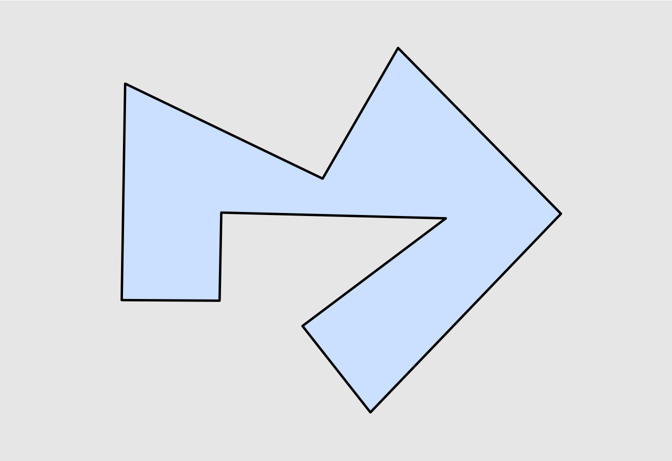
k-Star-shaped Polygons

Jeff Sember
William Evans

University of British Columbia

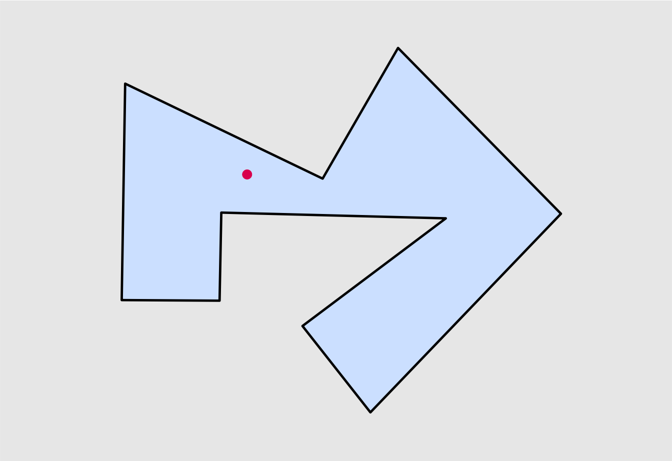
August 11, 2010

Introduction



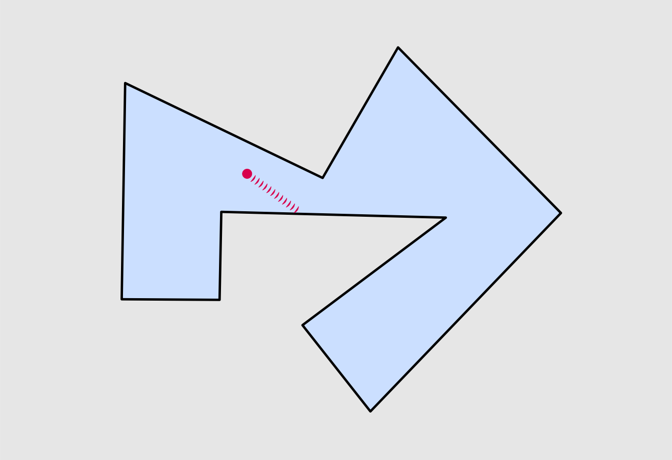
Placing a transmitter for a polygonal building

Introduction



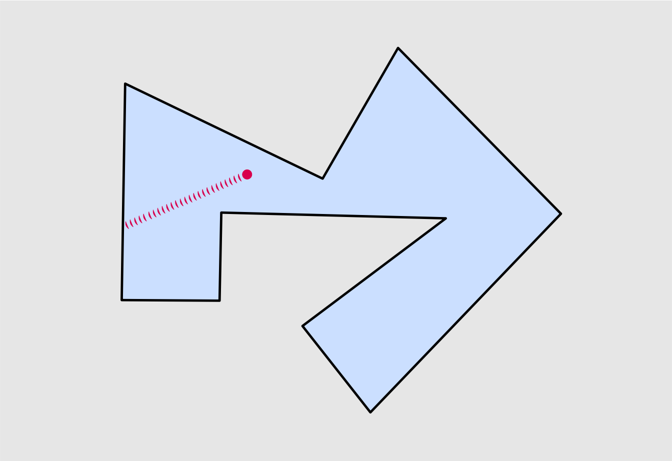
Placing a transmitter for a polygonal building

Introduction



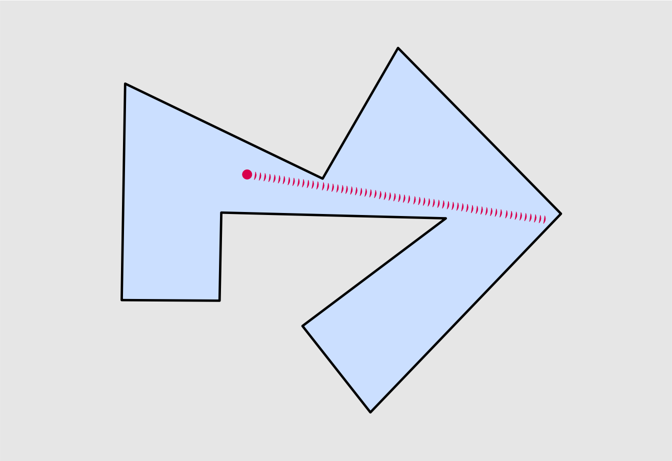
Placing a transmitter for a polygonal building

Introduction



Placing a transmitter for a polygonal building

Introduction



Placing a transmitter for a polygonal building

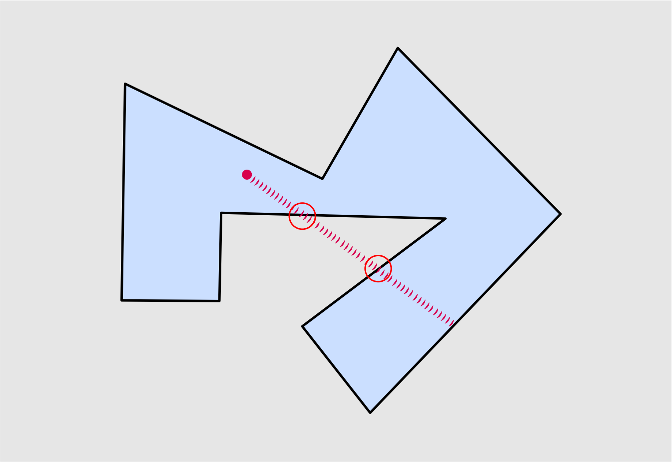
Introduction

- Signals do not bounce off walls...

Introduction

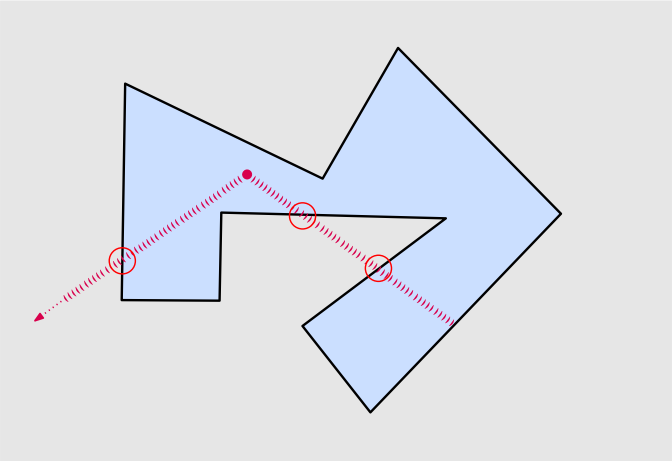
- Signals do not bounce off walls...
- ...but can penetrate up to k walls

