Computing (with) Curves on Surfaces I

Jeff Erickson
University of Illinois

ADFOCS
August 29, 2005

The problems

- Given two paths on a surface, can one be continuously deformed into the other?
- Given a path π on a surface, compute the shortest path that can be continuously deformed into π.

Why do we care?

- These are simple, natural problems.
- Well, okay, they also have applications:
  - ◇ VLSI: river routing
  - ◇ GIS: map simplification
  - ◇ Robotics: motion planning

What’s a “surface”?

- A **2-manifold with boundary** is a Hausdorff space in which every point has a neighborhood homeomorphic to either the plane or a closed half-plane.
- More intuitively, a *compact* surface is a space obtained from a set of triangles by gluing pairs of equal-length edges together.

[Dehn Heegaard 07][Radó 35]

Simple examples

- Polygons
- Polygons with holes
- The plane minus a finite set of points

We'll consider more general surfaces next time.

Paths and cycles

- A **path** in a surface Σ is (the image of) a continuous function $\pi:[0,1]\to\Sigma$.
  Its endpoints are $\pi(0)$ and $\pi(1)$.
- A **cycle** in a surface Σ is (the image of) a continuous function $\gamma:S^1\to\Sigma$.
- A path or cycle is **simple** if it is one-to-one.

Paths and cycles (today)

- A **path** is a sequence of **k** line segments joined end to end, represented by vertex coordinates.
- A **cycle** is a circular sequence of **k** line segments joined end to end, represented by vertex coordinates.
- A path or cycle is **simple** if its segments intersect only at their common endpoints.

Homotopy

- Two cycles are **(freely) homotopic** if one can be continuously deformed into the other.
- A **(free) homotopy** between cycles γ and γ' is a continuous function $h:[0,1]\times S^1 \to \Sigma$ such that $h(0,t) = \gamma(t)$ and $h(1,t) = \gamma'(t)$ for all t.
- A cycle is **contractible** if it is homotopic to a constant cycle ($\gamma(t)=x$ for all t).

Path homotopy

- Two paths are **homotopic** if one can be continuously deformed into the other, keeping the endpoints fixed.
- A **path homotopy** between paths π and π' is a continuous function $h:[0,1]\times[0,1]\to\Sigma$ such that

1. $h(0,t) = \pi(t)$ and $h(1,t) = \pi'(t)$ for all t
2. $h(s,0) = \pi(0) = \pi'(0)$ and $h(s,1) = \pi(1) = \pi'(1)$ for all s

The problems, restated

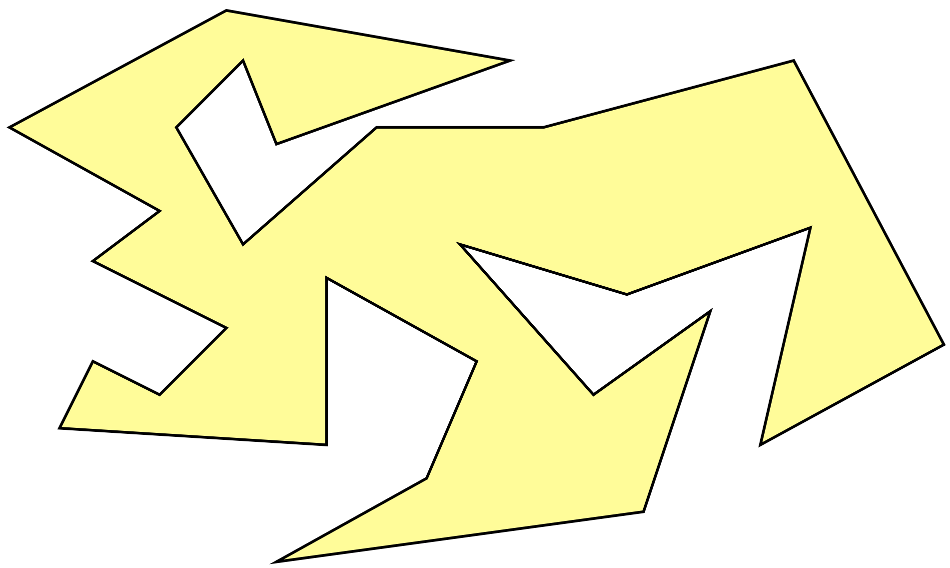
- Given two paths or cycles on a surface, are they homotopic?
- Given a path or cycle π on a surface, compute the shortest path or cycle homotopic to π.

Polygons

John Hershberger and Jack Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications* 4(2):63–97, 1994.

Polygon

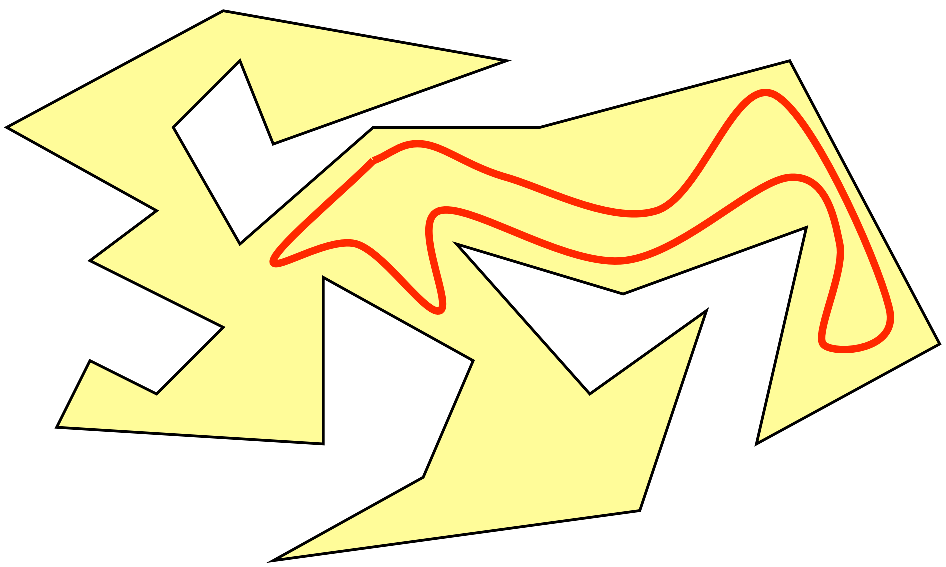
A region in the plane enclosed by a simple cycle of **n** line segments, represented by a sequence of vertex coordinates.



Homotopy is trivial

Polygons are **simply connected**:

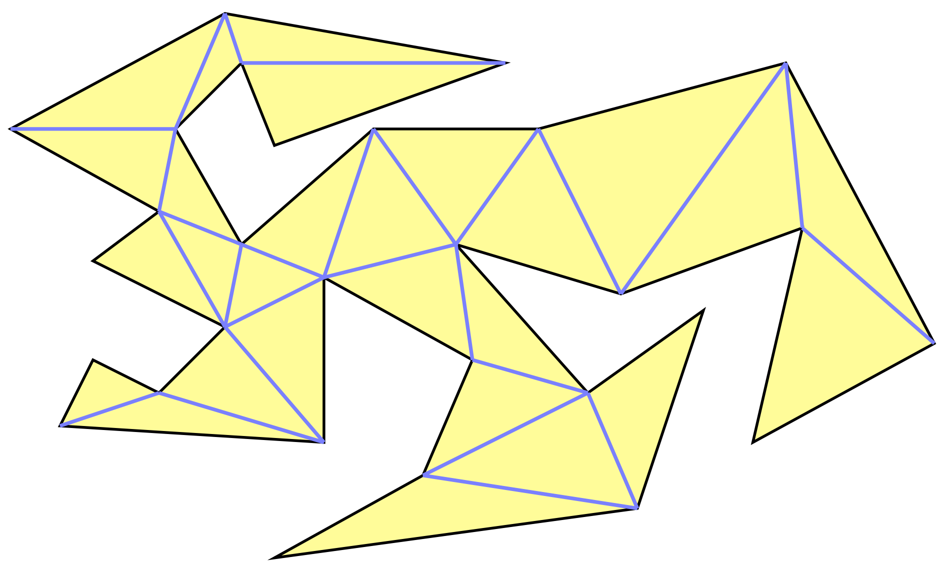
- Every cycle in a polygon is contractible.
- Two paths in a polygon are homotopic if and only if they have the same endpoints.



Testing homotopy

Preprocessing step: **triangulate** the polygon

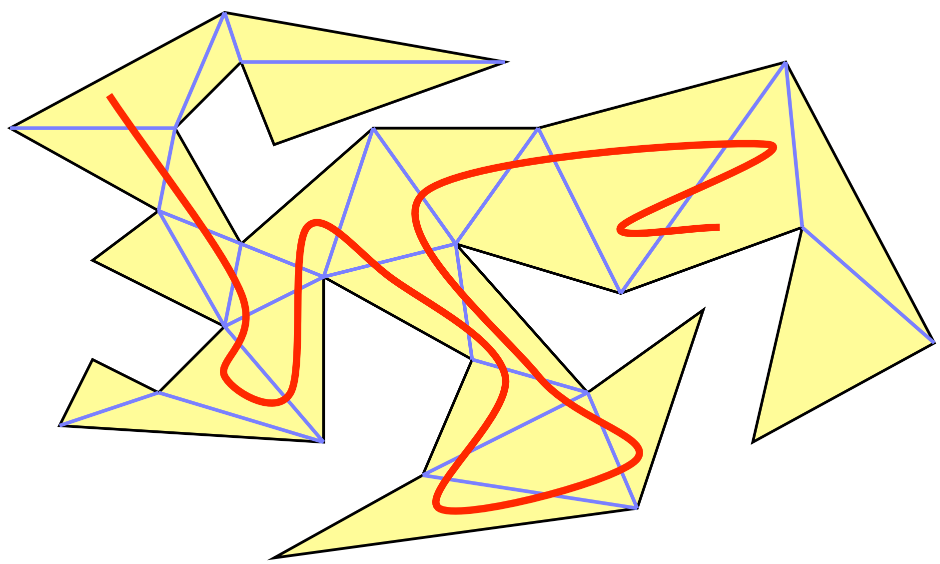
- ◇ O(n log n) [sweep-line]
- ◇ O(n log* n) expected [Seidel]
- ◇ O(n) [Chazelle]



