)$ which form the two triangles of the MWT in which the edge participates
- Algorithm:
  1. given polygon Q with vertices $P_1\ P_2 \dots P_N P_1$ in order about Q
  2. let E_j denote an edge from P_i to P_j
  3. let T_{ij} be the MWT of the polygon with vertices $P_i\ P_{i+1} \dots P_j\ P_i$ and let W_{ij} be its total edge length
     - if edge E_{ij} is not a legal edge of a triangulation of Q (i.e., some part of it lies outside Q), then T_{ij} is undefined and $W_{ij} = \infty$
     - if edge E_{ij} is a legal edge, then there exists a vertex P_k in T_{ij} such that $P_i\ P_j\ P_k$ form a triangle
       a. P_k is found by solving $\min\{W_{ik} + W_{kj} \mid i < k < j\}$, AND
       b. $W_{ij} = W_{ik} + W_{kj} + \text{length}\ (P_{ij})$
  5. compute all T_{ij} and all W_{ij} in $O(N^3)$ time using dynamic programming
  6. MWT of Q is T_{1N}

Copyright © 1998 by Hanan Samet

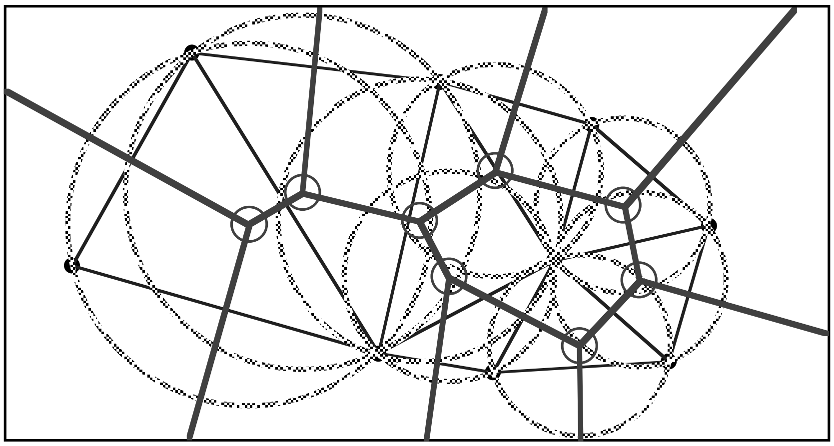
GREEDY TRIANGULATION

- Construct by adding the edges in increasing order of length, excluding a new edge if it intersects an existing edge
- Takes $O(N^2 \log(N^2)) = O(N^2 \log N)$ time since have to first sort the edges according to length, and there are $O(N^2)$ possible edges
- Checking if an edge intersects an existing set of at most N edges (because there are only $O(N)$ edges in a triangulation)
- can be done in $O(\log N)$ time
- Not unique if two properly intersecting edges have the same length
- Not used much since not even a good approximation to the optimal triangulation (MWT)
  1. edges are chosen to optimize a sequential condition
  2. choice of edges is somewhat local (i.e., do they intersect some previously selected edge?, which must be smaller by nature of the ordering)
  3. in contrast, optimal triangulation (MWT) is optimized on a global condition

Copyright © 1998 by Hanan Samet

DELAUNAY TRIANGULATION (DT)

Circle Property: no data point lies inside the circumcircle of any other triangle



Delaunay triangles

circumcircles

circumcenters

Voronoi diagram

- Unique as long as no local configuration of four or more points is cocircular
- Maximizes the minimum interior vertex angle
  1. biased against large circumcircles
  2. biased against “thin” triangles
- Effectively chooses edges connecting nearest natural neighbors
  1. i.e., data points at the vertices are closer to their mutual circumcenter than is any other data point
  2. a form of local optimization
- Triangles are as close as possible to being equiangular

Def: Voronoi diagram (also a Thiessen or Dirichlet tessellation)

- partition of the plane into regions so that each point in a region is closer to its region’s *site* (a vertex of the Delaunay triangulation) than to any other site
- geometric dual of DT with the circumcenters as its vertices

Copyright © 1998 by Hanan Samet

OUTWARD-GROWING DELAUNAY TRIANGULATION CONSTRUCTION (Watson)

- works for arbitrary dimensions
- Construct a triangle enclosing entire point set (vertices of the triangle need not be in the point set)
- Process data points in arbitrary order, say P (initially there is just one point)
  1. find all existing triangles whose circumcircles include P
     - can find them by walking along the edges of the existing triangle that contains P
       a. if triangle ABC along edge AB contains P, then recursively examine the triangles along edges AC and BC, …
       b. otherwise, continue with edge adjacent to AB
     - takes $O(N)$ time since $O(N)$ triangles
  2. remove these triangles and form a polygon from their union without the interior edges
     - new polygon is star-shaped since for each edge AB and triangle ABC that are removed, P lies in the region bounded by AB and the circumcircle of ABC, and thus P can “see” edges AC and BC
  3. form new triangles by connecting P with the polygon’s edges
     - triangles exist since polygon is star-shaped
     - star-shaped region guarantees that new triangles don't contain other points
     - in the worst case, $O(N)$ time since no more than $O(N)$ triangles
- In the worst case, may have to examine $O(N)$ circum–circles at each of N steps—hence $O(N^2)$
- If points are selected in random order, then the expected time is $O(N \log_2 N)$ in two dimensions (Guibas/Knuth/Sharir)