Introduction



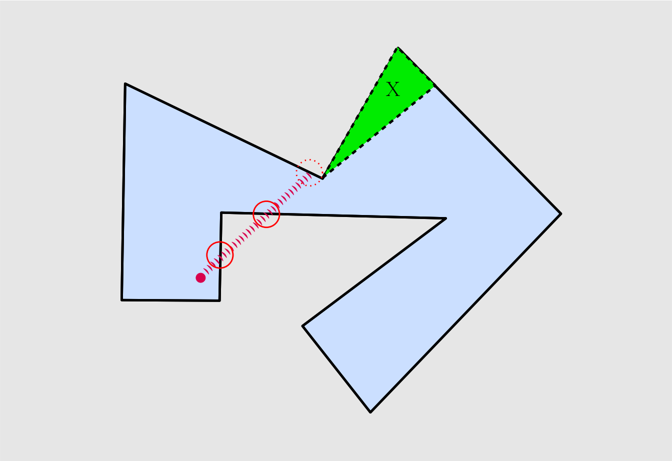
$k = 2$

Introduction



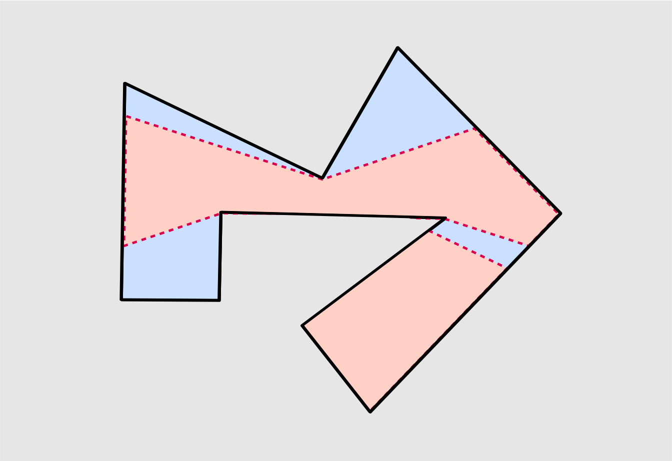
$k = 2$

Introduction



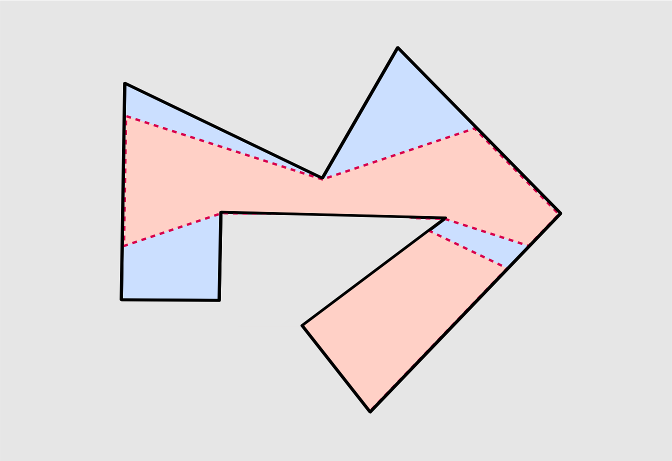
Transmitters must reach entire building

Introduction



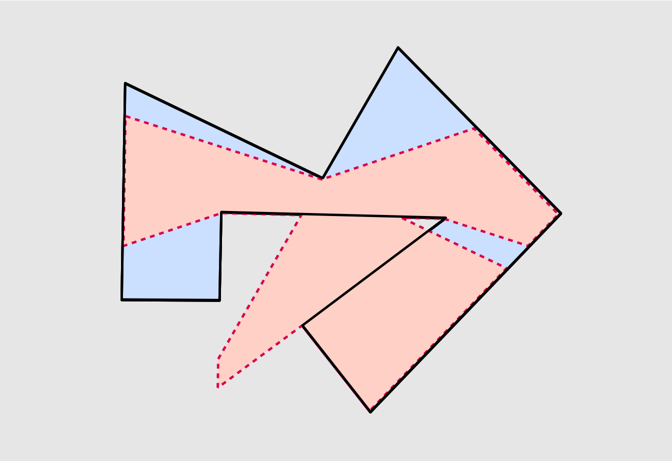
Possible transmitter locations form *k*-kernel

Introduction



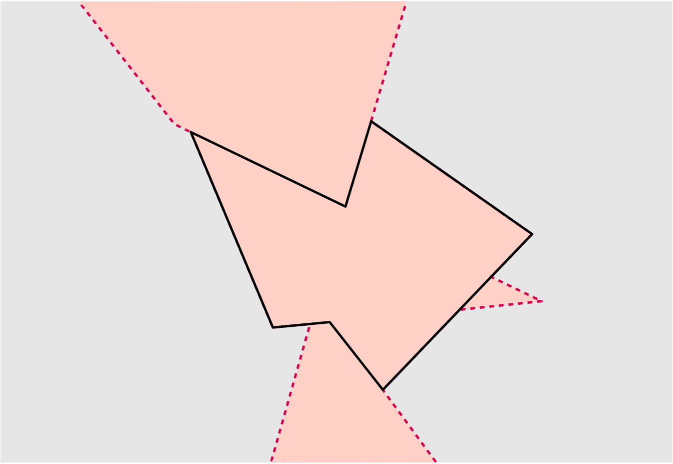
Nonempty *k*-kernel $\Longrightarrow$ polygon is *k*-star-shaped

Introduction



Transmitter can be located outside building

Introduction



Polygon is *k*-convex if it lies within its kernel

Problems

- Complexity of k-kernel

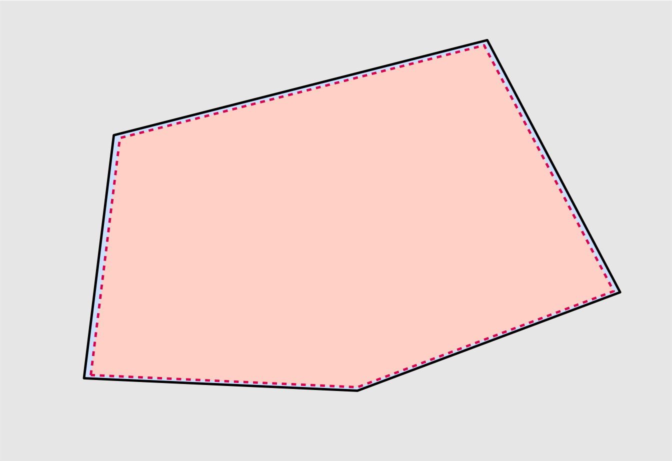
Problems

- Complexity of k-kernel
- Algorithm to construct k-kernel

Problems

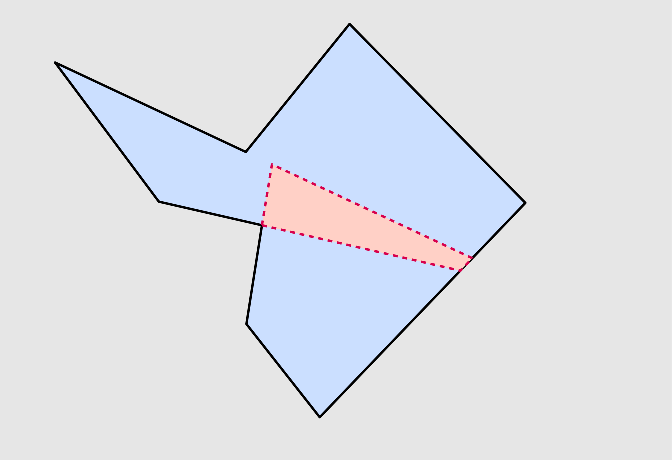
- Complexity of k-kernel
- Algorithm to construct k-kernel
- Algorithm to recognize k-convexity

Related Work: $k = 0$



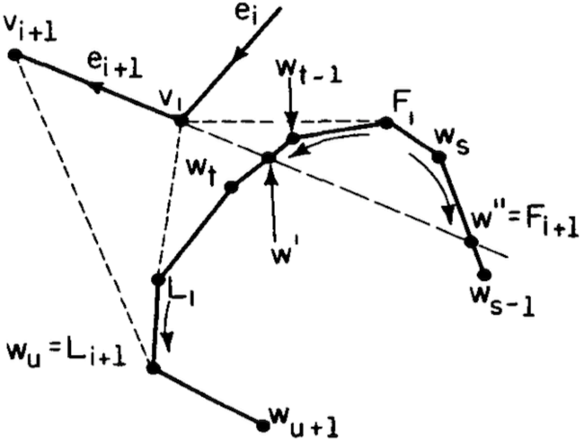
When $k = 0$, standard notion of convexity...

Related Work: $k = 0$



...and standard notion of star-shapedness

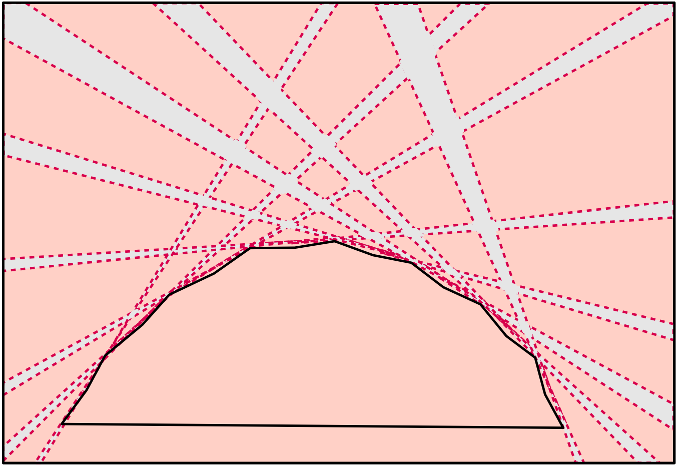
Related Work: $k = 0$



Lee & Preparata, 1979

[Lee & Preparata, 1979] $O(n)$ algorithm for 0-kernel

Related Work: $k = 1$



[Dean et al., 1988] $\Theta(n^2)$ 1-kernel complexity

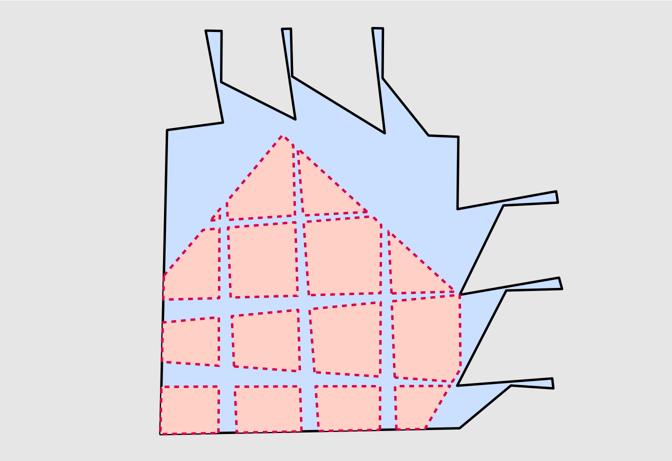
Related Work: $k = 1$

```
for each convex vertex w_k of Q do
  begin
    rename the vertices of Q such that
      (w_k, w_{k+1 mod n}, ..., w_{k+n-1 mod n})
      = (p_0, p_1, ..., p_{n-1});
    find the sequences od(0), od(1), ..., od(n)
      and id(1), ..., id(n);
    for j = 1, ..., n-1 do
    if w_{k+j mod n} is convex then
      begin
        PK(k, k+j mod n) := od(j) ∩ id(j);
        return PK(k, k+j mod n)
      end
  end
```

