Computational Geometry • Lecture

Duality of Points and Lines

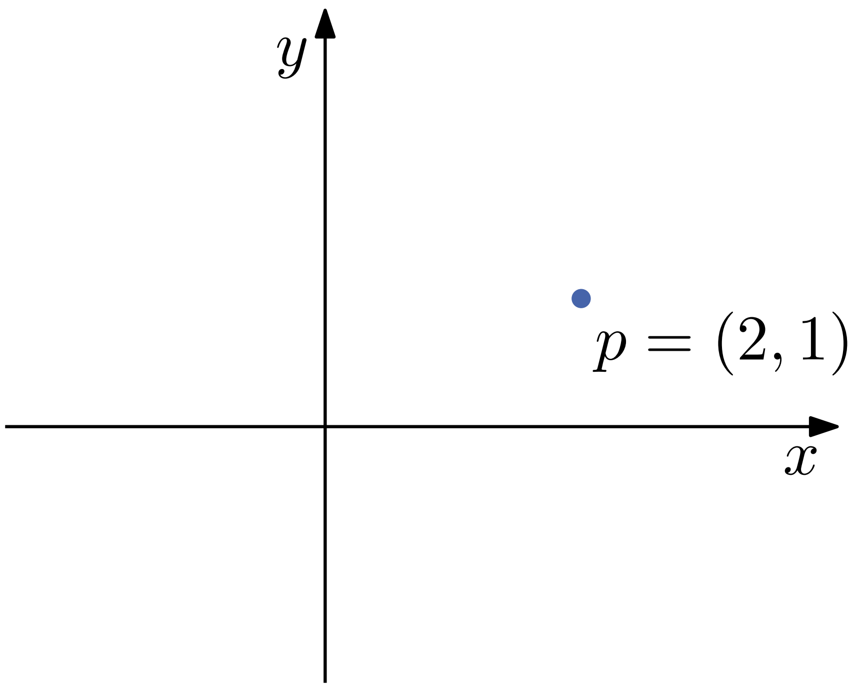
INSTITUTE FOR THEORETICAL INFORMATICS · FACULTY OF INFORMATICS

Tamara Mchedlidze

13.7.2018

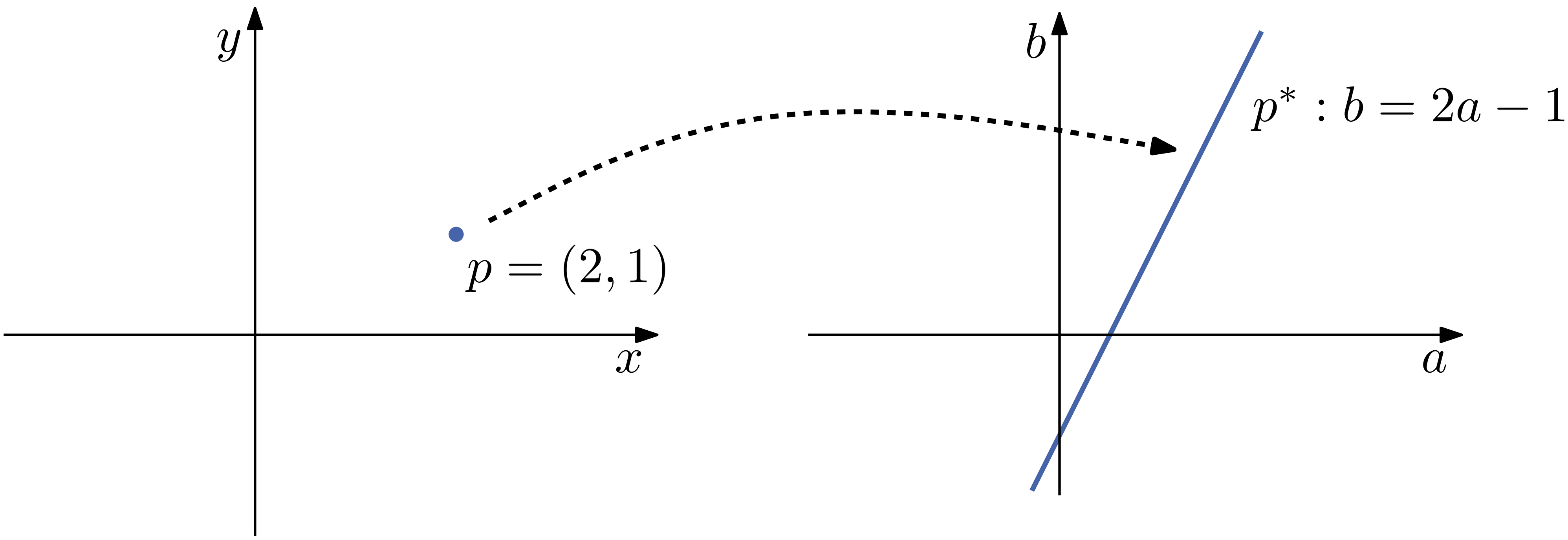