Crossing sequence

- Sequence of edges crossed by the path π
- Easy to compute in $O(k+x)$ time
  k = number of segments in π
  x = number of edge crossings = $O(kn)$

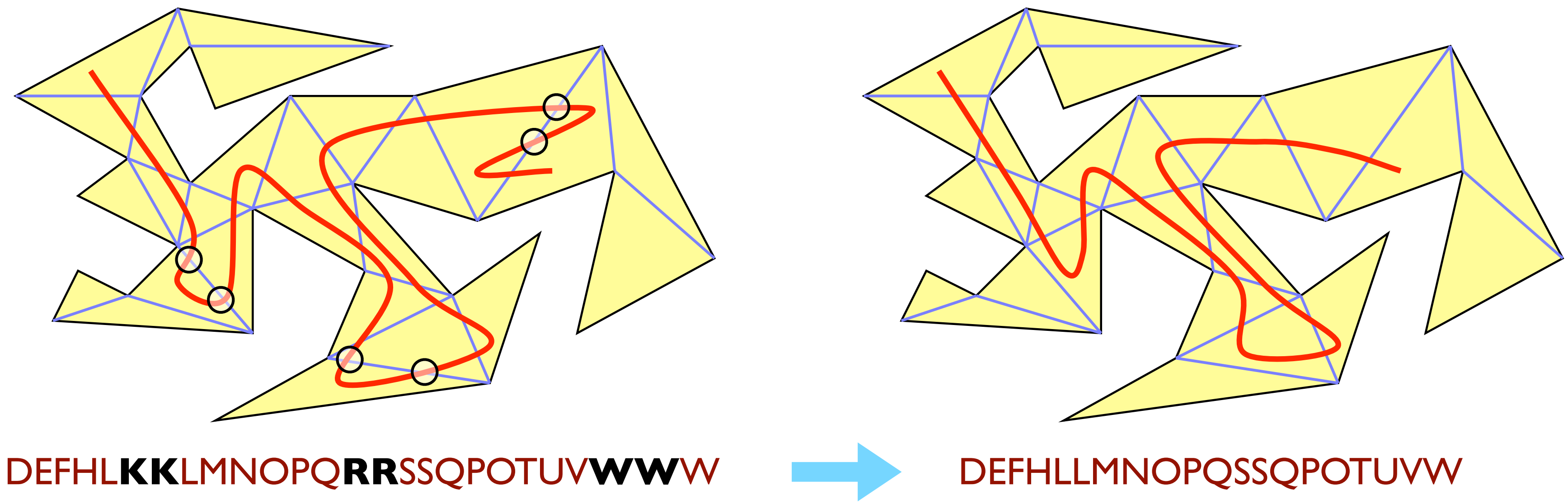
C_α in [HS94]
Δ_α in [HS94]



DEFHLKKLMNOPQRRSSQPTUVVV

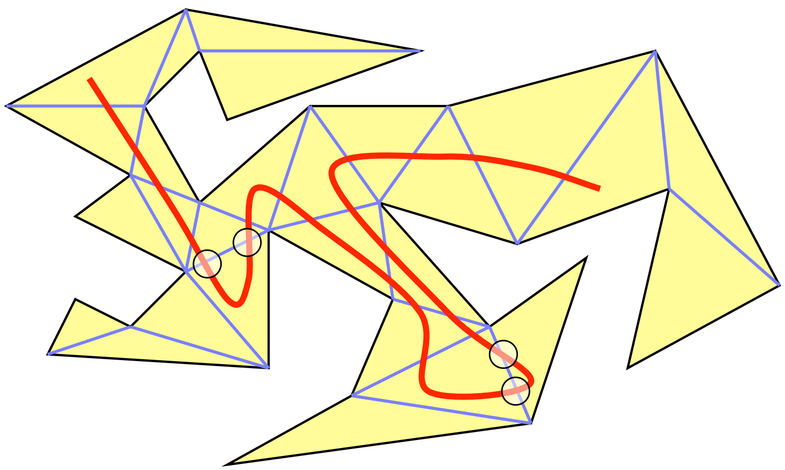
Reduction

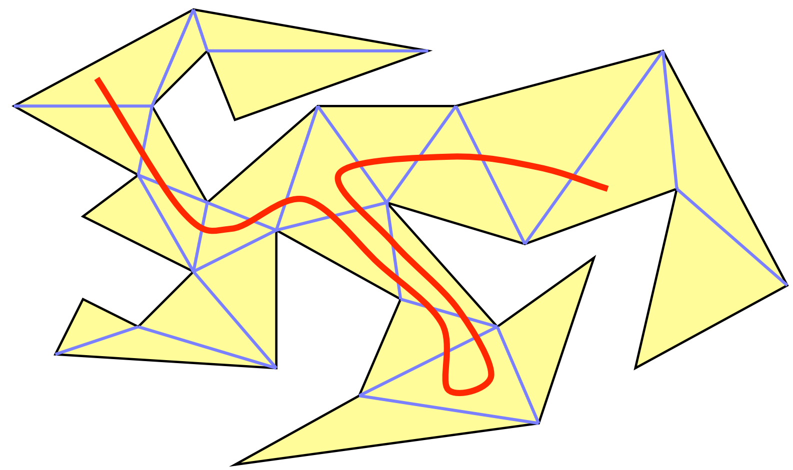
We can **reduce** any crossing sequence by removing all repeating pairs in O(x) time.



Reduction

We can **reduce** any crossing sequence by removing all repeating pairs in O(x) time.

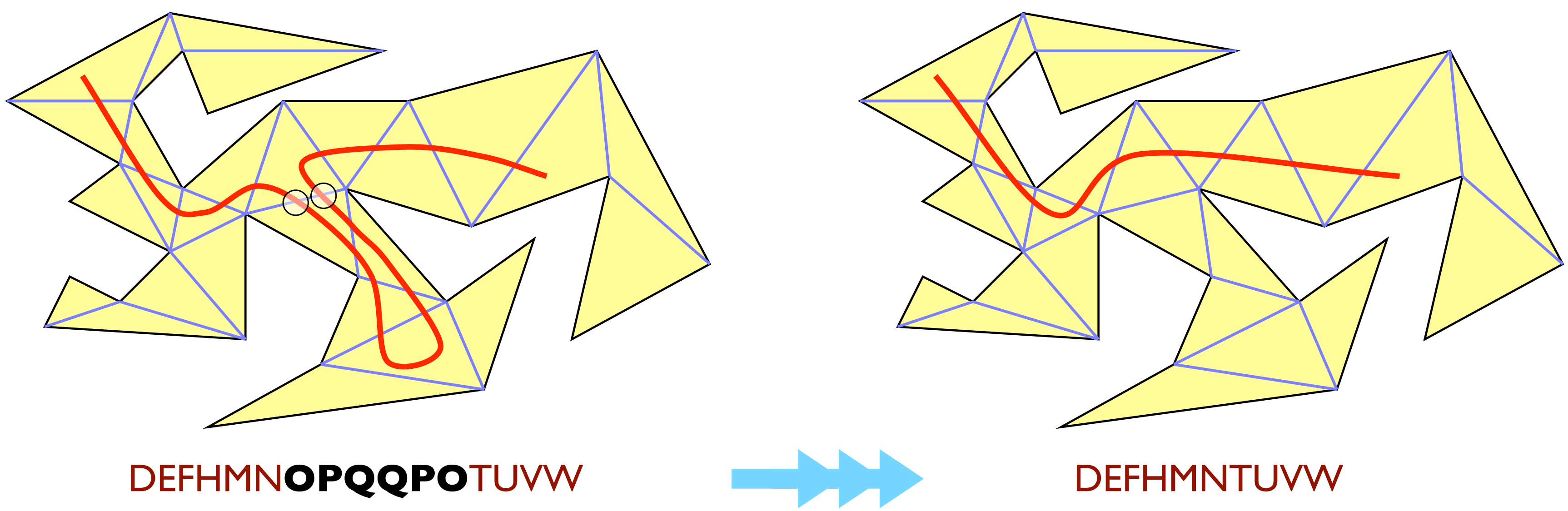




DEFH**LL**MNOPQ**SS**QPOTUVW → DEFHMNOPQQPOTUVW

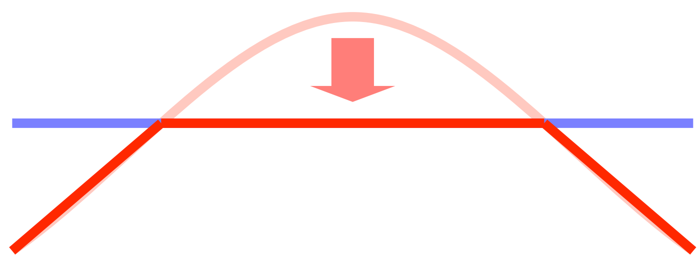
Reduction

We can **reduce** any crossing sequence by removing all repeating pairs in O(x) time.



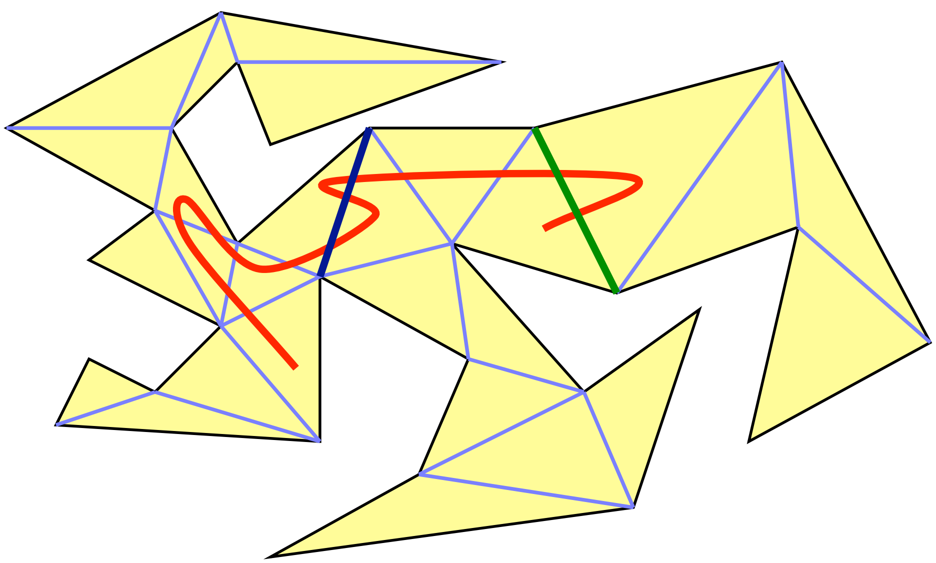
No bigons

Lemma: In any simply connected space, two shortest paths cross at most once.



