Voronoi Diagrams & Delaunay Triangulations

O’Rourke: Chapter 5

de Berg et al.: Chapters 7, 9 (and a touch of Ch. 8)

Wednesday, 9/26/12
Applications Preview

- Nearest neighbors
- Minimum spanning tree
- Medial axis

Common 2D Computational Geometry Structures

- Convex Hull
- Voronoi Diagram
- New Point
- Delaunay Triangulation

Triangle whose circumcircle’s center is just outside triangle

Combinatorial size in $O(n)$
Basics

- Voronoi region $V(p_i)$ is set of all points at least as close to $p_i$ as to any other site:

$$V(p_i) = \{ x : |p_i - x| \leq |p_j - x| \quad \forall j \neq i \}$$

$$V(p_i) = \bigcap_{i \neq j} H(p_i, p_j)$$

Note: This definition is independent of dimension.

source: O’Rourke
Digression on Point/Line Duality

- Point in 2D plane has coordinates $x$ and $y$
- (Non-vertical) line has slope and $y$-intercept
- One-to-one mapping of set of points to set of lines (and vice versa) is a type of duality mapping
- Example of one duality:
  - dual of point $p$ (denoted $p^*$) is line $(y = p_x x - p_y)$
  - dual of line $l$ $(y = mx + b)$ is point $l^* = (m, -b)$

This duality preserves:
- Incidence
- Order (above/below)

source: de Berg et al. Ch 8
Delaunay Triangulations: Properties

Voronoi Diagram / Delaunay Triangulation

- **The mouse**: Click the mouse in the drawing region to add new sites to the Voronoi diagram or Delaunay triangulation.
- **The Voronoi Diagram and Delaunay Triangulation checkboxes**: These toggle between the Voronoi diagram and the Delaunay triangulation. Your current set of sites remains the same for both diagrams.
- **The Clear button**: Press this to begin a new diagram with no sites.
- **The Show... regions**: If you move the mouse over one of these regions then the corresponding thing will be shown on the screen. For instance, if you move the mouse over Show Empty Circles then the empty circles for the current diagram will be shown.

2D Delaunay Triangulations: Properties

- D1. \( D(P) \) is the straight-line dual of \( V(P) \) [by definition]
- D2. \( D(P) \) is a triangulation if no 4 points of \( P \) are cocircular: Every face is a triangle.
- D3. Each face of \( D(P) \) corresponds to a vertex of \( V(P) \)
- D4. Each edge of \( D(P) \) corresponds to an edge of \( V(P) \)
- D5. Each node of \( D(P) \) corresponds to a region of \( V(P) \)
- D6. The boundary of \( D(P) \) is the convex hull of the sites
- D7. The interior of each face of \( D(P) \) contains no sites

D3-D5 define a *duality* mapping between features of the Delaunay Triangulation and Voronoi Diagram.

Source: O'Rourke
2D Delaunay Triangulations:
Properties of Voronoi Diagrams

Voronoi Diagram / Delaunay Triangulation

- **The mouse**: Click the mouse in the drawing region to add new sites to the Voronoi diagram or Delaunay triangulation.
- **The Voronoi Diagram and Delaunay Triangulation checkboxes**: These toggle between the Voronoi diagram and the Delaunay triangulation. Your current set of sites remains the same for both diagrams.

http://www.cs.cornell.edu/Info/People/chew/Delaunay.html
2DVoronoi Diagram: Properties

- V1. Each Voronoi region $V(p_i)$ is convex
- V2. $V(p_i)$ is unbounded iff $p_i$ is on convex hull of point set
- V3. If $v$ is a Voronoi vertex at junction of $V(p_1)$, $V(p_2)$, $V(p_3)$, then $v$ is center of circle $C(v)$ determined by $p_1$, $p_2$, and $p_3$.
- V4. $C(v)$ is circumcircle for Delaunay triangle for $v$
- V5. Interior($C(v)$) contains no sites (proof by contradiction)
- V6. If $p_i$ is a nearest neighbor to $p_j$, then $(p_i, p_j)$ is an edge of $D(P)$
- V7. If there is some circle through $p_i$ and $p_j$ that contains no other sites, then $(p_i, p_j)$ is an edge of $D(P)$. [reverse holds too]

source: O’Rourke
Back to Delaunay Triangulation: Another Property (in 2D)

- Let $P$ be a set of points in the plane (in general position).
- Let number of triangles in triangulation $T$ be $m$.
- **Angle-Vector:** Sorted angle list: $A(T) = (\alpha_1, \alpha_2, \ldots, \alpha_m)$

\[ A(T) > A(T') \] if $\exists \ 1 \leq i \leq 3m$ such that $\alpha_j = \alpha'_j \ \forall j < i$ and $\alpha_i > \alpha'_i$

- **Angle-Optimal Triangulation $T$:** $A(T) \geq A(T')$ for all triangulations $T'$ of $P$

- **Theorem 9.9:** Any angle-optimal triangulation of $P$ is a Delaunay triangulation of $P$. Furthermore, any Delaunay triangulation of $P$ maximizes the minimum angle over all triangulations of $P$.