Duality Transforms

We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.



Duality Transforms

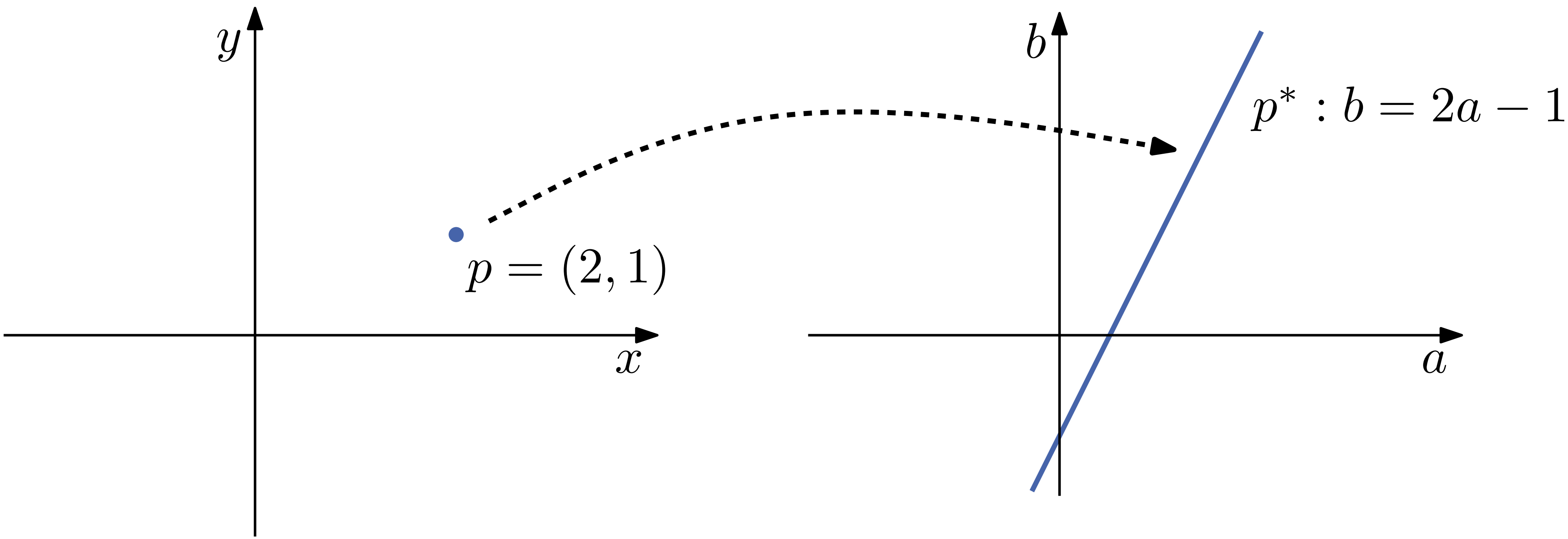
We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.