Dean et al., 1988

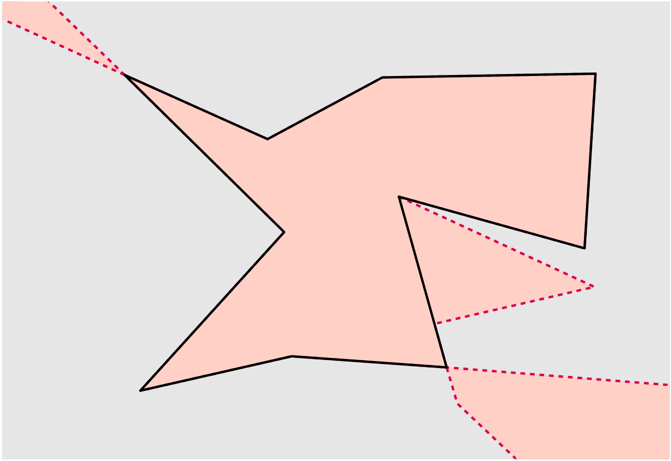
[Dean et al., 1988] $O(n^2)$ algorithm for 1-kernel

Related Work: $k = 2$



[Aicholzer et al., 2009] $\Omega(n^2)$ 2-kernel complexity

Related Work: $k = 2$

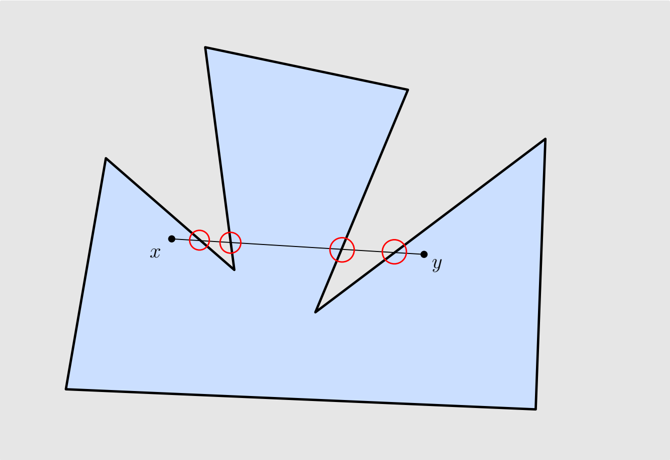


[Aicholzer et al., 2009] $O(n \log n)$ algorithm for recognizing 2-convexity

Related Work: Summary

k	kernel complexity		kernel algorithm	
0	$\Theta(n)$		$O(n)$	[Lee & Preparata, '79]
1	$\Theta(n^2)$	[Dean et al., '88]	$O(n^2)$	[Dean et al., '88]
2	$\Omega(n^2)$	[Aicholzer et al., '09]	?	
≥ 3	?		?	

Properties: k-visibility



x,y are mutually 4-visible

Properties

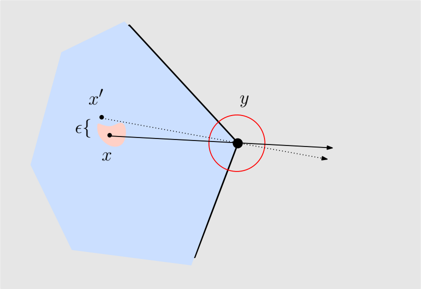
Lemma 1

Points on the boundary of a k-kernel must lie on one of the $\binom{n}{2}$ lines containing two polygon vertices

Properties

Lemma 1

Points on the boundary of a k-kernel must lie on one of the $\binom{n}{2}$ lines containing two polygon vertices



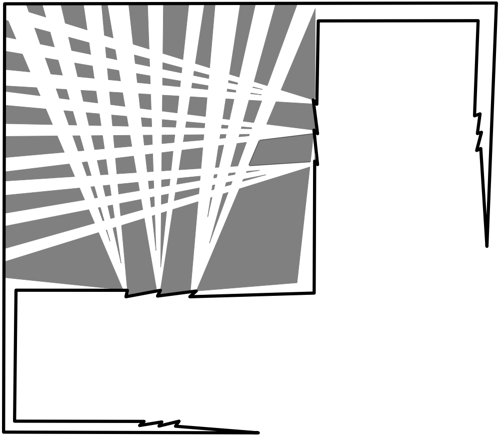
$\overline{xy}$ and $\overline{x'y}$ have same crossing count

Properties

Theorem 2

k-kernels have $O(n^4)$ complexity

Properties



Theorem 3

For $k \geq 4$, some k-kernels have $\Theta(n^4)$ complexity

Properties

k	kernel complexity	kernel algorithm
0	$\Theta(n)$	$O(n)$ [Lee & Preparata, '79]
1	$\Theta(n^2)$ [Dean et al., '88]	$O(n^2)$ [Dean et al., '88]
2	$\Omega(n^2)$ [Aicholzer at el., '09] $O(n^4)$	?
3	$O(n^4)$	?
≥ 4	$\Theta(n^4)$ [Evans & Sember, '10]	?

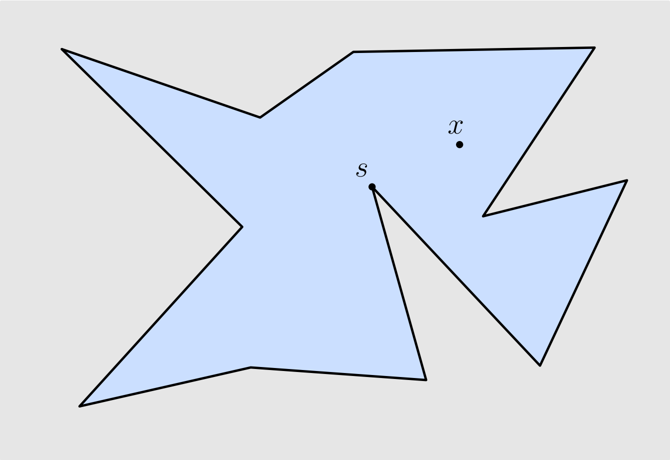
v-regions

- v-region: associated with polygon vertex

v-regions

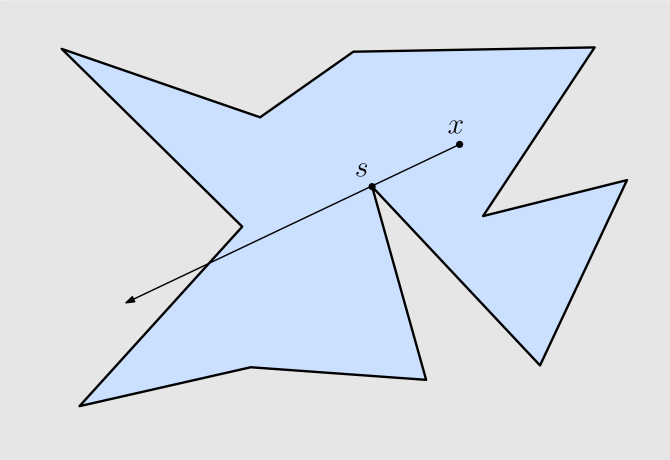
- v-region: associated with polygon vertex
- k-kernel $=$ intersection of v-regions

v-region for vertex s ($k = 2$)



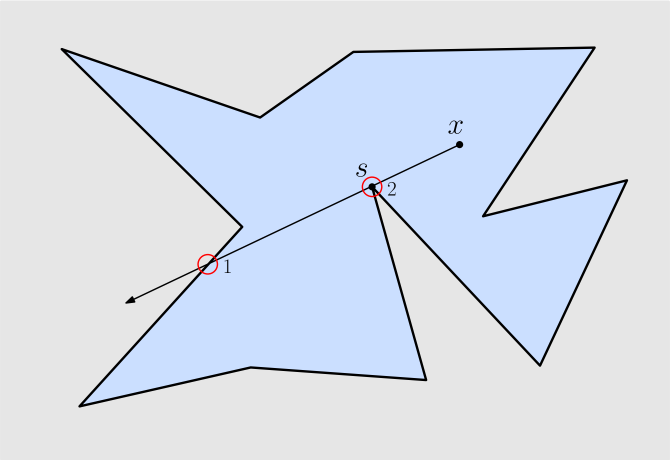
Is x within v-region for s?