Try to avoid skinny triangles!

source: de Berg et al. Ch 8
• Delaunay Triangulation:
  • Maximizes the minimum angle (as on previous slide)
  • Minimizes maximum circumcircle radius
  • **Minimizes maximum radius of an enclosing circle**
  • Maximizes sum of inscribed circle radii
  • And more…

*Warning: Most of these 2D optimality properties do not extend to higher dimensions! (exception: minimize maximum radius of simplex-enclosing sphere)*

source: *Handbook of Discrete & Computational Geometry*
Voronoi Diagram Algorithms

One strategy: Find Delaunay Triangulation then use duality (as in CGAL).

- **Half-Plane Intersection:**
  - Construct each Voronoi region separately:
    - Intersect n-1 halfplanes
      - Intersecting n halfplanes in 2D is dual* to constructing convex hull of n points
    - O(n lg n) time
    - Total time: O(n² lg n)
  - Incremental Construction:
    - Add one point at a time
    - If new point is inside some Voronoi circle, locally update diagram
    - O(n) work per point
    - Total time: O(n²)
- **Divide-and-Conquer:** O(n lg n)
  - Optimal but complex to implement
- **Lift points onto 3D paraboloid:**
- **Parabolic Sweep:** O(n lg n)
  - Brief summary on next slide; detailed treatment in Ch. 7 of de Berg et al.

*see other slides
Voronoi Diagram Algorithms: Fortune’s Parabolic Sweep

- Plane-sweep complication: What if encounter Voronoi edges before corresponding site?
- Solution:
  - Expanding circles interpretation (z is time dimension)
  - 2 cones intersect in branch of hyperbola that projects down onto (potential) Voronoi edge
  - If cones opaque, then Voronoi diagram is view from \( z = -\infty \)
  - Sweep-line is intersection of slanted plane \( \pi \) with xy-plane
  - Because \( \pi \) slanted at same angle as cones, \( L \) hits site when \( \pi \) first hits cone for site
  - Invariant: Everything visible underneath \( \pi \) is always finished (that part of Voronoi diagram is done)
  - Maintain parabolic front = projection of intersection of \( \pi \) with cones onto xy-plane
  - Events (see next slide):
    - Site event: new (degenerate, single-line) parabola forms. 2 (duplicate) intersection points with parabolic front form new degenerate Voronoi diagram edge
    - Circle event: 2 parabolic front intersection points meet to form Voronoi vertex.
  - As parabolic front evolves, parabola intersection points trace out Voronoi diagram.
- Site event: new (degenerate, single-line) parabola forms. 2 (duplicate) intersection points with parabolic front form new degenerate Voronoi diagram edge
- Circle event: 2 parabolic front intersection points meet to form Voronoi vertex.
Delaunay Triangulation Algorithms

One strategy: Find Voronoi Diagram then use duality.

- Lift points onto 3D paraboloid *
  - CGAL uses lifting for d-dimensional Delaunay
- Start with any triangulation & flip edges
  - Described in deBerg et al. *
- Randomized incremental with edge flipping
  - Described in deBerg et al. *
  - Used by CGAL for 2D and 3D (tetrahedra in 3D)
- Constrained Delaunay triangulation
  - User specifies line segments that *must* be in the triangulation.
  - See Shewchuck paper for an algorithm description.
  - Available in CGAL for 2D only.
Application: Mesh Generation for Finite Element Modeling & Visualization [Research Note for 17th Int. Meshing Roundtable, 2008; also presented at Fall CG Workshop; 10th Int. Symp. on Experimental Algorithms, 2011]

- Needed for signal integrity in printed circuit board interconnect routing
- 2D constrained Delaunay triangulation is extruded into 3D to form triangular prism mesh
- “Angle optimality” of Delaunay triangulation is useful for this type of modeling.

*Courtesy of Cadence Design Systems*

*Doctoral student S. Ye*
Delaunay Triangulation Algorithms

Start with any triangulation & flip edges

**Algorithm** \textsc{LegalTriangulation}(\mathcal{T})

*Input.* Some triangulation \(\mathcal{T}\) of a point set \(P\).

*Output.* A legal triangulation of \(P\).