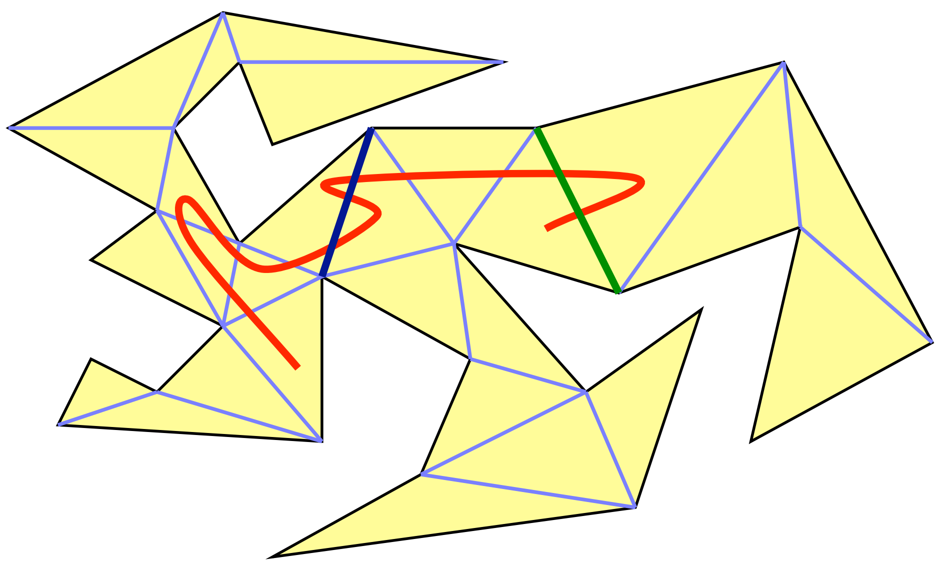
Reduction

Lemma: *An edge appears in the reduced crossing sequence iff π crosses it an odd number of times.*



Reduction

Lemma: *Each edge appears in the reduced crossing sequence at most once.*



Reduction

Lemma: *The reduced crossing sequence of π equals the crossing sequence of the shortest path homotopic to π.*

Proof: *Every* path homotopic to π crosses all the edges in π's reduced crossing sequence. Crossing *more* edges (or the same edges more than once) makes the path longer.

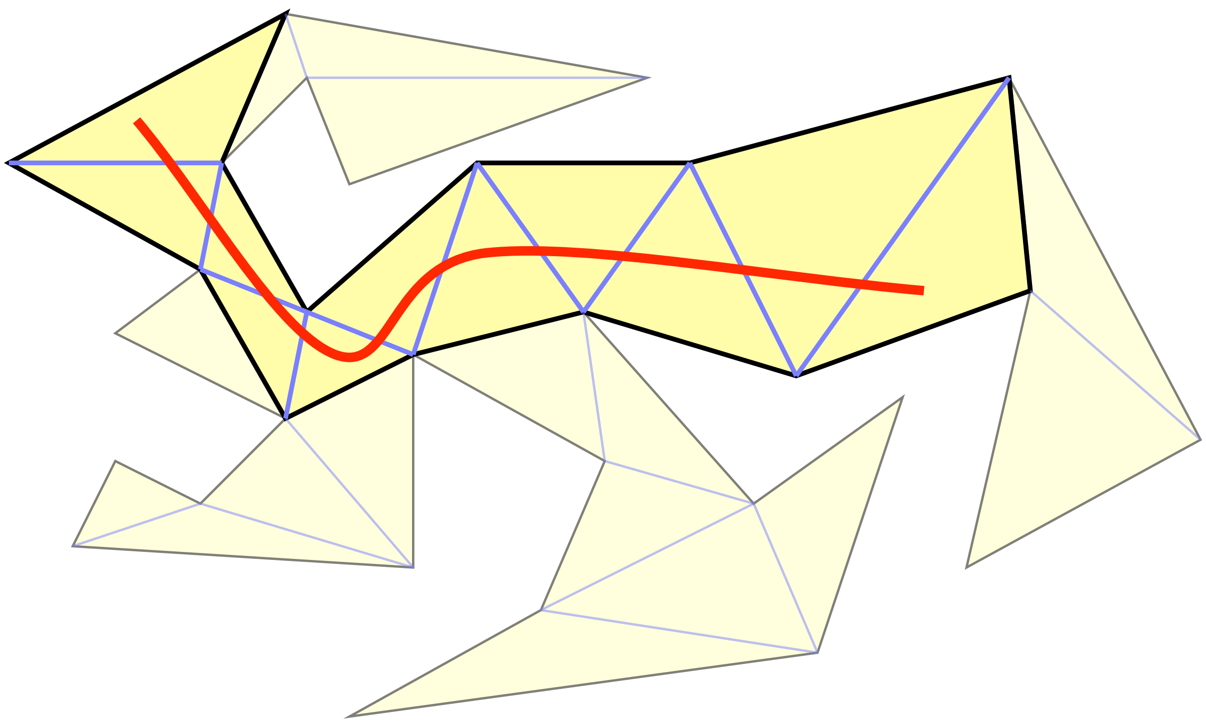
Testing homotopy

Theorem: *Two paths are homotopic if and only if their reduced crossing sequences are identical.*

Given two paths inside a polygon, we can decide whether they are homotopic in $O(nk)$ time.

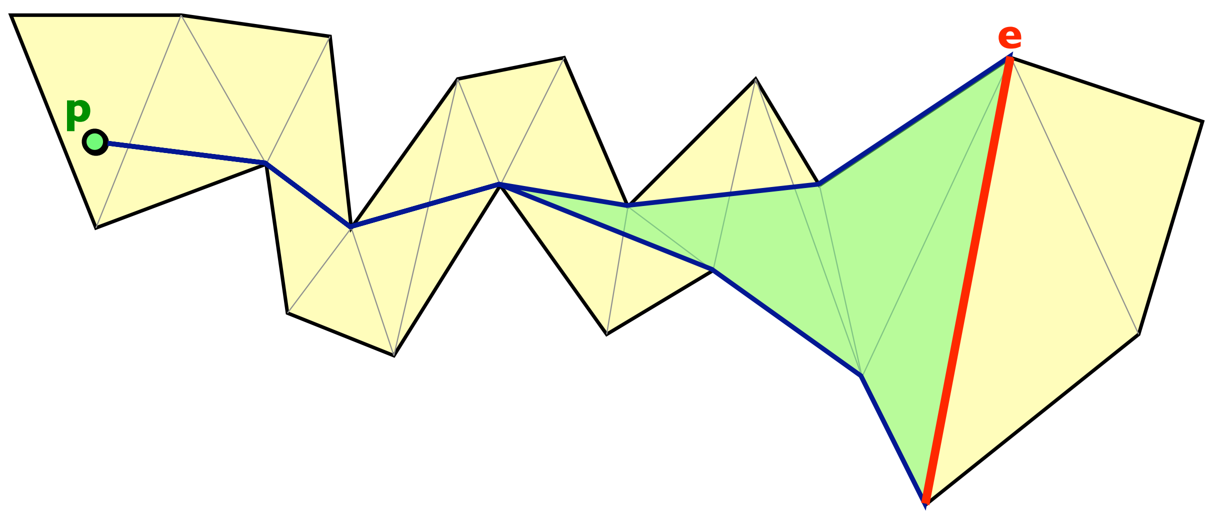
Sleeve

Sequence of triangles containing the reduced path; determined by the reduced crossing sequence



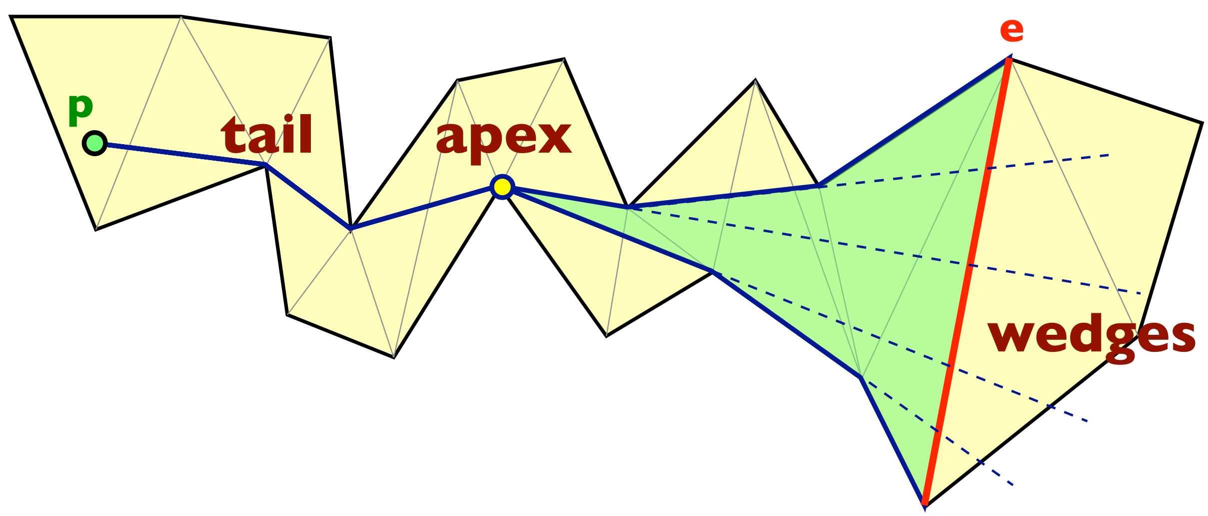
Shortest path in a sleeve

Any source point p and target edge e define a **funnel** of shortest paths.

