Geometric Algorithms

Lecture 5: Delaunay Triangulations
Motivation: Terrains

- A terrain is the graph of a function $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$.
Motivation: Terrains

- a terrain is the graph of a function \( f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R} \)
- we know only height values for a set of measurement points
Motivation: Terrains

- A terrain is the graph of a function $f : A \subset \mathbb{R}^2 \to \mathbb{R}$.
- We know only height values for a set of measurement points.
- How can we interpolate the height at other points?
a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \to \mathbb{R}$

we know only height values for a set of measurement points

how can we interpolate the height at other points?

using a triangulation
Motivation: Terrains

- a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$
- we know only height values for a set of measurement points
- how can we interpolate the height at other points?
- using a triangulation – but which?

**Interpolated height**

- $q$ with interpolated height = 985
- $q$ with interpolated height = 23
Let $P = \{p_1, \ldots, p_n\}$ be a point set. A **triangulation** of $P$ is a maximal planar subdivision with vertex set $P$. 
Let $P = \{p_1, \ldots, p_n\}$ be a point set. A triangulation of $P$ is a maximal planar subdivision with vertex set $P$.

**Complexity:**

- $2n - 2 - k$ triangles
- $3n - 3 - k$ edges

**Euler’s formula** for connected plane graphs:

$\# \text{faces} - \# \text{edges} + \# \text{vertices} = 2$, also counting the outer face.
Let $\mathcal{T}$ be a triangulation of $P$ with $m$ triangles and $3m$ vertices. Its angle vector is $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ where $\alpha_1, \ldots, \alpha_{3m}$ are the angles of $\mathcal{T}$ sorted by increasing value.
Let $\mathcal{T}$ be a triangulation of $P$ with $m$ triangles and $3m$ vertices. Its angle vector is $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ where $\alpha_1, \ldots, \alpha_{3m}$ are the angles of $\mathcal{T}$ sorted by increasing value.

Let $\mathcal{T}'$ be another triangulation of $P$. We define $A(\mathcal{T}) > A(\mathcal{T}')$ if $A(\mathcal{T})$ is lexicographically larger than $A(\mathcal{T}')$. 

$A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_6)$
Let $\mathcal{T}$ be a triangulation of $P$ with $m$ triangles and $3m$ vertices. Its angle vector is $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ where $\alpha_1, \ldots, \alpha_{3m}$ are the angles of $\mathcal{T}$ sorted by increasing value.

Let $\mathcal{T}'$ be another triangulation of $P$. We define $A(\mathcal{T}) > A(\mathcal{T}')$ if $A(\mathcal{T})$ is lexicographically larger than $A(\mathcal{T}')$.

$\mathcal{T}$ is angle optimal if $A(\mathcal{T}) \geq A(\mathcal{T}')$ for all triangulations $\mathcal{T}'$ of $P$. 

\[ A(T) = (\alpha_1, \ldots, \alpha_6) \]
Edge Flipping

Geometric Algorithms  Lecture 5: Delaunay Triangulations
Edge Flipping

- Change in angle vector:
  \( \alpha_1, \ldots, \alpha_6 \) are replaced by \( \alpha'_1, \ldots, \alpha'_6 \).
Edge Flipping

- Change in angle vector: 
  $\alpha_1, \ldots, \alpha_6$ are replaced by $\alpha'_1, \ldots, \alpha'_6$.

- The edge $e = p_i p_j$ is illegal if $\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i$. 
Edge Flipping

- Change in angle vector: \(\alpha_1, \ldots, \alpha_6\) are replaced by \(\alpha'_1, \ldots, \alpha'_6\).
- The edge \(e = p_i p_j\) is illegal if \(\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i\).
- Flipping an illegal edge increases the angle vector.
How do we determine if an edge is illegal?
How do we determine if an edge is illegal?

**Lemma:** The edge $p_ip_j$ is illegal if and only if $p_l$ lies in the interior of the circle $C$. 
Thales Theorem

**Theorem:** Let $C$ be a circle, $\ell$ a line intersecting $C$ in points $a$ and $b$, and $p, q, r, s$ points lying on the same side of $\ell$. Suppose that $p, q$ lie on $C$, $r$ lies inside $C$, and $s$ lies outside $C$. Then

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

where $\angle abc$ denotes the smaller angle defined by three points $a, b, c$. 
**Theorem:** Let $C$ be a circle, $\ell$ a line intersecting $C$ in points $a$ and $b$, and $p, q, r, s$ points lying on the same side of $\ell$. Suppose that $p, q$ lie on $C$, $r$ lies inside $C$, and $s$ lies outside $C$. Then

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

where $\angle abc$ denotes the smaller angle defined by three points $a, b, c$. 
**Theorem:** Let $C$ be a circle, $\ell$ a line intersecting $C$ in points $a$ and $b$, and $p, q, r, s$ points lying on the same side of $\ell$. Suppose that $p, q$ lie on $C$, $r$ lies inside $C$, and $s$ lies outside $C$. Then

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

where $\angle abc$ denotes the smaller angle defined by three points $a, b, c$. 
Legal Triangulations

A **legal triangulation** is a triangulation that does not contain any illegal edge.