Duality Transforms

Karlsruhe Institute of Technology

We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.

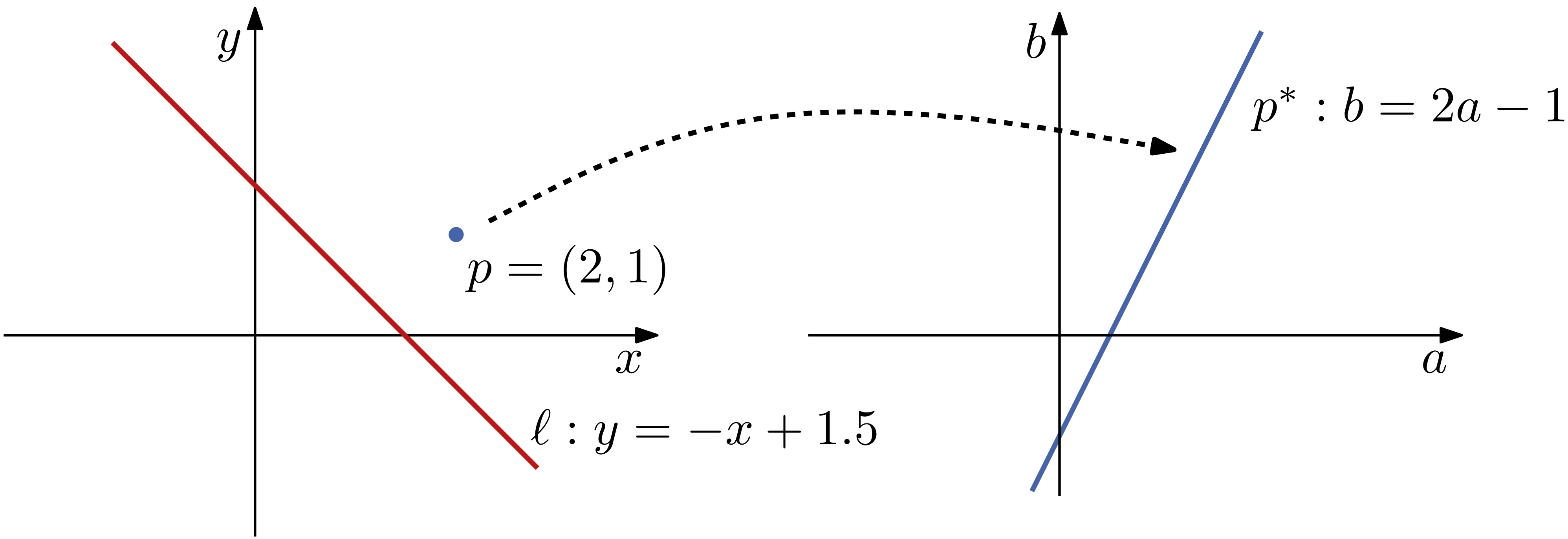


Def: The duality transform $(\cdot)^*$ is defined by

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

Duality Transforms

We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.