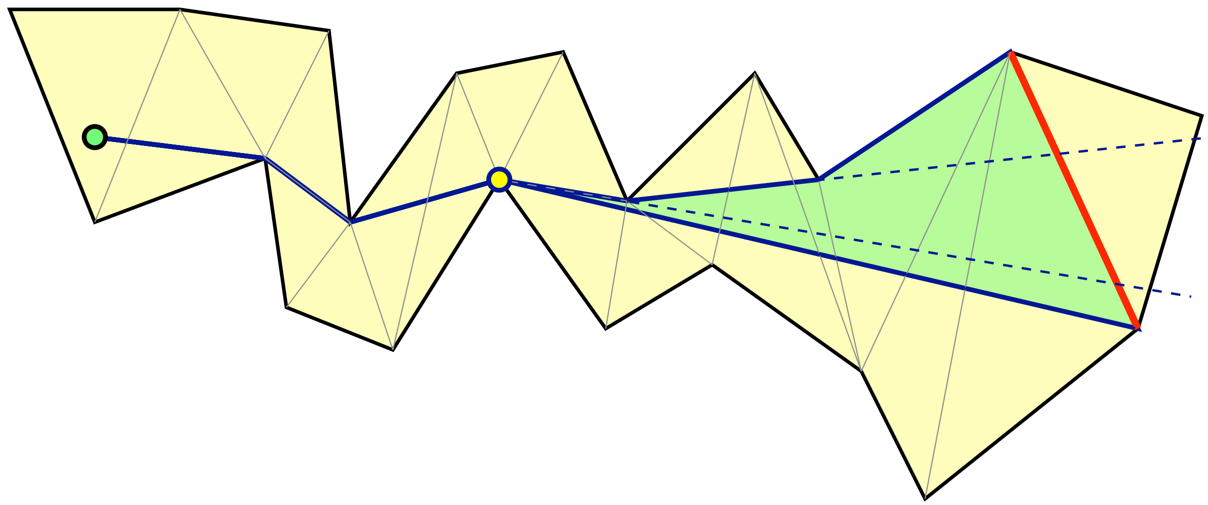


[Chazelle] [Lee and Preparata]

Anatomy of a funnel



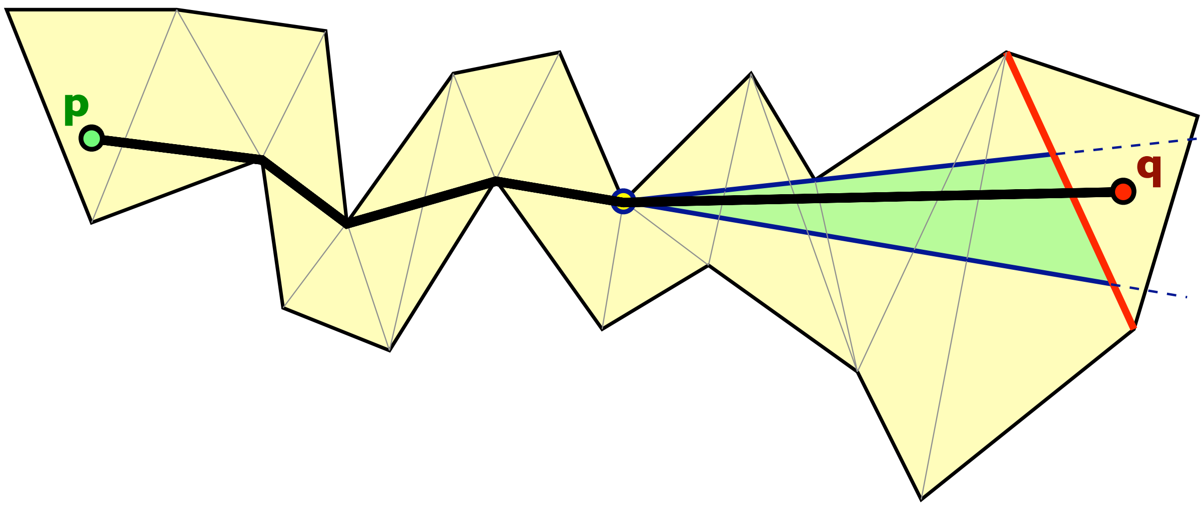
Extending a funnel



O(1) time to delete each vertex
+ O(1) time to add one new vertex

Total time = O(x), where x = #diagonals

Finding the shortest path



When the funnel reaches the last edge, restrict the funnel to the wedge containing the target.

Shortest path = tail + one line segment

Summary

To compute the shortest path homotopic to π:

1. Triangulate the polygon: $O(n \log n)$
2. Compute the crossing sequence of π: $O(x+k)$
3. Reduce the crossing sequence: $O(x)$
4. Construct the sleeve: $O(x)$
5. Compute the shortest path inside the sleeve using the funnel algorithm: $O(x)$

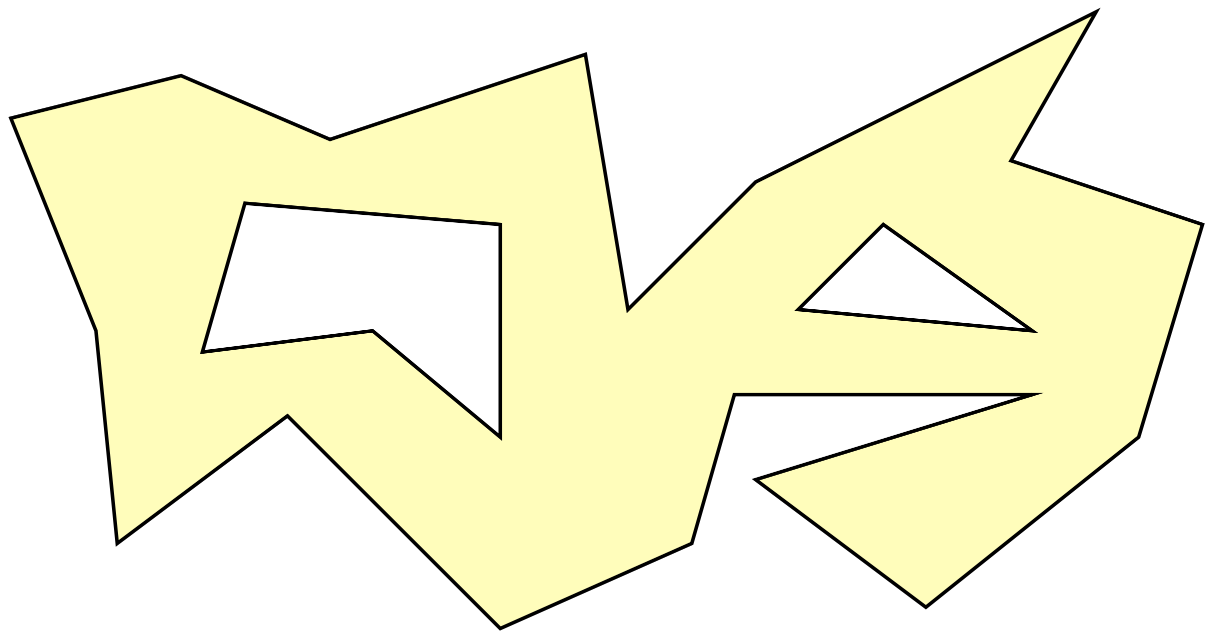
Overall time: $O(n \log n + x) = O(n \log n + nk)$

Polygons with holes

John Hershberger and Jack Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications* 4(2):63–97, 1994.

Polygon with holes

A polygon with one or more smaller polygons removed from its interior



Shortest homotopic paths

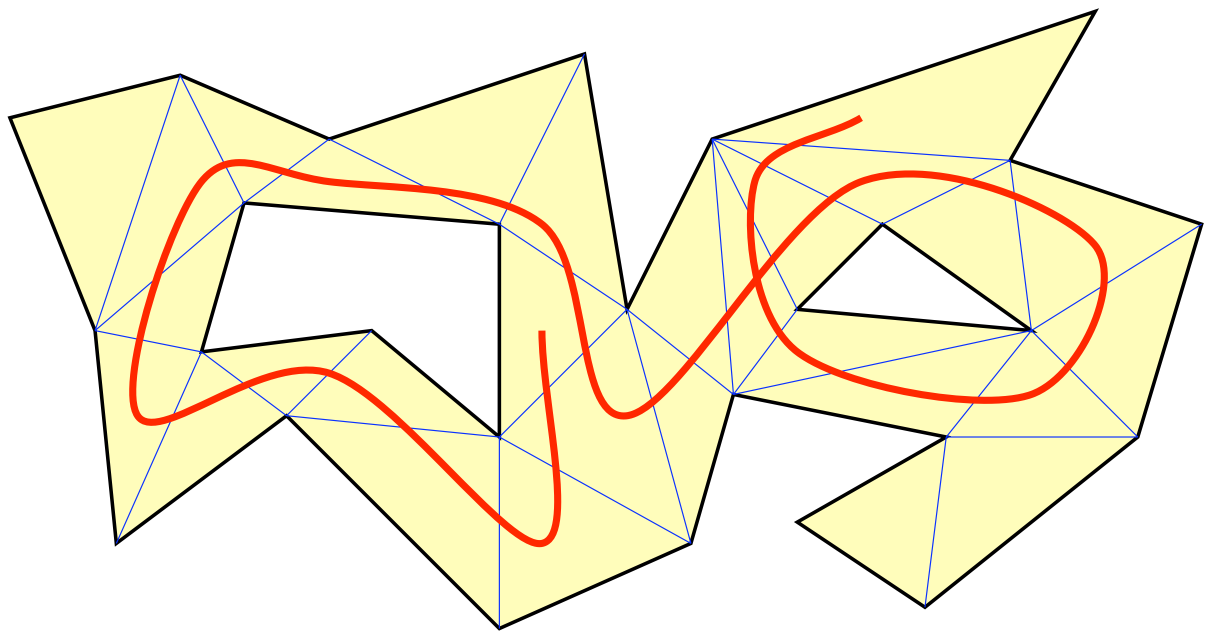
Same basic algorithm!

