Triangulation and Voronoi Regions

CS535

Daniel G. Aliaga
Department of Computer Science
Purdue University

[some slides based on Profs. Shmuel Wimer and Andy Mirzaian]
Lemma: Every polygon must have at least one strictly convex vertex.

Proof: Let the vertices be counterclockwise ordered. Traversing the boundary, a convex vertex corresponds to a left turn.

Pick the lowest vertex (pick the rightmost if there are a few). \( L \) is a line passing through \( v \). The edge following \( v \) must lie above \( L \). ■
Lemma: Every polygon of \( n \geq 4 \) vertices has a diagonal.

Proof: There exists a strictly convex vertex \( v \). Let \( a \) and \( b \) be vertices adjacent to \( v \). If \([a,b]\) is a diagonal we are done. Else...
\( \Delta avb \) must contain at least one vertex of \( P \).

Let \( x \) be the closest vertex to \( v \), measured orthogonal to the line passing through \( ab \).

The interior of \( \Delta cvd \) cannot contain any point of \( \partial P \).

Therefore \([x, v] \cap P = \{x, v\}\), hence a diagonal. ■
Theorem: Every $n$-vertex polygon $P$ can be triangulated.

Proof: By induction on $n$.

If $n=3$ $P$ is a triangle.

Let $n\geq 4$. By lemma, $P$ has a diagonal $d$ which divides $P$ into two polygons $P_1$ and $P_2$, having $n_1 < n$ and $n_2 < n$ vertices, respectively. $P_1$ and $P_2$ can be triangulated by induction hypothesis. ■
Interesting Question: Do all triangulations of a given polygon have the same number of diagonals and triangles?

Answer: Every triangulation of an $n$-vertex polygon $P$ has $n-3$ diagonals and $n-2$ triangles.
Triangulation Complexity

Implementing triangulation as in the existence theorem requires $O(n^4)$ time.

There are $n(n-3)/2$ diagonal candidates.

Checking validity of a diagonal requires intersection test against all edges and previously defined diagonals, which takes $O(n)$ time.

This is repeated $n-3$ times, yielding total $O(n^4)$ time.
Triangulation Complexity

• Naïve: $O(n^4)$
  – There are $n(n-3)/2$ diagonal candidates
  – Checking the validity of a diagonal against all previous edges and diagonals takes $O(n)$
  – This is repeated $n-3$ times
  – Total time $O(n^4)$
Triangulation Complexity

• Lennes, 1911: $O(n^2)$
  – Pick the leftmost vertex $v$ of $P$ and connect its two neighbors $u$ and $w$. Checking whether $uw$ is a diagonal takes $O(n)$. If it is, the rest is a $(n-1)$-vertex polygon.
  – If $uw$ is not a diagonal, get $x$, the farthest vertex from $uw$ inside $\Delta uvw$. This takes $O(n)$ time. $vx$ is a diagonal dividing $P$ into $P_1$ and $P_2$, having $n$ total number of vertices.
  – Recursive application of the above procedure consumes total $O(n^2)$ time.
**Triangulation Complexity**

**Definition:** $P$ is monotone w.r.t to a line $l$ if $P$ intersects with any line $l'$ perpendicular to $l$ in a single segment, a point or it doesn’t intersect.

---

The diagram illustrates the concept of monotonicity and non-monotonicity with respect to a line $l$. The left side shows an example of a monotone polygon, while the right side shows a non-monotone polygon.
Triangulation Complexity

• An n-vertex simple polygon can be partitioned into y-monotone polygons in $O(n \log n)$ time and $O(n)$ storage

• Monotone polygon can be triangulated in $O(n)$
Triangulating y-Monotone Polygon (Sketch)

$P$'s vertices are sorted in descending $y$.

$P$'s boundary is traversed with one leg on \textbf{left} and one leg on \textbf{right}, by popping and pushing vertices from a stack.

Diagonals insertions are decided according to stack's top status.
Triangulation Complexity

• Theorem: (Gary et. al. 1978) A simple n-vertex polygon can be triangulated in $O(n \log n)$ time and $O(n)$ storage.