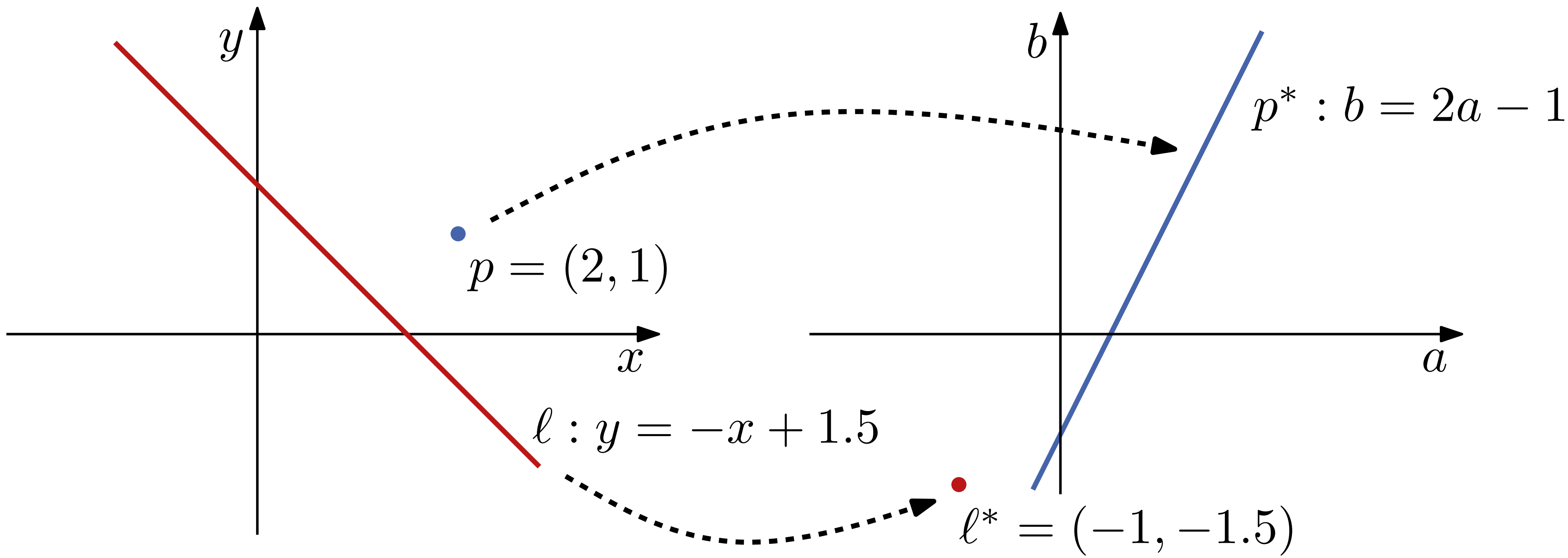


Def: The duality transform $(\cdot)^*$ is defined by

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

Duality Transforms

We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.