1. Triangulate the polygon: $O(n \log n)$
2. Compute the crossing sequence: $O(x+k)$
3. Reduce the crossing sequence: $O(x)$
4. **Construct the sleeve: $O(x)$**
5. Compute the shortest path inside the sleeve: $O(x)$

Total time: $O(n \log n + x) = O(n \log n + nk)$

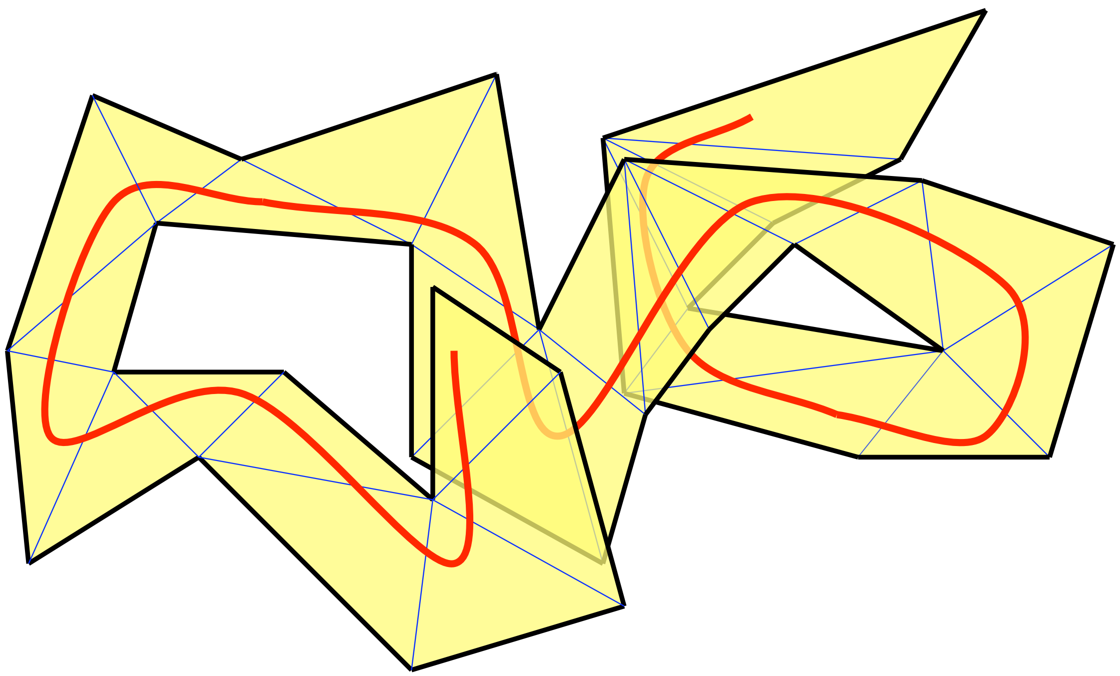
The only change

Whenever the reduced path enters a triangle, glue **a new copy of** that triangle onto the sleeve.



The only change

Geometrically, the sleeve overlaps itself, but the funnel algorithm doesn't care!



— Break —

Plane minus points

Sergio Cabello, Yuanxin Liu, Andrea Mantler, and Jack Snoeyink. Testing homotopy for paths in the plane. *Discrete & Computational Geometry* 31(1):61--81, 2004.

Sergei Bespamyatnikh. Computing homotopic shortest paths in the plane. *Journal of Algorithms* 49:284–303, 2003.

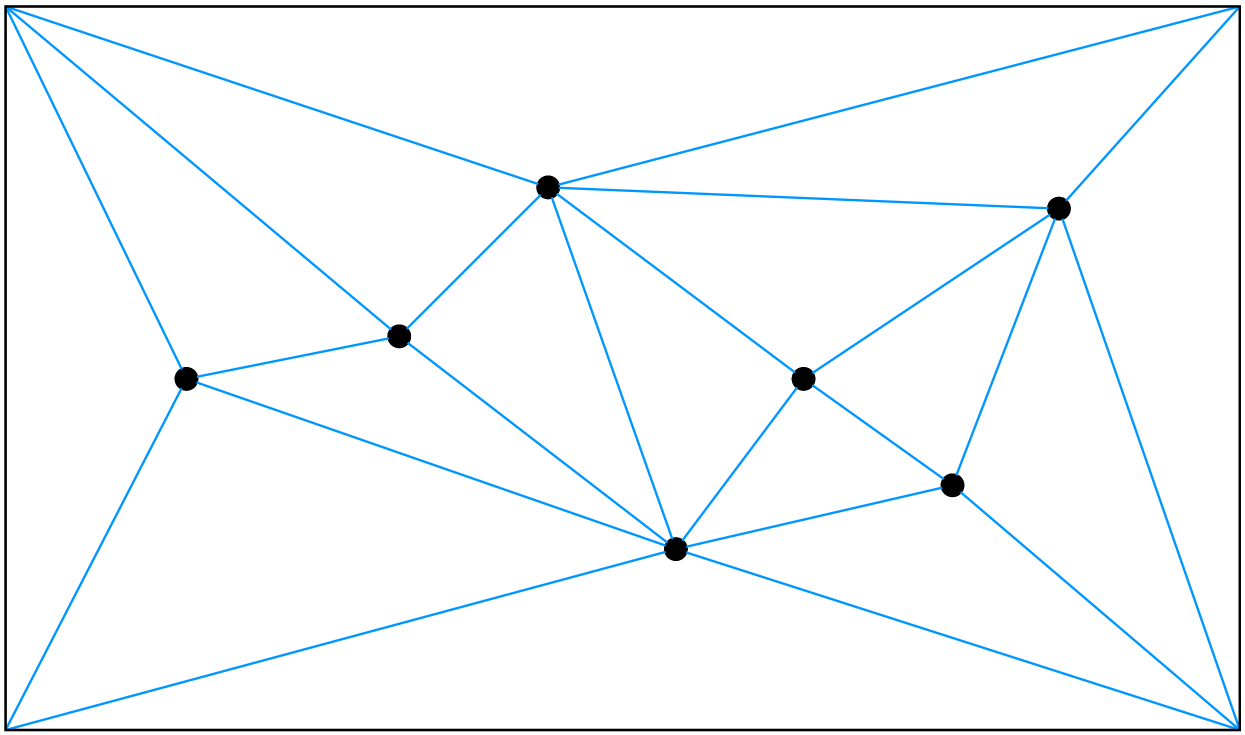
The problems, again

Let P be a set of n points in the plane in general position. Amen.

Given two paths in $\mathbf{R}^2\backslash P$, decide whether they are homotopic.

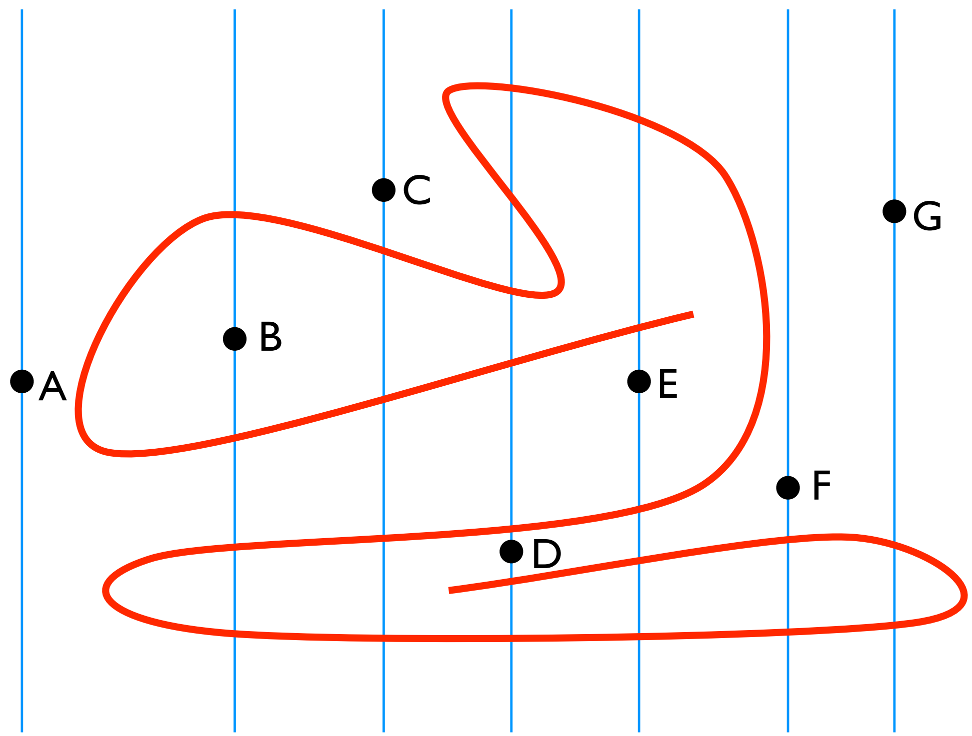
Given a path π in $\mathbf{R}^2\backslash P$, compute the shortest path homotopic to π.