---

**Algorithm**

```
LegalTriangulation(T)
```

**Input.** A triangulation \( T \) of a point set \( P \).

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

1. while \( T \) contains an illegal edge
2. do (\( \star \) Flip \( p_i p_j \) \( \star \))
3. Let \( p_i p_j p_k \) and \( p_i p_j p_l \) be the two triangles adjacent to \( p_i p_j \).
4. Remove \( p_i p_j \) from \( T \), and add \( p_k p_l \) instead.
5. return \( T \)

**Question:** Why does this algorithm terminate?
A **legal triangulation** is a triangulation that does not contain any illegal edge.

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

*Input.* A triangulation $\mathcal{T}$ of a point set $P$.

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

1. **while** $\mathcal{T}$ contains an illegal edge $\overline{pipj}$
2. **do** (* Flip $\overline{pipj}$ *)
3. Let $\overline{pijp_k}$ and $\overline{pijp_l}$ be the two triangles adjacent to $\overline{pipj}$.
4. Remove $\overline{pipj}$ from $\mathcal{T}$, and add $\overline{pkpl}$ instead.
5. **return** $\mathcal{T}$
A **legal triangulation** is a triangulation that does not contain any illegal edge.

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

*Input.* A triangulation \mathcal{T} of a point set \textit{P}.

*Output.* A legal triangulation of \textit{P}.

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

**Question:** Why does this algorithm terminate?
Let $P$ be a set of $n$ sites (points) in the plane.
Let $P$ be a set of $n$ sites (points) in the plane.

The Voronoi cell $V(p)$ for a site $p \in P$ is the set of all points in the plane that have $p$ as nearest site.
Let $P$ be a set of $n$ sites (points) in the plane.

The Voronoi cell $\mathcal{V}(p)$ for a site $p \in P$ is the set of all points in the plane that have $p$ as nearest site.

The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$. 
Let $P$ be a set of $n$ sites (points) in the plane.

The Voronoi cell $V(p)$ for a site $p \in P$ is the set of all points in the plane that have $p$ as nearest site.

The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $V(p)$ for all $p \in P$.

Let $G$ be the dual graph of $\text{Vor}(P)$. 
Let $P$ be a set of $n$ sites (points) in the plane.

The Voronoi cell $\mathcal{V}(p)$ for a site $p \in P$ is the set of all points in the plane that have $p$ as nearest site.

The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$.

Let $\mathcal{G}$ be the dual graph of $\text{Vor}(P)$.

The Delaunay graph $\mathcal{D}\mathcal{G}(P)$ is the straight line embedding of $\mathcal{G}$.
**Theorem:** The Delaunay graph of a planar point set is a plane graph.
If the point set $P$ is in *general position* then the Delaunay graph is a triangulation.
Empty Circle Property

**Theorem:** Let $P$ be a set of points in the plane, and let $\mathcal{T}$ be a triangulation of $P$. Then $\mathcal{T}$ is a Delaunay triangulation of $P$ if and only if the circumcircle of any triangle of $\mathcal{T}$ does not contain a point of $P$ in its interior.
**Theorem:** Let $P$ be a set of points in the plane. A triangulation $\mathcal{T}$ of $P$ is legal if and only if $\mathcal{T}$ is a Delaunay triangulation.
**Theorem:** Let $P$ be a set of points in the plane. 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 $P$. 
**Algorithm** DelaunayTriangulation($P$)

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

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

1. Initialize $\mathcal{T}$ as the triangulation consisting of an outer triangle $p_{-2}p_{-1}p_0$ containing points of $P$.
2. Compute a random permutation $p_1, p_2, \ldots, p_n$ of $P$.
3. for $r \leftarrow 1$ to $n$
4.     do
5.         LOCATE($p_r, \mathcal{T}$)
6.         INSERT($p_r, \mathcal{T}$)
7.     Discard $p_0, p_{-1}$ and $p_{-2}$ with all their incident edges from $\mathcal{T}$.
8. return $\mathcal{T}$
$p_r$ lies in the interior of a triangle

$pr$ falls on an edge
**Randomized Incremental Construction**

\[ \text{INSERT}(p_r, T) \]