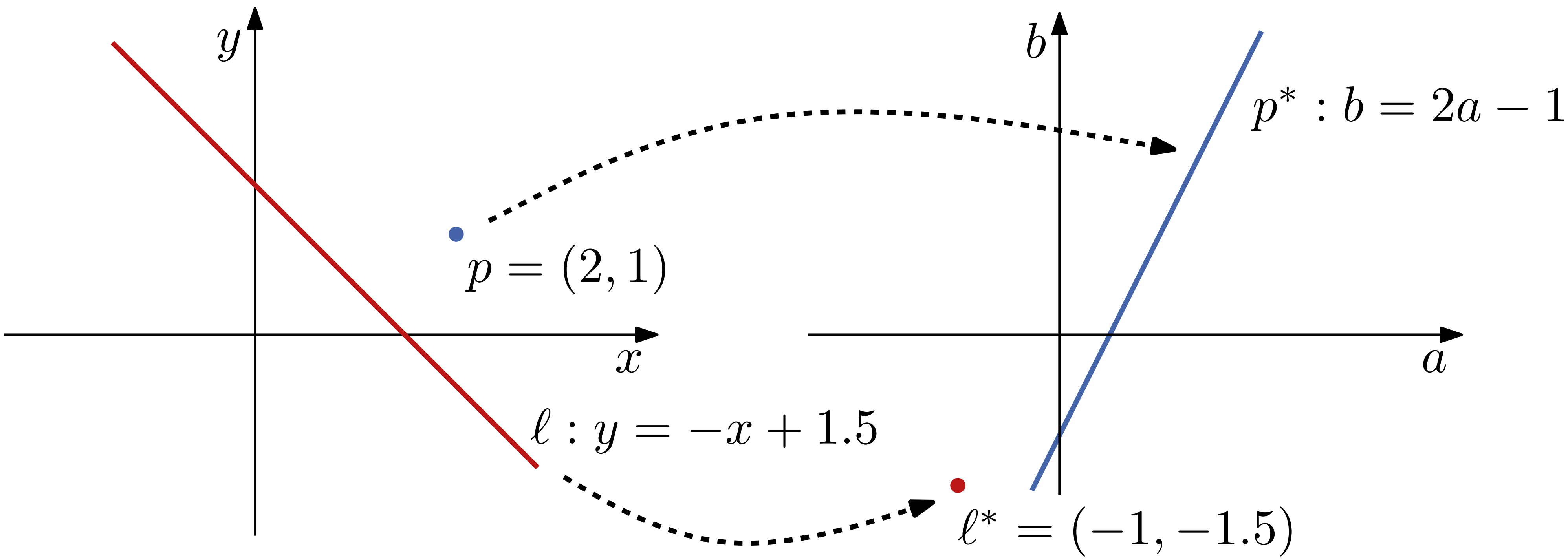


Def: The duality transform $(\cdot)^*$ is defined by

$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

Duality Transforms

We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.



Def: The duality transform $(\cdot)^*$ is defined by

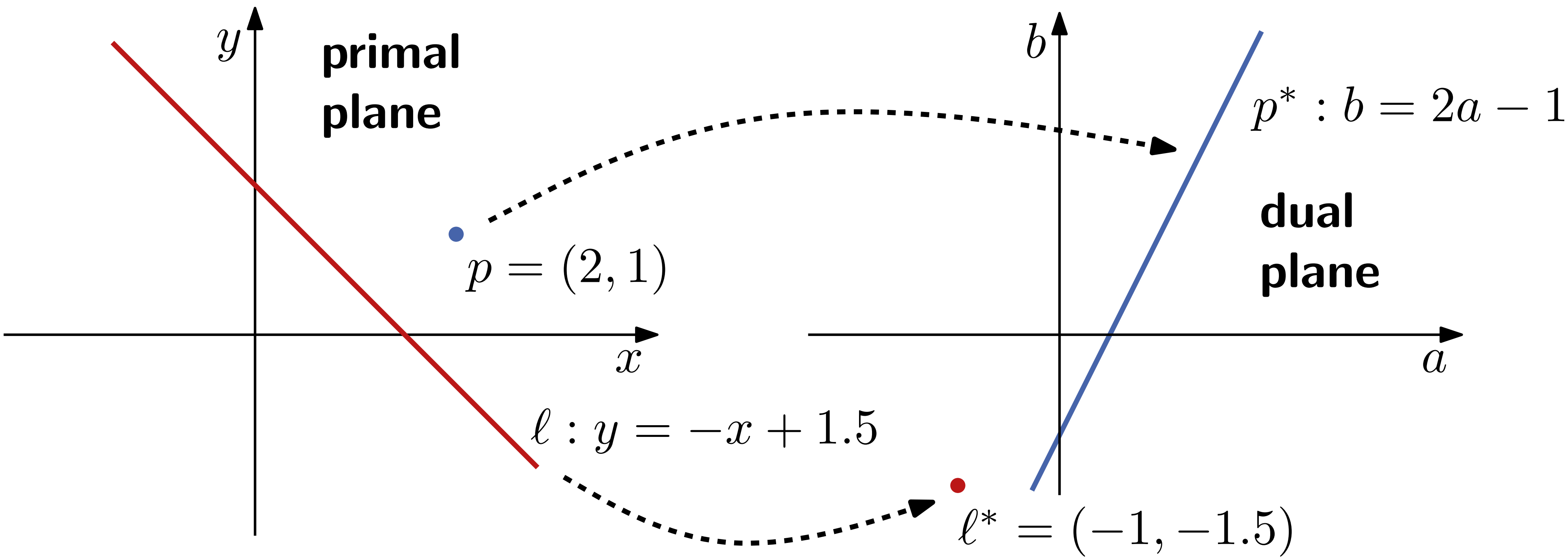
$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

$$\ell : y = mx + c \quad \mapsto \quad \ell^* = (m, -c)$$

Duality Transforms

Karlsruhe Institute of Technology

We have seen duality for planar graphs and duality of Voronoi diagrams and Delaunay triangulations. Here we will see a duality of points and lines in $\mathbb{R}^2$.