• The problem has been studied extensively between 1978 and 1991, when in 1991 Chazelle presented an $O(n)$ time complexity algorithm.
P = \{ p_1, p_2, \ldots, p_n \} a set of n points in the plane.
Voronoi Diagram:

Voronoi(P):  # regions = n, # edges ≤ 3n-6, # vertices ≤ 2n-5.
Delaunay Triangulation =
Dual of the Voronoi Diagram

DT(P):
- # vertices = n,
- # edges ≤ 3n-6,
- # triangles ≤ 2n-5.
Delaunay triangles have the “empty circle” property.
Voronoi Diagram & Delaunay Triangulation
\( P = \{ p_1, p_2, \ldots, p_n \} \) a set of \( n \) points in the plane.

Assume: no 3 points collinear, no 4 points cocircular.

\( H(p_i, p_j) \) half-plane

\( PB(p_i, p_j) \) perpendicular bisector of \( p_i p_j \).

Voronoi Region of \( p_i \):

\[
V(p_i) = \bigcap_{\substack{j=1 \atop j \neq i}}^n H(p_i, p_j)
\]

Voronoi Diagram of \( P \):

\[
VD(P) = \bigcup_{i=1}^n \{ V(p_i) \}
\]
Each Voronoi region $V(p_i)$ is a convex polygon (possibly unbounded).

$V(p_i)$ is unbounded $\iff$ $p_i$ is on the boundary of $\text{CH}(P)$.

Consider a Voronoi vertex $v = V(p_i) \cap V(p_j) \cap V(p_k)$.
Let $C(v)$ = the circle centered at $v$ passing through $p_i$, $p_j$, $p_k$.

$C(v)$ is circumcircle of Delaunay Triangle $(p_i, p_j, p_k)$.

$C(v)$ is an empty circle, i.e., its interior contains no other sites of $P$.

$p_j$ = a nearest neighbor of $p_i$ $\Rightarrow$ $V(p_i) \cap V(p_j)$ is a Voronoi edge
$\Rightarrow$ $(p_i, p_j)$ is a Delaunay edge.
DT Properties

- DT(P) is straight-line dual of VD(P).
- DT(P) is a triangulation of P, i.e., each bounded face is a triangle (if P is in general position).
- \((p_i, p_j)\) is a Delaunay edge \(\iff\ \exists\) an empty circle passing through \(p_i\) and \(p_j\).
- Each triangular face of DT(P) is dual of a Voronoi vertex of VD(P).
- Each edge of DT(P) corresponds to an edge of VD(P).
- Each node of DT(P), a site, corresponds to a Voronoi region of VD(P).
- Boundary of DT(P) is CH(P).
- Interior of each triangle in DT(P) is empty, i.e., contains no point of P.
Computing Delaunay Triangulation

• Many algorithms: $O(n \log n)$

• Lets use flipping:
  – Recall: A Delaunay Triangulation is a set of triangles $T$ in which each edge of $T$ possesses at least one empty circumcircle.
  – Empty: A circumcircle is said to be empty if it contains no nodes of the set $V$
What is a flip?

A non-Delaunay edge flipped
Flip Algorithm

• ??
Flip Algorithm

1. Let V be the set of input vertices.
2. T = Any Triangulation of V.
3. Repeat until all edges of T are Delaunay edges.
   a. Find a non-delaunay edge that is flippable
   b. Flip

Naïve Complexity: O(n^2)
Locally Delaunay $\rightarrow$ Globally Delaunay

• If $T$ is a triangulation with all its edges locally Delaunay, then $T$ is the Delaunay triangulation.

• Proof by contradiction:
  – Let all edges of $T$ be locally Delaunay but an edge of $T$ is not Delaunay, so flip it...
Flipping

• Other flipping ideas?
Randomized Incremental Flipping

• Complexity can be $O(n \log n)$
Popular Method

• Fortune’s Algorithm
The Wave Propagation View

Simultaneously drop pebbles on calm lake at n sites.

Watch the intersection of expanding waves.
Time as 3rd dimension

- All sites have identical opaque cones.
All sites have identical opaque cones.
cone(p) \cap cone(q) = vertical hyperbola h(p,q).
Vertical projection of h(p,q) on the xy base plane is PB(p,q).