Polygon Triangulation

Input: Simple polygon in plane

Specified as sequence of vertices in cyclic order

\[
\begin{align*}
2, 5 & \quad \text{NO} \\
3, 6 & \quad \text{NO}
\end{align*}
\]

Compute a (frugal, interior) triangulation of \( P \)

\[ \uparrow \]

Jordan curve theorem (1906)

\[ \rightarrow \text{Lennes (1911)} \]
\[ \text{Dehn (1899)} \quad \text{- unpublished} \]

Not Frugal

Find one interior diagonal

\[ \text{Lemma: Every simple polygon with \( \geq 3 \) vertices has an interior diagonal} \]

\[ \text{Proof:} \]

\[ \begin{align*}
\min x & \quad \max x \\
p, r & \quad \text{pred, succ of } q
\end{align*} \]

Case 1: \( pr \) is interior diagonal

Case 2: \( pr \) is not an (interior) diagonal

\( P \) has vertices inside \( \Delta pqr \)

Connect \( q \) to one of these vertices

\( S = \text{interior vertex furthest from } pr \)

\( qs \) is int. diagonal!
Algorithm: we can find a diagonal in $O(n)$ time

$\Rightarrow$ we can compute a triangulation in $O(n^2)$ time

$T(n) = T(k) + T(n-2-k) + O(n)$

$= O(n^2)$ worst case

Interesting problem:

Find a diagonal splits vertices $\frac{2}{3} + \frac{\sqrt{3}}{3}$ or better.

$O(n \log n)$ today

$O(n \log^* n)$ later

$O(n)$ Chazelle '90

Amato, Goodrich, Ramos OS?

---

Trapezoidal Decomposition

Shoot rays $\uparrow$ and $\downarrow$ from every vertex.

Stop at first edge.

PSLG

Events & Vertices

At each event in order:

Find succ & pred.

Build walls up and down.

Add $O(1)$ verts and edges to trap-decomp graph.

We can build trap-decomp.

in $O(n \log n)$ time using a sweep algo.
Define $p \equiv q$ ("equivalent") if some edges above and below.

Trapezoid = equivalence class!

How to triangulate

1. Build trap decomp
2. Discard exterior walls
   - interior
     
     Every trap has
     - a vertex on L wall
     - vertex on R wall

Trapezoid is boring unless both on floor or both on ceiling.

Every boring trap contains a diagonal!

Linear sequence of interesting traps

Walk all with defining pts

on the ceiling

"Monotone mountain"

Triangulate using

the 3-penny algorithm $O(n)$ time