“Polygon” with “holes”



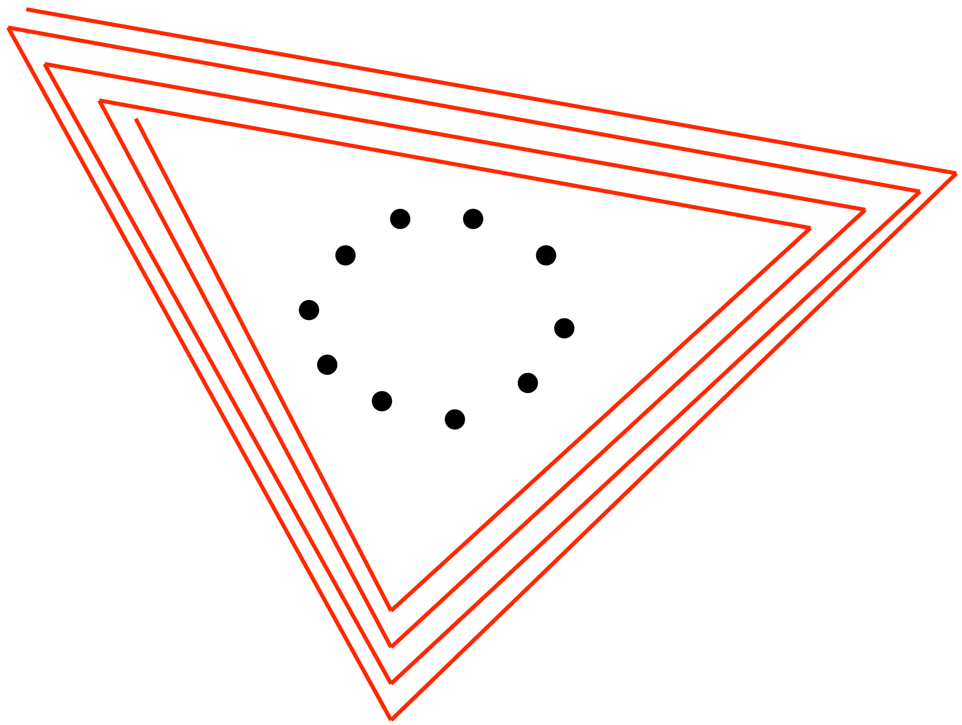
So we can test homotopy in O(nk) time.

Slab decomposition



Crossing sequence:
defggfedcbbcDeEDDDcBbcDE

Worst-case input



Crossing sequence has length $\Omega(nk)$
Shortest homotopic path has $\Omega(nk)$ edges

Faster! Faster!

To improve the running time, we must represent the crossing sequence implicitly.

Let's start with easier problem: Given a *simple* cycle in in $\mathbf{R}^2 \setminus P$, decide whether it is contractible.

(This is easy using other methods.)

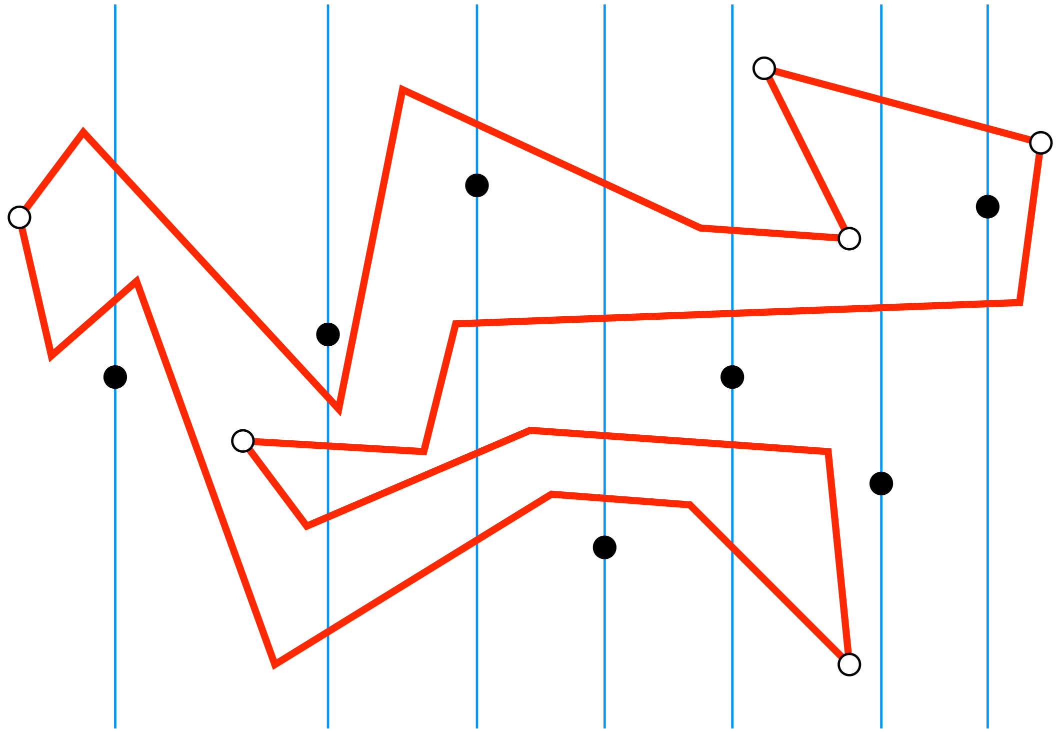
Outline

1. Break the cycle into x-monotone paths
2. Vertically order the paths and points
3. Replace paths with horizontal line segments
4. Compute compressed crossing word
5. Apply crossing reductions
6.

The cycle is contractible if and only if the final reduced crossing word is empty.

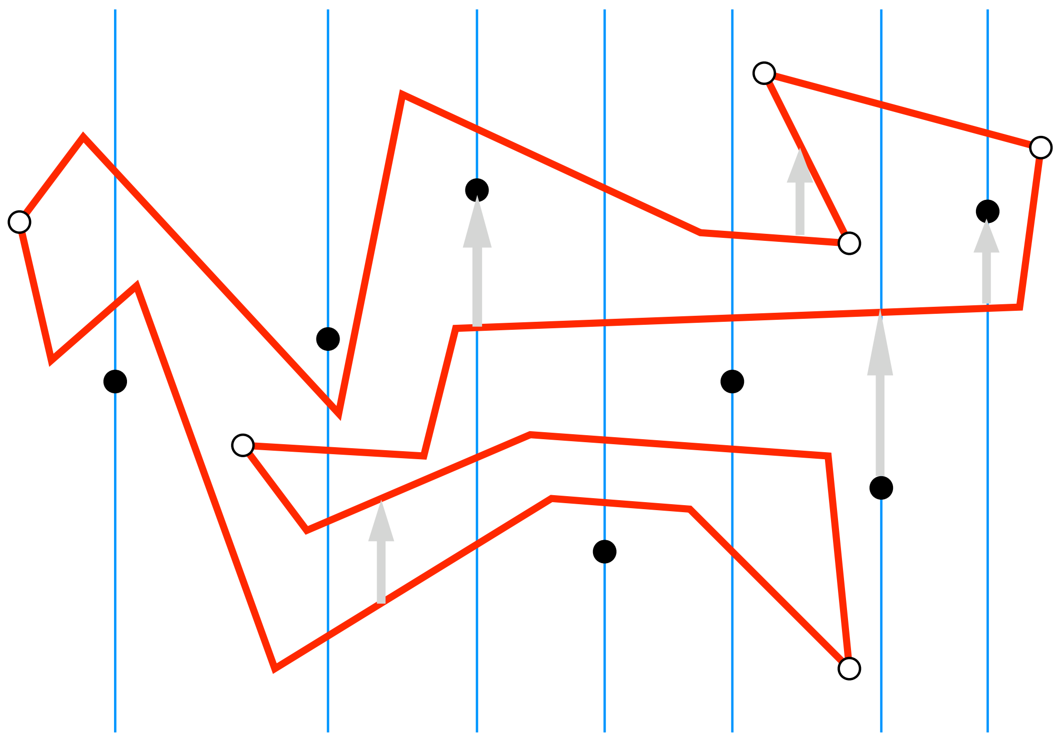
x-monotone

= intersects any vertical line in at most one point



Vertical ordering

The relationship "x is above y" is a partial order.



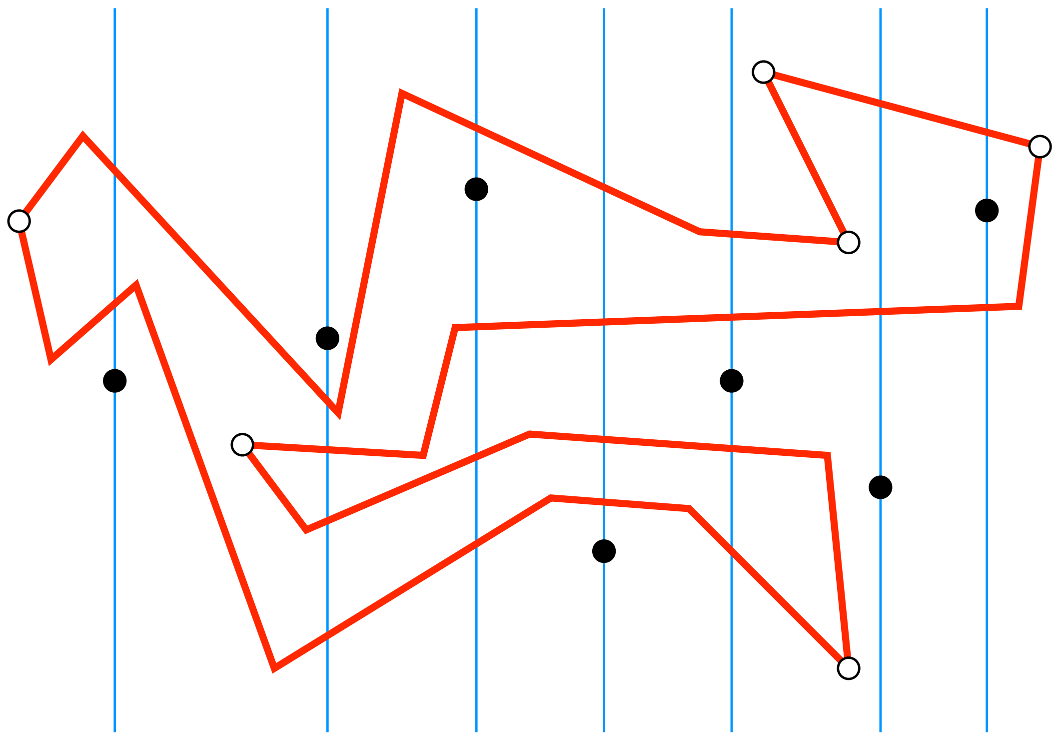
Rectification

We can replace the points and paths with points and horizontal segments with the same above/below partial order:

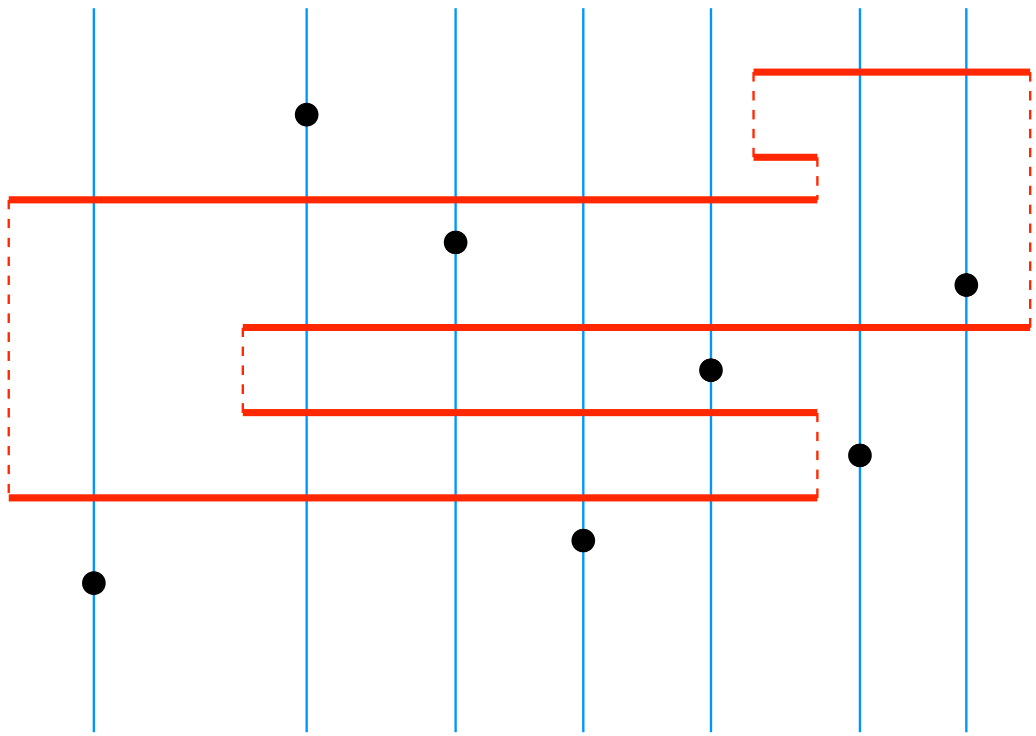
- Keep x-coordinates
- Replace y-coordinates with ranks

This changes the geometry *but not the topology*!

Rectification

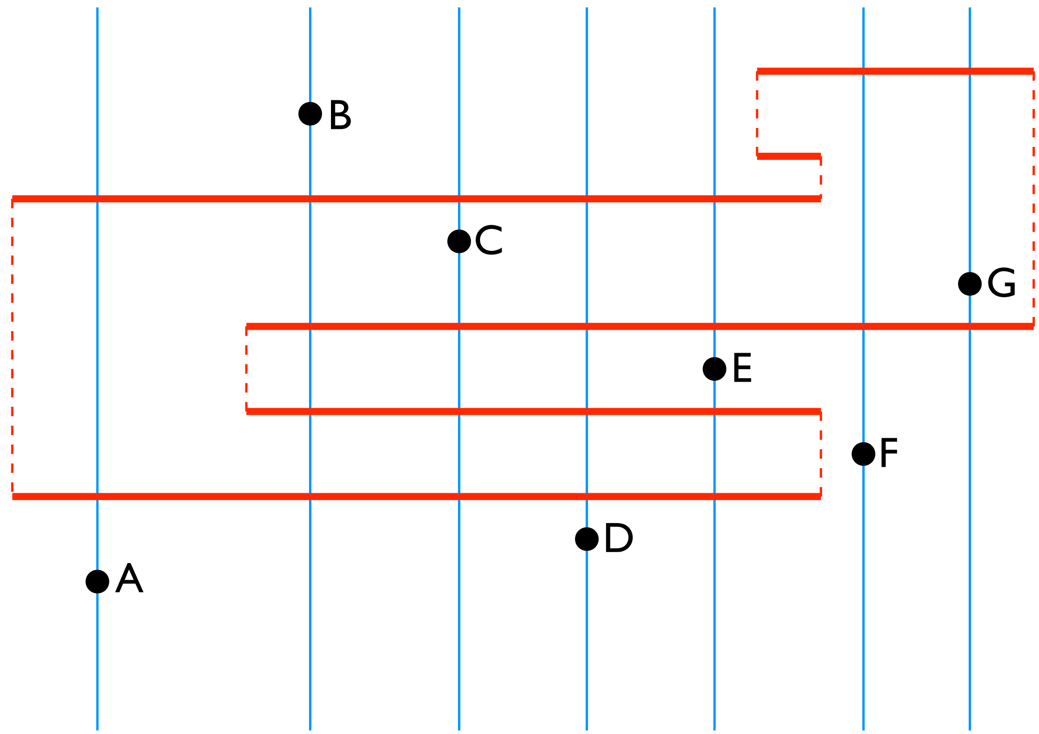


Rectification



Compressed crossing word

Store first and last crossing and vertical rank of each horizontal segment

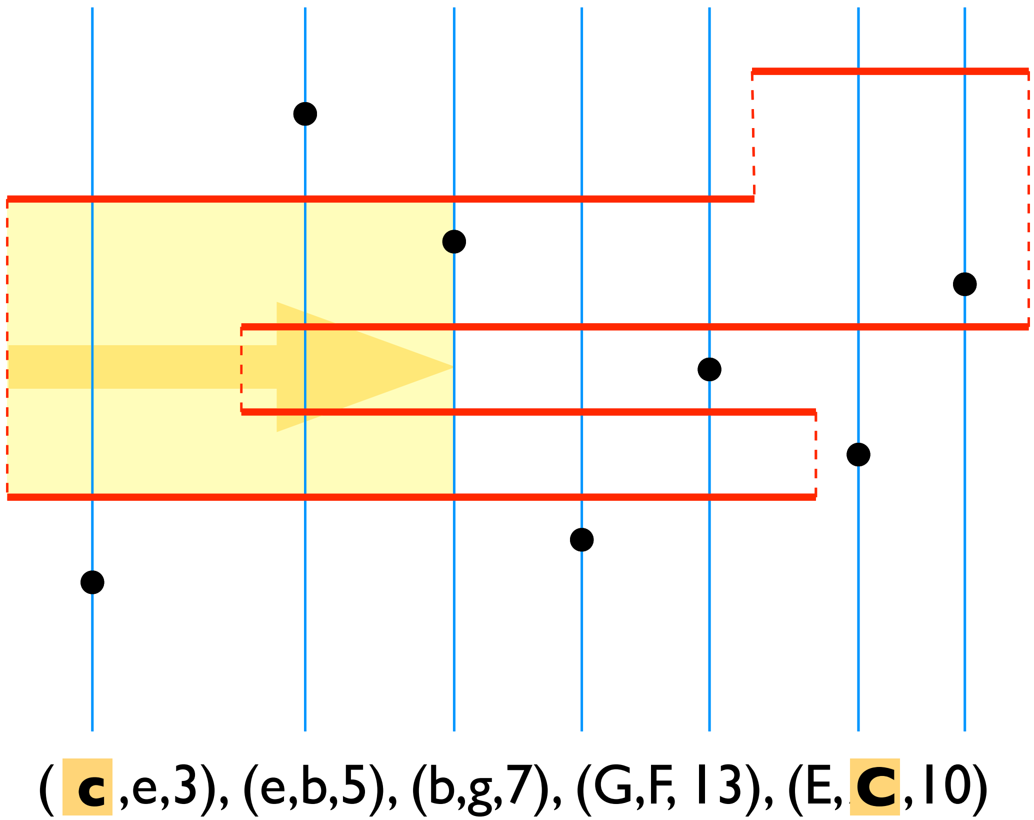


Crossing word: AbcDe eDcb bcDEFg GF EDCbA

Compressed: (A,e,3), (e,b,5), (b,g,7), (G,F, 13), (-,-,11), (E,A,10)

Reductions

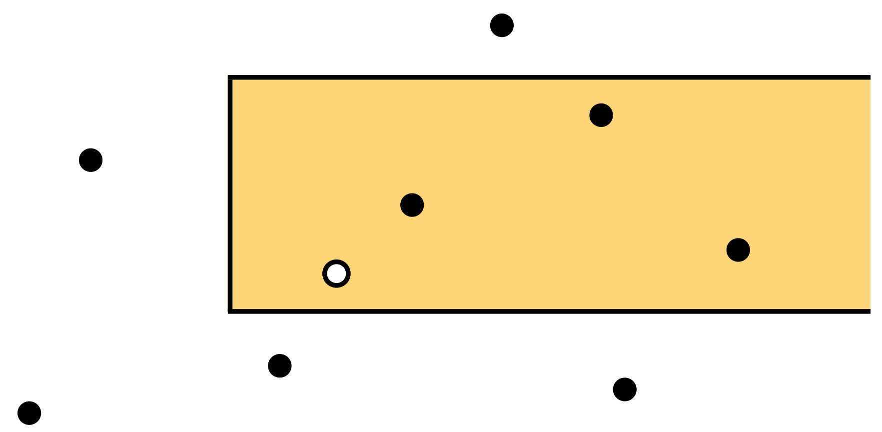
Move each U-turn as far inward as possible.
Move each step as far to the right as possible.



Geometric primitive

Given a three-sided rectangle, find the enclosed point closest to the "end"

- O(log n) query time
- O(n log n) preprocessing time



Analysis

Each reduction either moves a vertical segment to a point or removes a horizontal segment.

After $O(k)$ reductions, there's nothing left to do.

Each reduction takes $O(\log n)$ time.