Def: The duality transform $(\cdot)^*$ is defined by

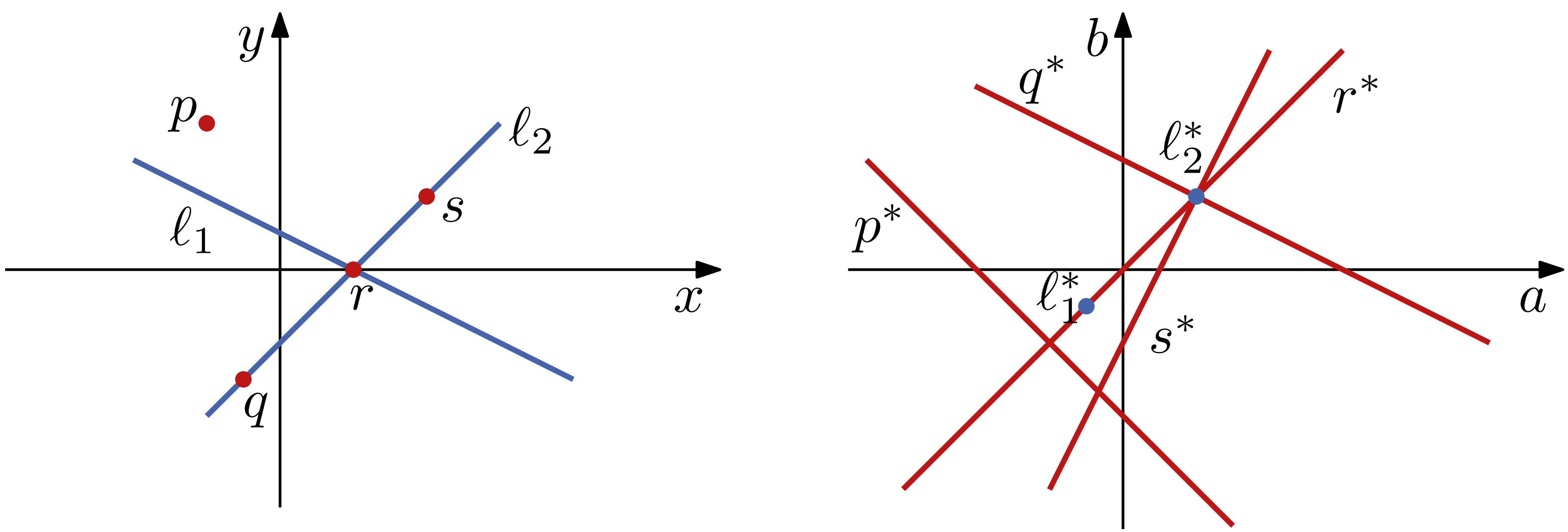
$$p = (p_x, p_y) \quad \mapsto \quad p^* : b = p_x a - p_y$$

$$\ell : y = mx + c \quad \mapsto \quad \ell^* = (m, -c)$$

Properties

Lemma 1: The following properties hold

- $(p^*)^* = p$ and $(\ell^*)^* = \ell$
- p lies below/on/above $\ell \Leftrightarrow p^*$ passes above/through/below ℓ^*
- ℓ_1 and ℓ_2 intersect in point r
  $\Leftrightarrow r^*$ passes through ℓ_1^* and ℓ_2^*
- q, r, s collinear
  $\Leftrightarrow q^*, r^*, s^*$ intersect in a common point



Properties

Lemma 1: The following properties hold

- $(p^*)^* = p$ and $(\ell^*)^* = \ell$
- p lies below/on/above $\ell \Leftrightarrow p^*$ passes above/through/below ℓ^*
- ℓ_1 and ℓ_2 intersect in point r
  $\Leftrightarrow r^*$ passes through ℓ_1^* and ℓ_2^*
- q, r, s collinear
  $\Leftrightarrow q^*, r^*, s^*$ intersect in a common point