v-region for vertex s ($k = 2$)



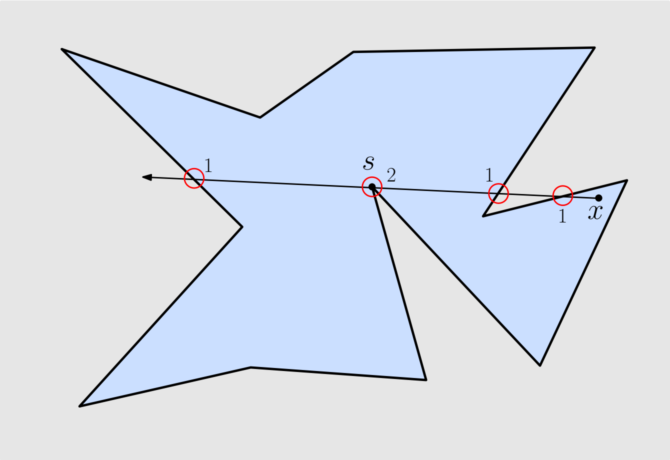
Examine ray $\overrightarrow{xs}$

v-region for vertex s ($k = 2$)



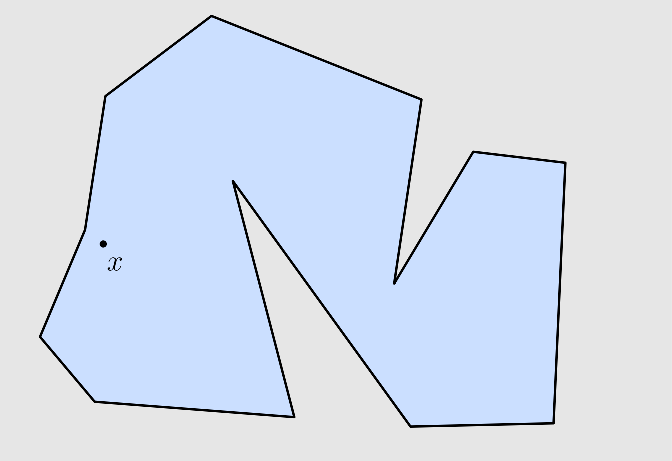
At most $k + 1$ crossings $\implies$ x within v-region

v-region for vertex s ($k = 2$)



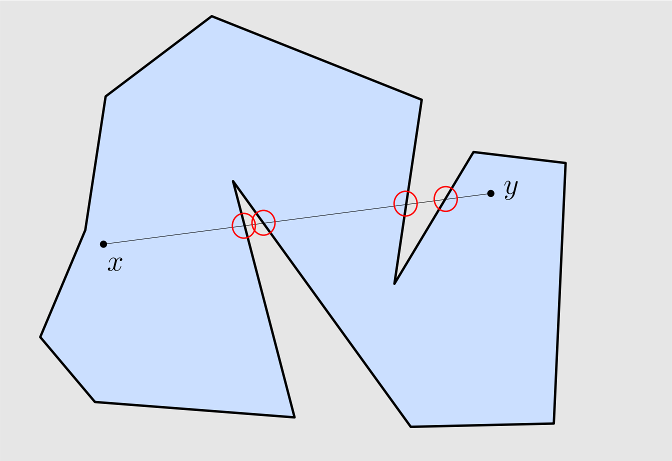
4 crossings $\Longrightarrow$ x not within v-region

Theorem 4



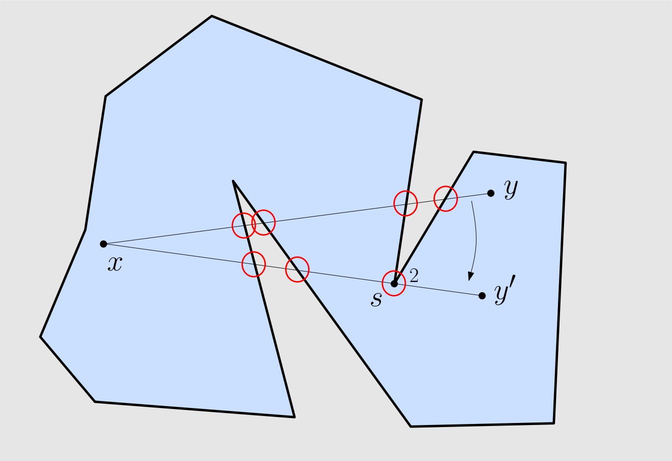
Suppose x is not within 2-kernel

Theorem 4



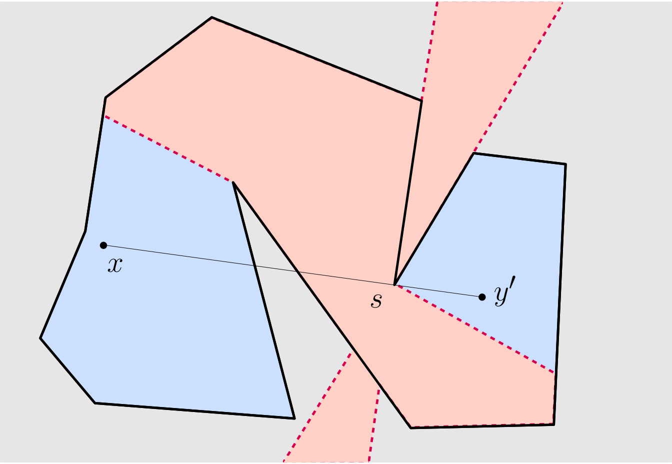
Must exist y such that $\overline{xy}$ has > 2 crossings

Theorem 4



Rotate $\overrightarrow{xy}$ until it contains vertex

Theorem 4



x is not within v-region for vertex

Theorem 4

- x not in kernel $\implies$ x not within some v-region

Theorem 4

- x not in kernel $\implies$ x not within some v-region
- x not in v-region $\implies$ x not within kernel (trivial)

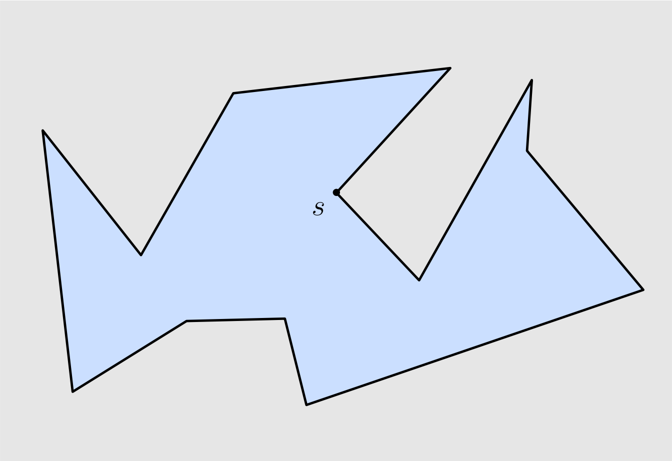
Theorem 4

- x not in kernel $\implies$ x not within some v-region
- x not in v-region $\implies$ x not within kernel (trivial)