Copyright © 1998 by Hanan Samet

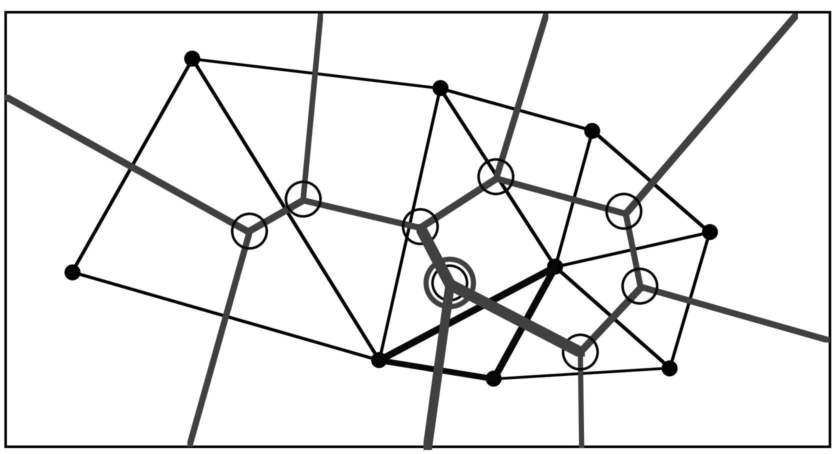
INWARD-GROWING DELAUNAY TRIANGULATION CONSTRUCTION

- Compute the convex hull—$O(N\log N)$ time
- Push all edges of convex hull into a queue with an orientation so that the rest of the point set is to the left of the edge
- While queue is nonempty do
  1. pop an edge AB
  2. find the point C to the left of AB so that the circle through A, B, and C is empty
     - guaranteed to exist by circle property
     - $O(N)$ time but could be $O(1)$ if use a grid
  3. if AC is not in the DT, insert it in the DT and push it on the queue
  4. if CB is not in the DT, insert it in the DT and push it on the queue
     - steps 3 and 4 require $O(N)$ time but could be $O(\log N)$ by assigning each edge a unique encoding and storing the edges in a balanced binary tree
- Total of $O(N^2)$ time since N iterations and each takes $O(N)$ time

Copyright © 1998 by Hanan Samet

VORONOI DIAGRAM CONSTRUCTION

- Construct perpendicular bisectors of edges of Delaunay triangulation
- Bisectors meet in triplets at the circumcenters of the triangles
- Region surrounding a data point and bounded by the bisectors is a Voronoi region and is closer to that point than to any other point in the set
- Voronoi regions are not closed when the data point lies on the boundary of the set
- Notice that the bisectors of the sides of a triangle *T* do not necessarily intersect at a point in *T*



Copyright © 1998 by Hanan Samet

PROPERTIES OF DELAUNAY TRIANGULATIONS

1. Circle Property: No data point lies inside the circumcircle of any other triangle
   - fully determined by the circle property
   - always exists
2. Unique if no subset of four points are co-circular
   - implies non-unique for gridded data as 4 adjacent grid points are always co-circular
   - always unique subject to swapping the diagonals within the polygons whose vertices are co-circular
3. Disk Property: if no four points are co-circular, then an edge belongs to the unique Delaunay triangulation iff there exists a disk containing both endpoints and no other data points
4. Invariant under any transformation of the point set that preserves circles and circle containment
   - rigid motions, scalings, reflections
   - all combinations (affine transformations)

Copyright © 1998 by Hanan Samet

5. DT is planar graph dual of Voronoi diagram (VD)

   - can convert from DT to VD in $O(N)$ time

     a. compute the circumcenter of each triangle and that of its three adjacent triangles (takes $O(1)$ time)

     b. connect the circumcenters

   - can convert from VD to DT in $O(N)$ time

     a. connect two sites if their Voronoi regions share an edge

     b. easy since each edge usually indicates which regions are adjacent to it

6. Maximizes the minimum interior vertex angle

   - avoids small angles which cause distortion

   - does not minimize maximum angle but does limit its size by virtue of the three angles summing up to 180

Copyright © 1998 by Hanan Samet

7. Produces the lexicographically largest nondecreasing sequence of angles possible in any triangulation—i.e.,
   - if $a_1 \le a_2 \le \ldots \le a_i \le \ldots$
   - and if another triangulation has $b_1 \le b_2 \le \ldots \le b_i \le \ldots$
   - then there exists an i such that $a_i > b_i$
8. Can update locally with addition or deletion of vertices
   - local update only affects triangles whose circumcircles contain the new point
   - search and update in $O(\log N)$ time with appropriate data structures
9. $O(N \log N)$ building time as a result of need to sort
   - use divide and conquer (not easy!)
10. Delaunay triangles are not hierarchical
    - can't be aggregated to form larger triangles
    - subdividing into smaller triangles leads to "elongated" rather than "fat" triangles

Copyright © 1998 by Hanan Samet