1. \textbf{while} \(\mathcal{T}\) contains an illegal edge \(\overline{p_ip_j}\)
2. \quad \textbf{do} (* Flip \(\overline{p_ip_j}\) *)
3. \quad \quad Let \(p_ip_jp_k\) and \(p_ip_jp_l\) be the two triangles adjacent to \(\overline{p_ip_j}\).
4. \quad \quad Remove \(\overline{p_ip_j}\) from \(\mathcal{T}\), and add \(\overline{p_kp_l}\) instead.
5. \quad \textbf{return} \(\mathcal{T}\)

Termination is guaranteed due to finite # different triangulations and increasing triangle angles, but this version of the algorithm is slow...

source: deBerg et al.
Randomized incremental with edge flipping:

**Algorithm** DELAUNAYTRIANGULATION($P$)

*Input.* A set $P$ of $n + 1$ points in the plane.

*Output.* A Delaunay triangulation of $P$.

1. Let $p_0$ be the lexicographically highest point of $P$, that is, the rightmost among the points with largest y-coordinate.
2. Let $p_{-1}$ and $p_{-2}$ be two points in $\mathbb{R}^2$ sufficiently far away and such that $P$ is contained in the triangle $p_0p_{-1}p_{-2}$.
3. Initialize $T$ as the triangulation consisting of the single triangle $p_0p_{-1}p_{-2}$.
4. Compute a random permutation $p_1, p_2, \ldots, p_n$ of $P \setminus \{p_0\}$.
5. for $r \leftarrow 1$ to $n$
6.     do (/* Insert $p_r$ into $T$; */
7.             Find a triangle $p_ip_jp_k \in T$ containing $p_r$.
8.             if $p_r$ lies in the interior of the triangle $p_ip_jp_k$
9.                 then Add edges from $p_r$ to the three vertices of $p_ip_jp_k$, thereby splitting $p_ip_jp_k$
10.                    into three triangles.
11.             *LEGALIZEEDGE($p_r, p_ip_j, T$)
12.             *LEGALIZEEDGE($p_r, p_jp_k, T$)
13.             *LEGALIZEEDGE($p_r, p_kp_i, T$)
14.             else (/* $p_r$ lies on an edge of $p_ip_jp_k$, say the edge $p_ip_j$ */)
15.                 Add edges from $p_r$ to $p_k$ and to the third vertex $p_l$ of the other triangle that
16.                     is incident to $p_ip_j$, thereby splitting the two triangles incident to $p_ip_j$
17.                     into four triangles.
18.             *LEGALIZEEDGE($p_r, p_ip_l, T$)
19.             *LEGALIZEEDGE($p_r, p_ip_l, T$)
20.             *LEGALIZEEDGE($p_r, p_jp_l, T$)
21.             *LEGALIZEEDGE($p_r, p_kp_l, T$)
22. Discard $p_{-1}$ and $p_{-2}$ with all their incident edges from $T$.
23. return $T$

*Expected running time is in $O(n \log n)$. See future slide.*
Cases for randomized incremental with edge flipping:

- **$p_r$ lies in the interior of a triangle**
  - Produces 3 calls to `LEGALIZEEDGE`

- **$p_r$ falls on an edge**
  - Produces 4 calls to `LEGALIZEEDGE`

*Source: deBerg et al.*
**Delaunay Triangulation Algorithms**

**Edge flipping:**

```
LEGALIZEEDGE(p_r, p_i p_j, T)
1. (∗ The point being inserted is p_r, and p_i p_j is the edge of T that may need to be flipped. ∗)
2. if p_i p_j is illegal
3. then Let p_i p_j p_k be the triangle adjacent to p_r p_i p_j along p_i p_j.
4. (∗ Flip p_i p_j: ∗) Replace p_i p_j with p_r p_k.
5. LEGALIZEEDGE(p_r, p_i p_k, T)
6. LEGALIZEEDGE(p_r, p_k p_j, T)
```

*Existing edges may become illegal*

*Termination guaranteed since each flip makes angle-vector of triangulation larger.*

*source: deBerg et al.*
Delaunay Triangulation Algorithms

- **Edge Flipping:**
  - Key contributor to $O(n \log n)$ expected running time is fast location of triangle containing point $p_r$: DAG point location structure
  - Making a leaf into an internal node yields at most 3 outgoing edges.
  - **Lemma 9.11:** The expected number of triangles created by algorithm DELAUNAYTRIANGULATION is at most $9n+1$.
    - Key idea: Bound the expected degree of point $p_r$.
  - Use Lemma 9.11 together with ideas from randomized incremental framework
    - Recall “conflict graph” used for randomized incremental convex hull construction.
    - In Delaunay case, conflict arises when a point is in circumcircle of a triangle.
How are Convex Hull, Delaunay Triangulation and Voronoi Diagram related?