Theorem 4

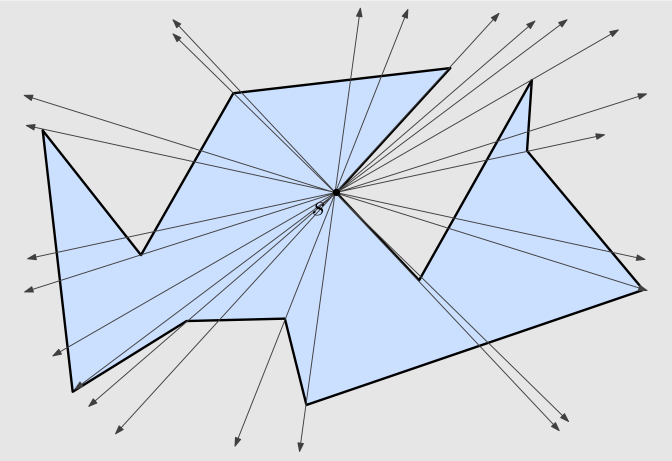
k-kernel is intersection of v-regions

Constructing v-regions



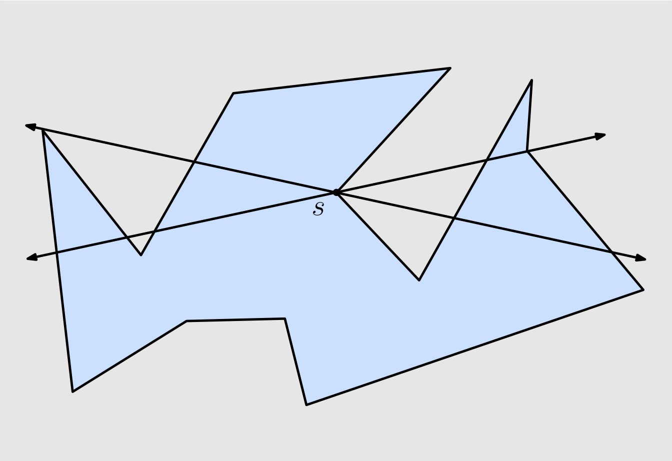
Construct a v-region for a vertex s

Constructing v-regions



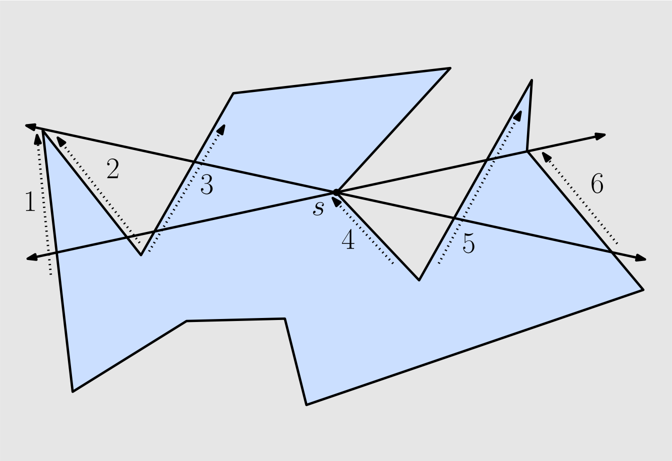
Standard plane sweep technique: sorted radials through vertices

Constructing v-regions



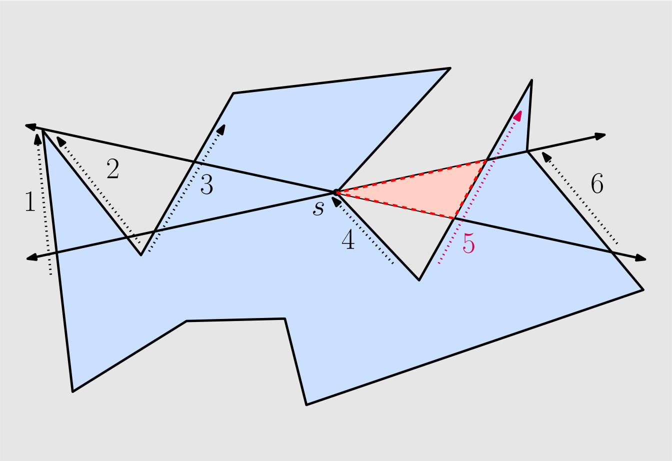
Each pair of radials defines a pair of wedges

Constructing v-regions



Clipping list: the active list of edges crossing the wedges

Constructing v-regions



Edge $k + 2$ is boundary of v-region within wedge

Constructing v-regions

Lemma 6

v-region has $O(n)$ complexity, and can be constructed in $O(n \log n)$ time

Constructing k-kernel

- recall Theorem 4: k-kernel is intersection of v-regions

Constructing k-kernel

- recall Theorem 4: k-kernel is intersection of v-regions
- v-regions are polygons, total of n^2 edges

Constructing k-kernel

- recall Theorem 4: k-kernel is intersection of v-regions
- v-regions are polygons, total of n^2 edges
- let κ be number of intersections between v-region edges

Constructing k-kernel

- recall Theorem 4: k-kernel is intersection of v-regions
- v-regions are polygons, total of n^2 edges
- let κ be number of intersections between v-region edges
- build trapezoidal decomposition in $O(n^2 \log n + \kappa)$ time
  [Chazelle & Edelsbrunner, 1992]

Constructing k-kernel

- recall Theorem 4: k-kernel is intersection of v-regions
- v-regions are polygons, total of n^2 edges
- let κ be number of intersections between v-region edges
- build trapezoidal decomposition in $O(n^2 \log n + \kappa)$ time

  [Chazelle & Edelsbrunner, 1992]
- linear traversal extracts intersection

Constructing k-kernel

Theorem 7

k-kernel can be found in $O(n^2 \log n + \kappa)$ time

Constructing k-kernel

Theorem 7

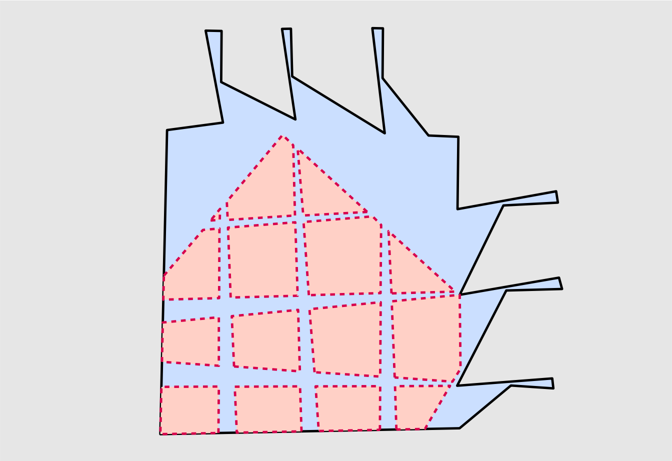
k-kernel can be found in $O(n^2 \log n + \kappa)$ time

- since κ can be $\Theta(n^4)$ (Theorem 3), algorithm is worst-case optimal

Constructing k-kernel

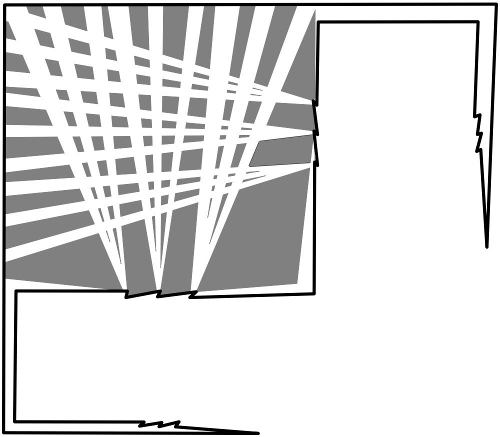
k	kernel complexity	kernel algorithm
0	$\Theta(n)$	$O(n)$ [Lee & Preparata, '79]
1	$\Theta(n^2)$ [Dean et al., '88]	$O(n^2)$ [Dean et al., '88]
2	$\Omega(n^2)$ [Aicholzer at el., '09] $O(n^4)$	$O(n^2 \log n + \kappa)$
3	$O(n^4)$	$O(n^2 \log n + \kappa)$
≥ 4	$\Theta(n^4)$ [Evans & Sember, '10]	$O(n^2 \log n + \kappa)$

Complexity of 2-kernel



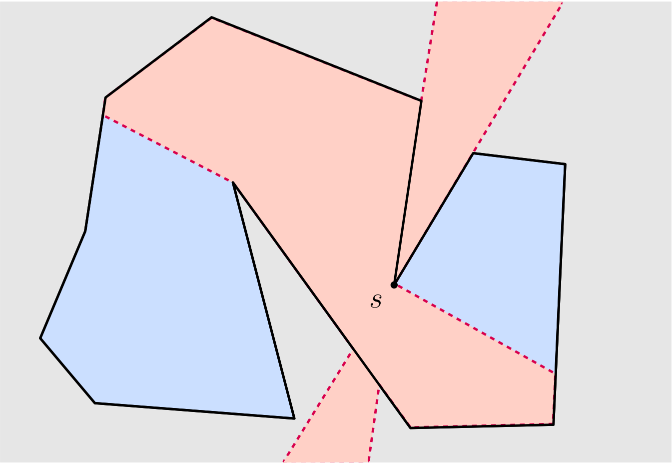
recall complexity can be $\Omega(n^2)$...

Complexity of 2-kernel



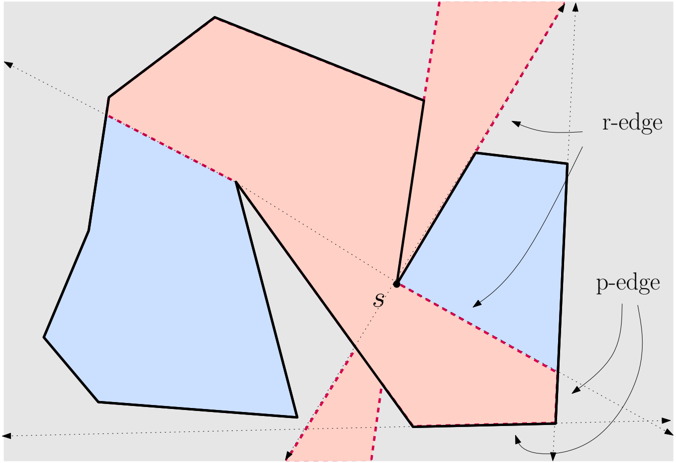
. . . can it be worse?