1. **if** \( p_r \) lies in the interior of the triangle \( p_i p_j p_k \)
2. **then** Add edges from \( p_r \) to the three vertices of \( p_i p_j p_k \), thereby splitting \( p_i p_j p_k \) into three triangles.
3. \text{LEGALIZEEDGE}(p_r, \overline{p_i p_j}, T)
4. \text{LEGALIZEEDGE}(p_r, \overline{p_j p_k}, T)
5. \text{LEGALIZEEDGE}(p_r, \overline{p_k p_i}, T)
6. **else** (* \( p_r \) lies on an edge of \( p_i p_j p_k \), say the edge \( \overline{p_i p_j} \) *)
7. Add edges from \( p_r \) to \( p_k \) and to the third vertex \( p_l \) of the other triangle that is incident to \( \overline{p_i p_j} \), thereby splitting the two triangles incident to \( \overline{p_i p_j} \) into four triangles.
8. \text{LEGALIZEEDGE}(p_r, \overline{p_i p_l}, T)
9. \text{LEGALIZEEDGE}(p_r, \overline{p_i p_j}, T)
10. \text{LEGALIZEEDGE}(p_r, \overline{p_j p_k}, T)
11. \text{LEGALIZEEDGE}(p_r, \overline{p_k p_l}, T)
LEGALIZEEDGE($p_r, \overline{pij}$; $\cal T$)

1. (* The point being inserted is $p_r$, and
   $\overline{pij}$ is the edge of $\cal T$ that may need to
   be flipped. *)

2. if $\overline{pij}$ is illegal

3. then Let $pijp_k$ be the triangle
   adjacent to $p_rpij$ along $\overline{pij}$.

4. (* Flip $\overline{pij}$: *) Replace $\overline{pij}$
   with $\overline{prp_k}$.

5. LEGALIZEEDGE($p_r, \overline{pik}$; $\cal T$)

6. LEGALIZEEDGE($p_r, \overline{pkg}$; $\cal T$)
Randomized Incremental Construction

All edges created are incident to $p_r$. 
Randomized Incremental Construction

All edges created are incident to $p_r$.

**Correctness:** Are new edges legal?
Correctness:
For any new edge there is an empty circle through endpoints. New edges are legal.
Randomized Incremental Construction

**Initializing triangulation:** treat $p_{-1}$ and $p_{-2}$ symbolically. No actual coordinates. Modify tests for point location and illegal edges to work as if far away.

**Point location:** search data structure. Point visits triangles of previous triangulations that contain it.
Randomized Incremental Construction

- Split $\Delta_1$
- Flip $p_i p_j$
- Flip $p_i p_k$

Geometric Algorithms Lecture 5: Delaunay Triangulations
1. Expected total number of triangles created in $O(n)$
2. Expected total number of triangles visited while search point location data structure: $O(n \log n)$

We will only consider the first (see book for second)
How many triangles are created?
Lemma: Expected total number of triangles created is at most $9n + 1$.

How many triangles are created when inserting $p_r$?
Lemma: Expected total number of triangles created is at most $9n + 1$.

- How many triangles are created when inserting $p_r$?
- Backwards analysis: Any point of $p_1, \ldots, p_r$ has the same probability $1/r$ to be $p_r$. 
Lemma: Expected total number of triangles created is at most $9n + 1$.

- How many triangles are created when inserting $p_r$?
- Backwards analysis: Any point of $p_1, \ldots, p_r$ has the same probability $1/r$ to be $p_r$.
- Expected degree of $p_r \leq 6$. 
Analysis

Lemma: Expected total number of triangles created is at most $9n + 1$.

- How many triangles are created when inserting $p_r$?
- Backwards analysis: Any point of $p_1, \ldots, p_r$ has the same probability $1/r$ to be $p_r$.
- Expected degree of $p_r \leq 6$.
- Number of triangles created $\leq 2\text{degree}(p_r) - 3$ (Why? Count flips.)
**Lemma:** Expected total number of triangles created is at most $9n + 1$.

- How many triangles are created when inserting $p_r$?

- Backwards analysis: Any point of $p_1, \ldots, p_r$ has the same probability $1/r$ to be $p_r$.

- Expected degree of $p_r \leq 6$.

- Number of triangles created $\leq 2\text{degree}(p_r) - 3$
  (Why? Count flips.)

- $2 \cdot 6 - 3 = 9$
Lemma: Expected total number of triangles created is at most $9n + 1$.

- How many triangles are created when inserting $p_r$?
- Backwards analysis: Any point of $p_1, \ldots, p_r$ has the same probability $1/r$ to be $p_r$.
- Expected degree of $p_r \leq 6$.
- Number of triangles created $\leq 2\text{degree}(p_r) - 3$ (Why? Count flips.)
- $2 \cdot 6 - 3 = 9$
- + outer triangle
**Theorem:** The Delaunay triangulation of $n$ points can be computed in $O(n \log n)$ expected time.