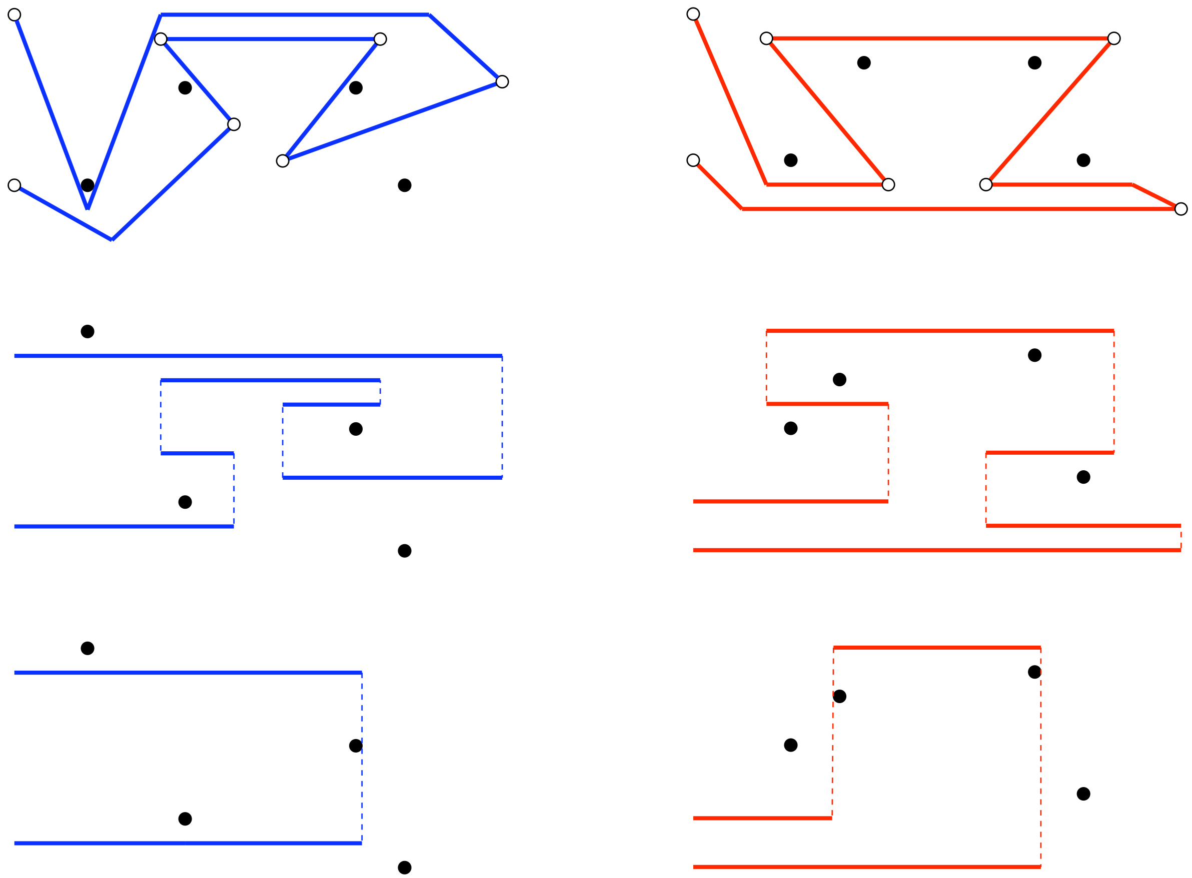
Overall reduction time is $O(n \log n + k \log n)$.

Testing homotopy

- Two paths are homotopic iff their reduced crossing words are identical.
- If the paths are simple, we can compute their compressed reduced crossing words quickly.
- How do we compare compressed crossing words?

Main problem

Each rectification uses its own ranking order.

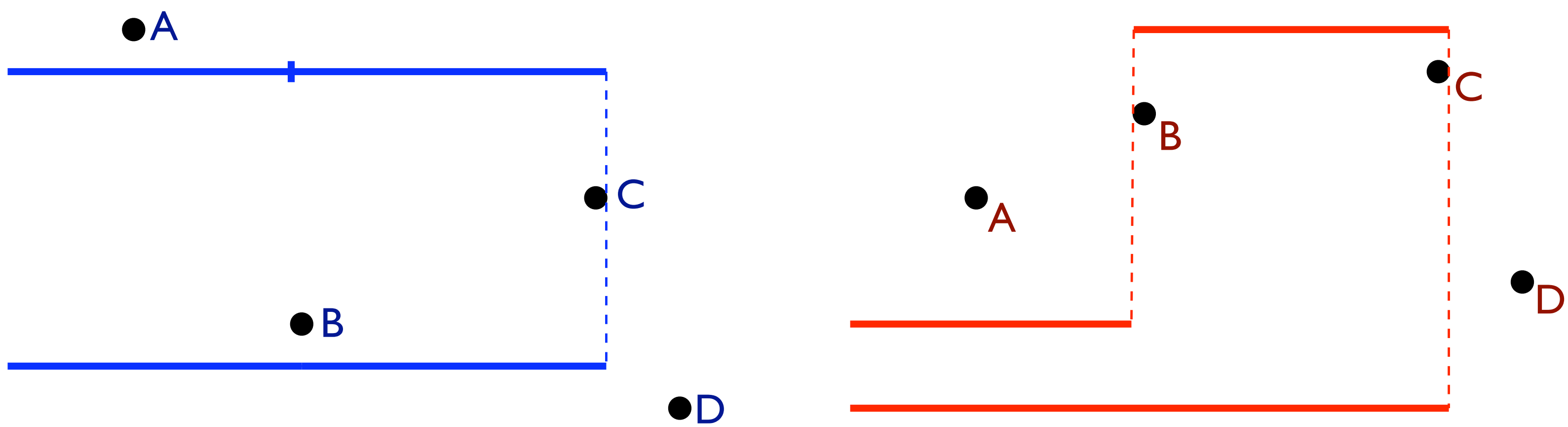


(a,C,5),(c,a,1)

(a,a,3),(B,C,7),(c,a,1)

Checking equivalence

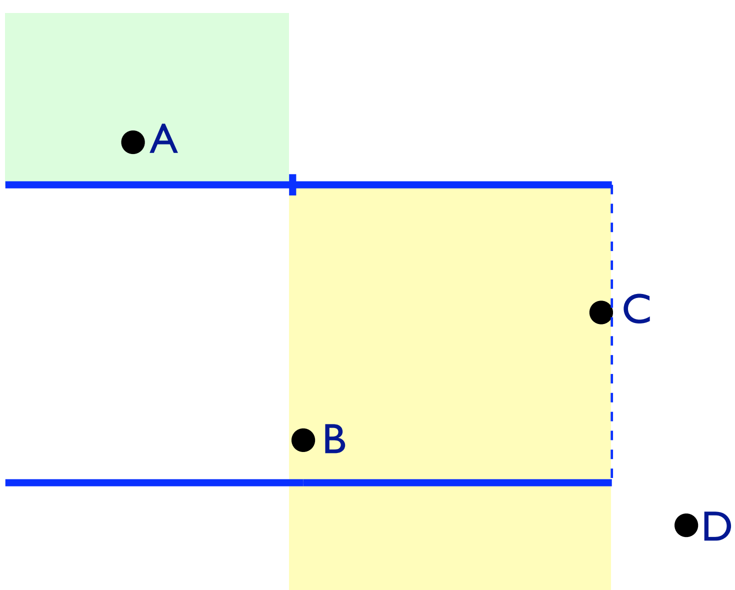
Split segments so endpoint x-coordinates match.

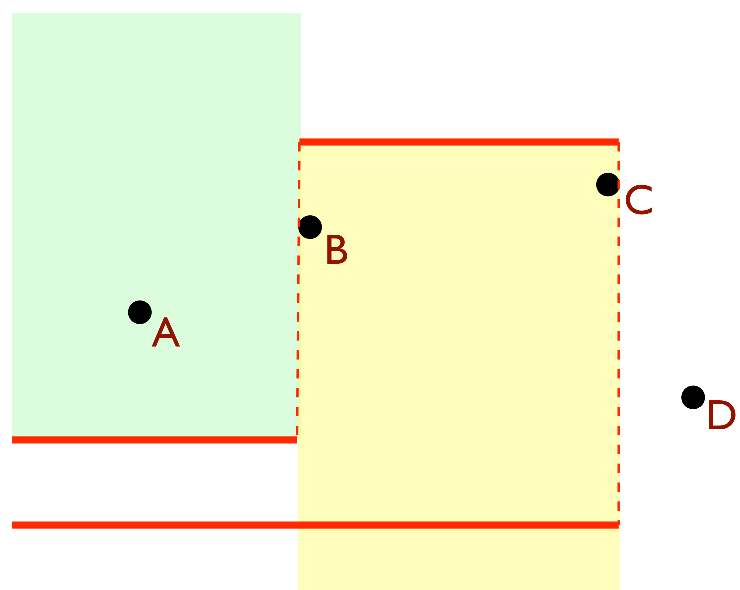


If this is not possible, or if segment directions don't match, the paths are not homotopic.

Checking equivalence

For each segment, check that the same points are above and below in both pictures.





3-sided queries again

- Weight each point is by its rank on the left
- For each segment, find the lowest rank point above it and the highest rank point below it, in both pictures.
- Paths are homotopic iff all these ranks match.

Summary

To determine whether two paths are homotopic:

1. Sort the points from left to right: $O(n \log n)$
2. Decompose paths into monotone pieces: $O(k)$
3. Compute vertical orderings: $O((n + k) \log (n + k))$
4. Rectify paths, compressed crossing words: $O(n + k)$
5. Reduce rectified paths: $O(n \log n + k \log n)$
6. Compare results: $O(k \log n)$

Overall time: $O((n + k) \log (n + k))$

Shortest homotopic paths

Combining rectified paths, compressed crossing sequences, funnel algorithm, other ideas....

- $O(n \log^2 n + k \log n)$ time if the input path is simple
- $O((n + n^{2/3}k^{2/3} + k) \text{ polylog } n)$ in general
- k = number of edges in input **and** output paths

[Bespamyatnikh]

That's all for now.