Complexity of 2-kernel



v-region consists of p-edges and r-edges

Complexity of 2-kernel



v-region consists of p-edges and r-edges

Complexity of 2-kernel

- by Theorem 4, every edge of 2-kernel lies on line containing p-edge or r-edge

Complexity of 2-kernel

- by Theorem 4, every edge of 2-kernel lies on line containing p-edge or r-edge
- complexity of 2-kernel bounded by intersections of these lines

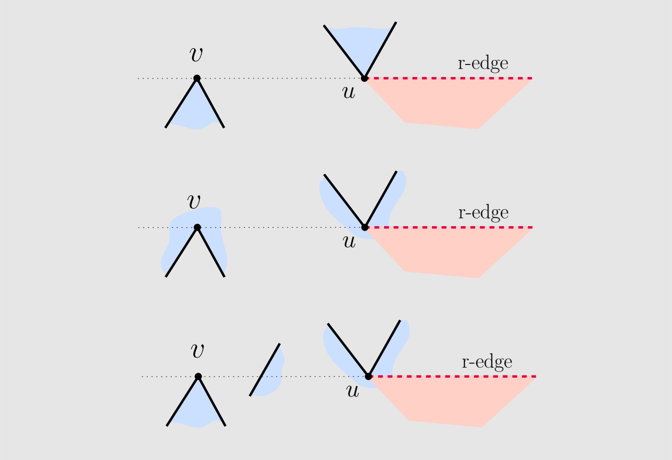
Complexity of 2-kernel

- by Theorem 4, every edge of 2-kernel lies on line containing p-edge or r-edge
- complexity of 2-kernel bounded by intersections of these lines
- $O(n)$ lines containing p-edges

Complexity of 2-kernel

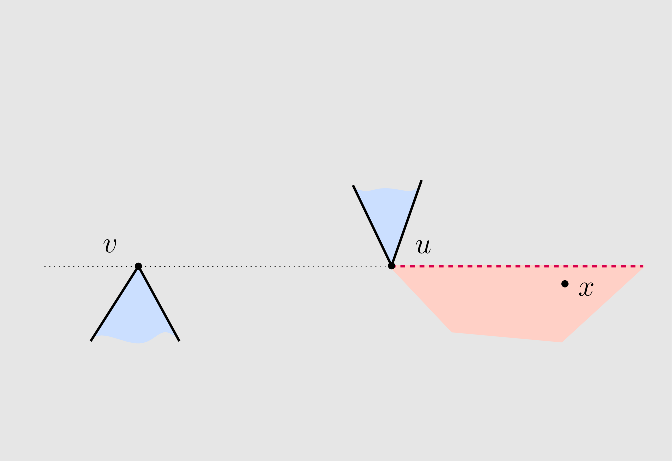
- by Theorem 4, every edge of 2-kernel lies on line containing p-edge or r-edge
- complexity of 2-kernel bounded by intersections of these lines
- $O(n)$ lines containing p-edges
- but how many lines containing r-edges are there?

Complexity of 2-kernel



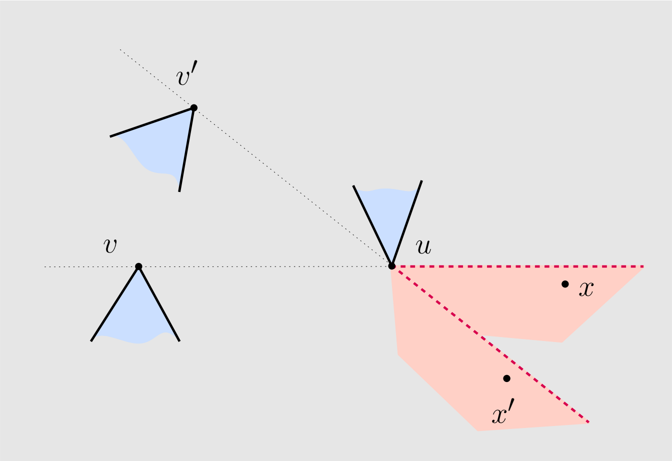
r-edges appear as one of three types (ignoring symmetric cases)

Complexity of 2-kernel



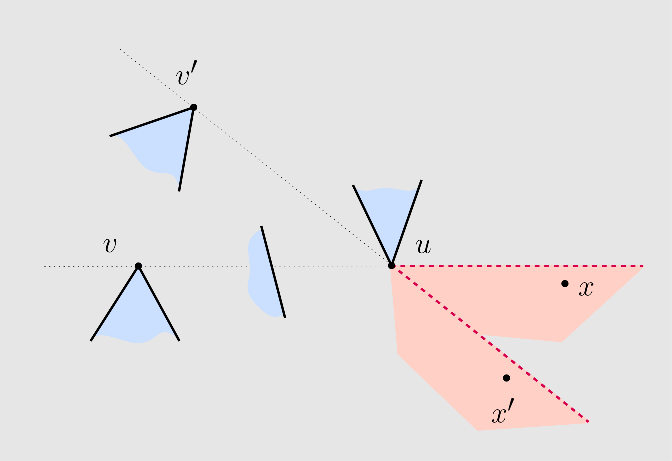
consider first type

Complexity of 2-kernel



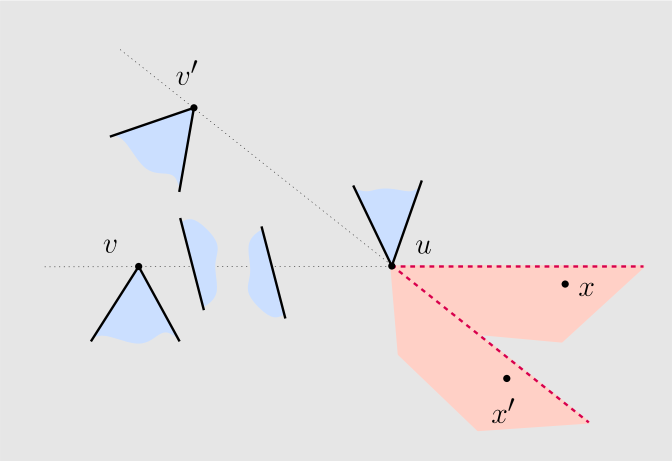
assume u contributes to a second r-edge of this type

Complexity of 2-kernel



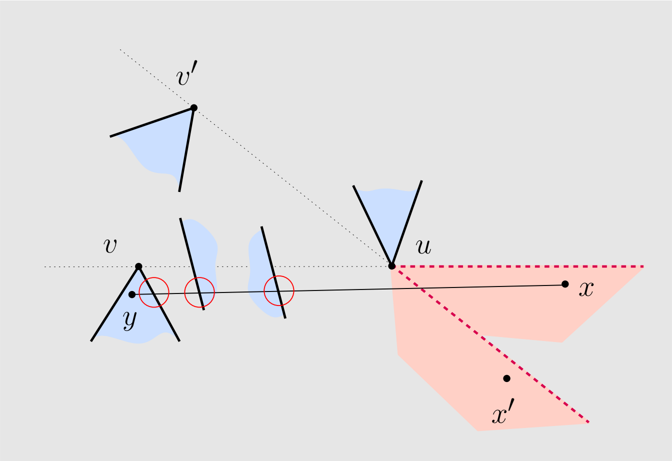
suppose polygon edge crosses ray $\overrightarrow{uv}$

Complexity of 2-kernel



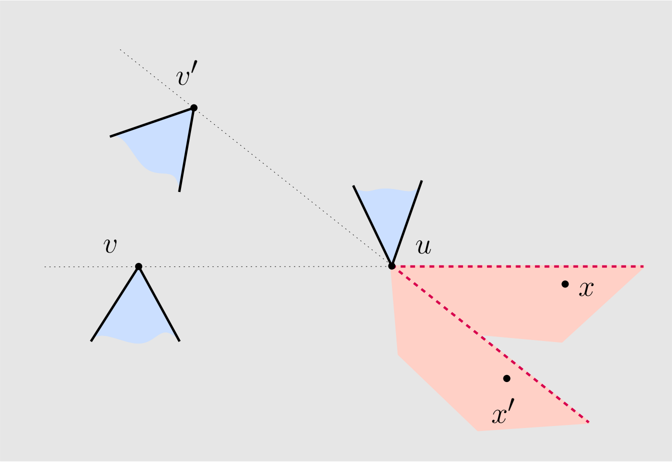
for parity, a second edge must cross the ray as well

Complexity of 2-kernel



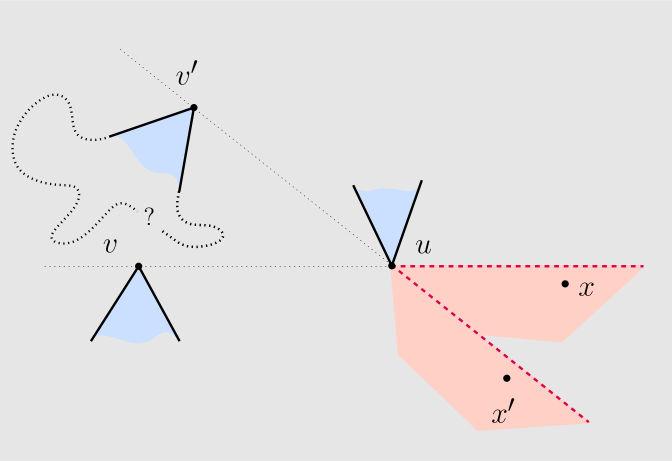
but then some y is not 2-visible from x; contradiction