Delaunay Triangulation

- Project each point upwards onto paraboloid \( z = x^2 + y^2 \)
- Construct 3D Convex Hull
- Discard “top” faces
- Project Convex Hull down to \( xy \) plane to form \textit{Delaunay Triangulation}
- View from \( z = -\infty \) to see \textit{Delaunay Triangulation}

source: O’Rourke
How are Convex Hull, Delaunay Triangulation and Voronoi Diagram related?

Voronoi Diagram

- Project each point upwards onto paraboloid $z = x^2 + y^2$
- At each projected point, construct plane tangent to paraboloid
  - tangent above $(a,b)$ is
    - $z = 2ax + 2by - (a^2 + b^2)$
- View from $z = +\infty$ to see Voronoi Diagram
  - “first intersection” of planes

source: O’Rourke
Voronoi Diagram Applications: Nearest Neighbors

Nearest Neighborhood query for point $q$ with respect to points $P = p_1, p_2, ..., p_n$ can be answered in $O(\log n)$ time using Voronoi diagram.

All Nearest Neighbors:

The Nearest Neighborhood Graph (NNG) of points $P$ is a graph (see CGAL example):
- whose nodes correspond to the points
- an arc from $p_i$ to $p_j$ iff $p_j$ is a nearest neighbor of $p_i$:

$$|p_i - p_j| \leq \min_{k \neq i} \{|p_i - p_k|\}$$

$NNG \subseteq D(P)$ So NNG can be constructed in $O(n\log n)$ time.

source: O’Rourke
Problem: Find a largest empty circle whose center is in the (closed) convex hull of a set of \( n \) sites \( S \) (empty in that it contains no sites in its interior, and largest in that there is no other such circle with strictly larger radius).

*Finite* set of potential largest empty circle centers:
- Voronoi vertices
- Intersections of Voronoi edges and hull of sites.
- *Illustrates Computational Geometry common theme:*
  - Reduce infinite candidate set to small finite list, then find these efficiently.

source: O’Rourke
Finite set of potential largest empty circle centers:
- Voronoi vertices: center strictly interior to hull of sites
- Intersections of Voronoi edges and hull of sites: center on hull

Circle can grow if only stopped by 1 or 2 sites. Circle stopped by 1 site allows movement along h. Center may follow Voronoi edge if circle only stopped by 2 sites.
MST can be used in TSP approximation algorithm.
More Applications: Euclidean Minimum Spanning Tree

\[ \text{MST} \subseteq D(P) \]

**Approach:** Let \( ab \in \text{MST} \) and suppose \( ab \notin D(P) \). Derive contradiction by showing supposed MST not minimal. w.l.o.g. \( c \in T_a \). Replace \( ab \) with \( cb \) to shorten length.

**Figure 5.16** \( T_a + bc + T_b \) is shorter than \( T_a + ab + T_b \).
More Applications: 
Relative Neighborhood Graph

The Relative Neighborhood Graph (RNG) of points \( p_1, p_2, \ldots, p_n \) is a graph:
- whose nodes correspond to the points
- 2 nodes \( p_i, p_2 \) share an arc iff they are at least as close to each other as to any other point:

\[
|p_i - p_j| \leq \max_{m \neq i, j} \{ |p_i - p_m|, |p_j - p_m| \}
\]

Note: this produces a set of constraints (1 for each \( m \))

It can be shown (O’Rourke exercise) that:
- every edge of the RNG is an edge of the Delaunay triangulation \( \text{RNG} \subseteq D(P) \)
- every edge of an MST is an edge of the RNG \( \text{MST} \subseteq \text{RNG} \)
More Applications: Medial Axis

- **Medial Axis:** of polygon $P = \text{set of points inside } P \text{ that have } > 1 \text{ closest point among } P\text{’s boundary points.}$
  - Tree whose leaves are vertices of $P$
  - Every point of medial axis is center of circle touching boundary in at least 2 points.
  - Vertices of medial axis are centers of circles touching 3 distinct boundary points.
  - “Grassfire” analogy: fire starts at boundary
    - Medial axis represents “quench points” where fires meet each other
- This is also the Voronoi diagram of the polygon.

*Medial axis of $n$-vertex polygon can be constructed in $O(n \log n)$ time.*

sources O’Rourke and de Berg et al.
More Delaunay Applications: Shewchuck Triangulation

Shewchuck

Triangulation

http://www.cs.cmu.edu/~quake/triangle.demo.html

http://www.cs.cmu.edu/~quake/triangle.demo.html