Complexity of 2-kernel



hence no edges cross $\overrightarrow{uv}$, or $\overrightarrow{uv'}$

Complexity of 2-kernel



v' is disconnected

Complexity of 2-kernel

- a polygon vertex can participate in a single r-edge of type 1

Complexity of 2-kernel

- a polygon vertex can participate in a single r-edge of type 1
- similar results hold for types 2 and 3

Complexity of 2-kernel

- a polygon vertex can participate in a single r-edge of type 1
- similar results hold for types 2 and 3
- $\Longrightarrow$ $O(n)$ r-edges

Complexity of 2-kernel

- a polygon vertex can participate in a single r-edge of type 1
- similar results hold for types 2 and 3
- $\implies$ $O(n)$ r-edges
- $\implies$ the $O(n)$ lines containing p-edges and r-edges intersect $O(n^2)$ times

Complexity of 2-kernel

- a polygon vertex can participate in a single r-edge of type 1
- similar results hold for types 2 and 3
- $\implies O(n)$ r-edges
- $\implies$ the $O(n)$ lines containing p-edges and r-edges intersect $O(n^2)$ times

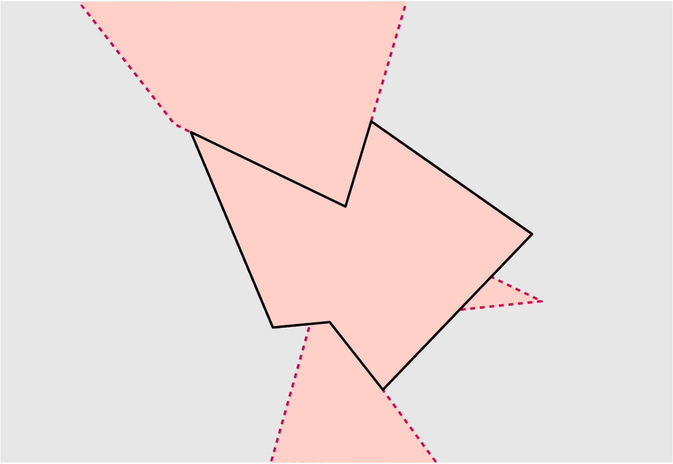
Theorem 8

2-kernels have $O(n^2)$ complexity

Complexity of 2-kernel

k	kernel complexity		kernel algorithm	
0	$\Theta(n)$		$O(n)$	[Lee & Preparata, '79]
1	$\Theta(n^2)$	[Dean et al., '88]	$O(n^2)$	[Dean et al., '88]
2	$\Omega(n^2)$	[Aicholzer at el., '09]	$O(n^2 \log n + \kappa)$	
	$\Theta(n^2)$	[Evans & Sember, '10]		
3	$O(n^4)$		$O(n^2 \log n + \kappa)$	
≥ 4	$\Theta(n^4)$		$O(n^2 \log n + \kappa)$	

Recognizing k-convex polygons



Polygon is *k*-convex if it lies within its kernel

Recognizing k-convex polygons

k	kernel complexity	kernel algorithm	k-convexity test
0	$\Theta(n)$	$O(n)$	$O(n)$
1	$\Theta(n^2)$	$O(n^2)$	$O(n)$ (same test as $k = 0$)
2	$\Omega(n^2)$, $\Theta(n^2)$	$O(n^2 \log n + \kappa)$	$O(n \log n)$ [Lee & Preparata, '79]
3	$O(n^4)$	$O(n^2 \log n + \kappa)$	?
≥ 4	$\Theta(n^4)$	$O(n^2 \log n + \kappa)$	?

Recognizing k-convex polygons

- polygon is k-convex $\iff$ every v-region contains the polygon (Theorem 4)

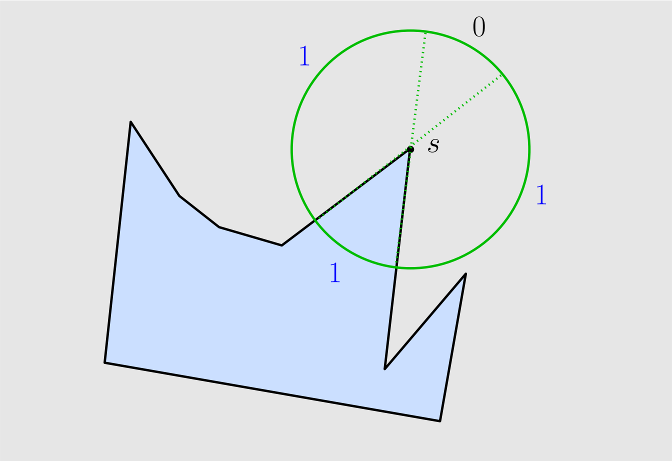
Recognizing k-convex polygons

- polygon is k-convex $\Longleftrightarrow$ every v-region contains the polygon (Theorem 4)
- two algorithms to test v-regions, run in parallel

Recognizing k-convex polygons

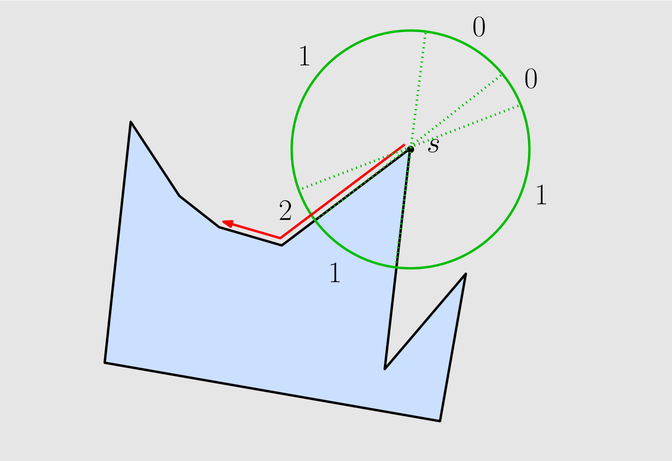
- polygon is k-convex $\iff$ every v-region contains the polygon (Theorem 4)
- two algorithms to test v-regions, run in parallel
- first algorithm is v-region construction algorithm, aborts if active list ever has $> k+2$ edges; $O(n \log n)$ running time

Polygon within v-region (second algorithm)



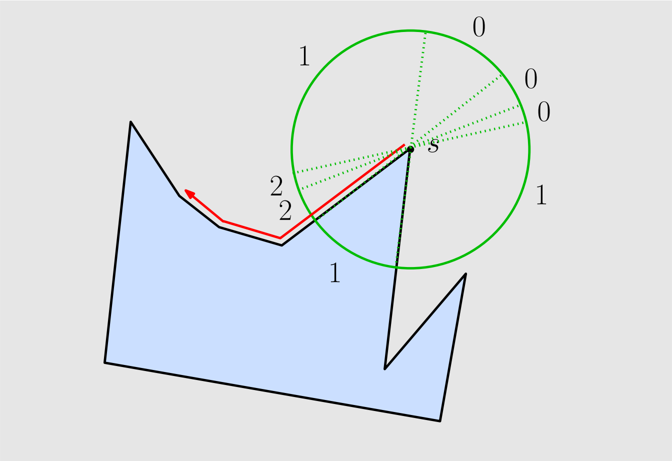
circular list of vertex and edge nodes, with crossing counts

Polygon within v-region (second algorithm)



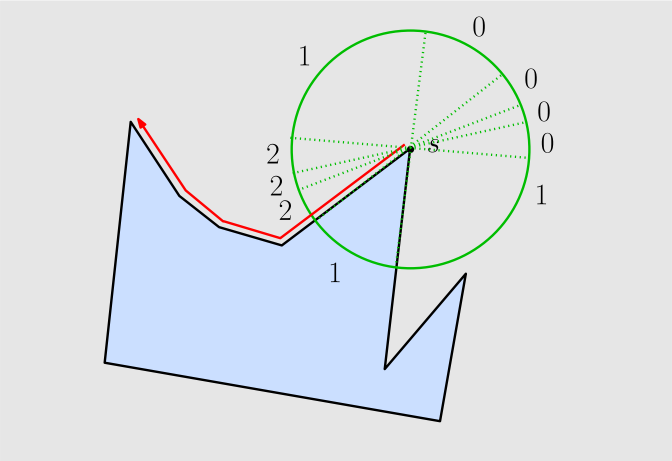
traverse polygon boundary, splitting nodes, incrementing crossing counts

Polygon within v-region (second algorithm)



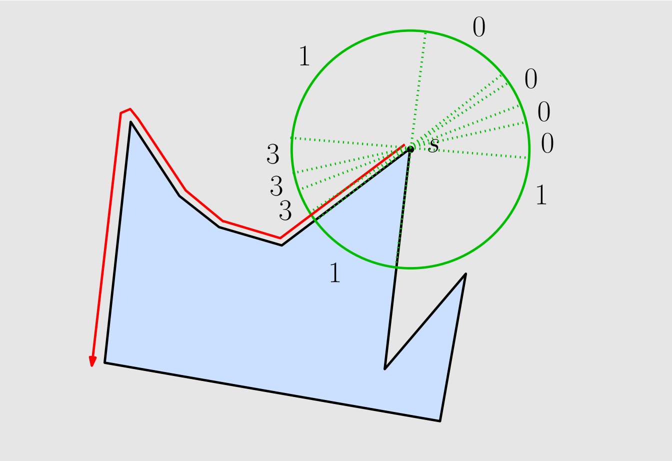
traverse polygon boundary, splitting nodes, incrementing crossing counts

Polygon within v-region (second algorithm)



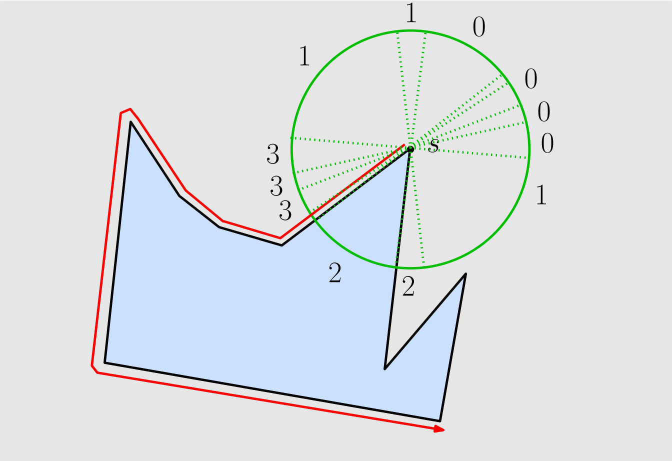
traverse polygon boundary, splitting nodes, incrementing crossing counts

Polygon within v-region (second algorithm)



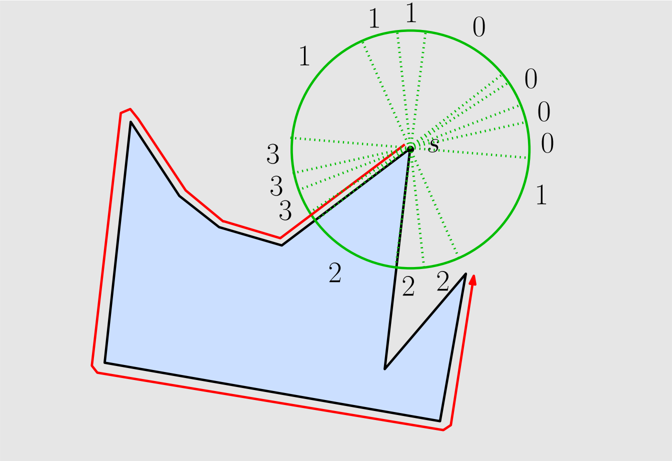
traverse polygon boundary, splitting nodes, incrementing crossing counts

Polygon within v-region (second algorithm)



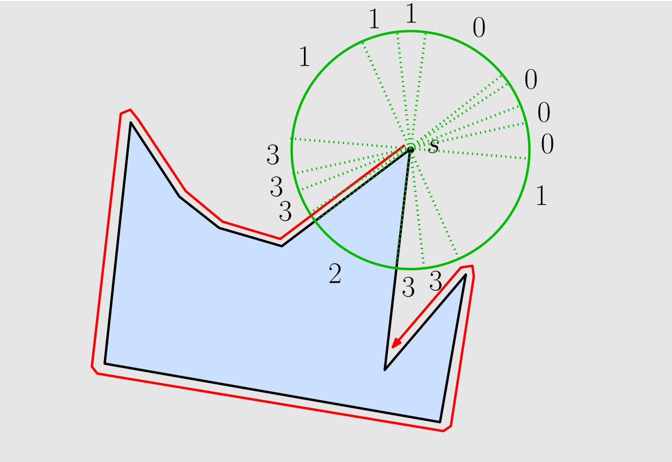
traverse polygon boundary, splitting nodes, incrementing crossing counts

Polygon within v-region (second algorithm)



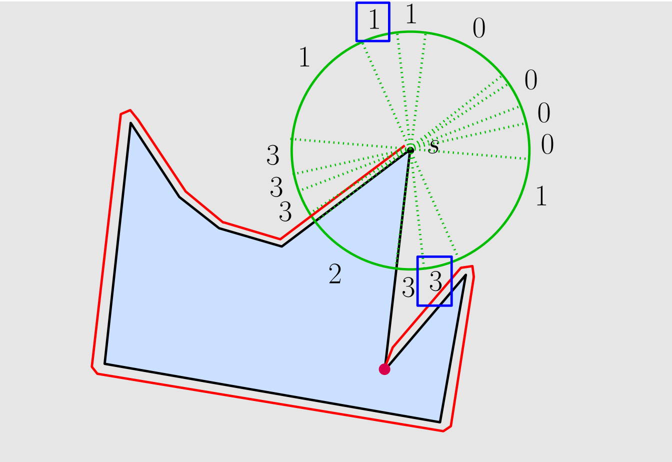
traverse polygon boundary, splitting nodes, incrementing crossing counts

Polygon within v-region (second algorithm)



stop when last vertex is reached

Polygon within v-region (second algorithm)



if sum of antipodal nodes ever exceeds $k+2$, abort

Polygon within v-region (second algorithm)

- each step of algorithm increments some node's crossing count

Polygon within v-region (second algorithm)

- each step of algorithm increments some node's crossing count
- $O(n)$ nodes created in total

Polygon within v-region (second algorithm)

- each step of algorithm increments some node's crossing count
- $O(n)$ nodes created in total
- maximum $O(k)$ visits to each node

Polygon within v-region (second algorithm)

- each step of algorithm increments some node's crossing count
- $O(n)$ nodes created in total
- maximum $O(k)$ visits to each node
- $\implies O(nk)$ running time

Polygon within v-region (second algorithm)

- each step of algorithm increments some node's crossing count
- $O(n)$ nodes created in total
- maximum $O(k)$ visits to each node
- $\implies O(nk)$ running time
- repeat for each of the n vertices

Polygon within v-region (second algorithm)

- each step of algorithm increments some node's crossing count
- $O(n)$ nodes created in total
- maximum $O(k)$ visits to each node
- $\implies O(nk)$ running time
- repeat for each of the n vertices

Theorem 8

k-convexity can be determined in $n^2 \cdot \min(1 + k, \log n)$ time

Results

k	kernel complexity	kernel algorithm	k-convexity test
0	$\Theta(n)$	$O(n)$	$O(n)$
1	$\Theta(n^2)$	$O(n^2)$	$O(n)$ (same test as $k=0$)
2	$\Omega(n^2)$, $\Theta(n^2)$	$O(n^2 \log n + \kappa)$	$O(n \log n)$ [Lee & Preparata, '79]
3	$O(n^4)$	$O(n^2 \log n + \kappa)$	$O(n^2)$ [Evans & Sember, '10]
≥ 4	$\Theta(n^4)$	$O(n^2 \log n + \kappa)$	$O(n^2 \cdot \min(1+k, \log n))$

Future Work

- prove that κ is $O(n^2)$ in 2-kernel algorithm

Future Work

- prove that κ is $O(n^2)$ in 2-kernel algorithm
- ...will reduce algorithm time to $O(n^2 \log n)$

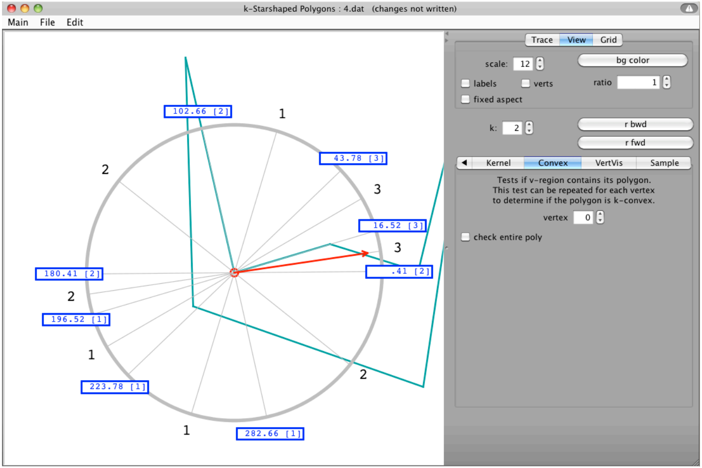
Future Work

- prove that κ is $O(n^2)$ in 2-kernel algorithm
- ...will reduce algorithm time to $O(n^2 \log n)$
- tighten $O(n^4)$ bound on size of 3-kernel

Future Work

- prove that κ is $O(n^2)$ in 2-kernel algorithm
- ...will reduce algorithm time to $O(n^2 \log n)$
- tighten $O(n^4)$ bound on size of 3-kernel
- ...will reduce algorithm time for 3-kernel

Additional Resources



Java Applet available at: www.cs.ubc.ca/~jpsember