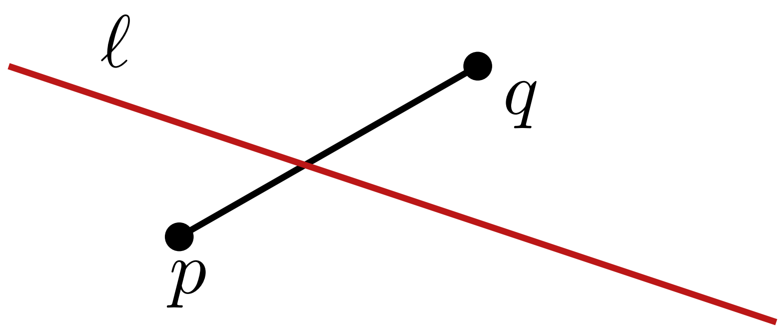
What is the dual object for a line segment $s = \overline{pq}$?
What dual property holds for a line ℓ, intersecting s?

Properties

Lemma 1: The following properties hold

- $(p^*)^* = p$ and $(\ell^*)^* = \ell$
- p lies below/on/above $\ell \Leftrightarrow p^*$ passes above/through/below ℓ^*
- ℓ_1 and ℓ_2 intersect in point r
  $\Leftrightarrow r^*$ passes through ℓ_1^* and ℓ_2^*
- q, r, s collinear
  $\Leftrightarrow q^*, r^*, s^*$ intersect in a common point

What is the dual object for a line segment $s = \overline{pq}$?
What dual property holds for a line ℓ, intersecting s?

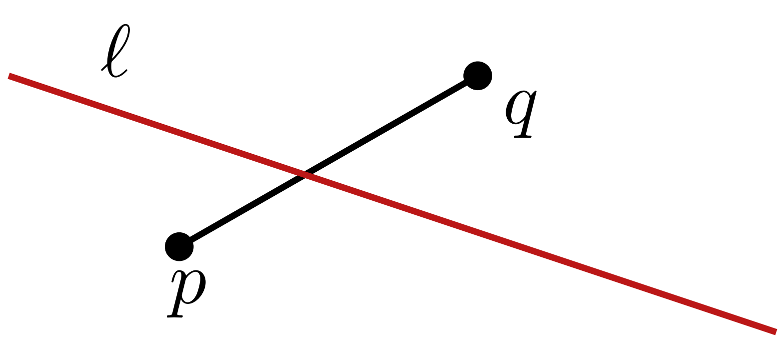


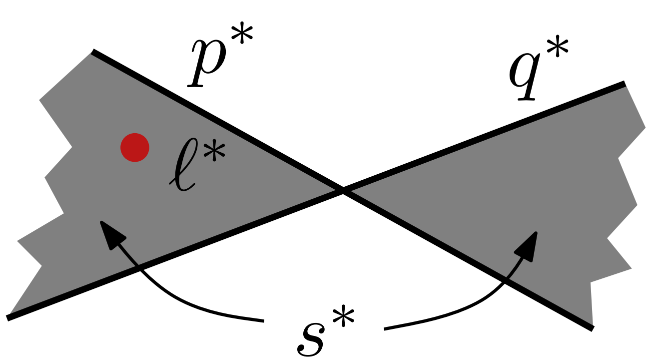
Properties

Lemma 1: The following properties hold

- $(p^*)^* = p$ and $(\ell^*)^* = \ell$
- p lies below/on/above $\ell \Leftrightarrow p^*$ passes above/through/below ℓ^*
- ℓ_1 and ℓ_2 intersect in point r
  $\Leftrightarrow r^*$ passes through ℓ_1^* and ℓ_2^*
- q, r, s collinear
  $\Leftrightarrow q^*, r^*, s^*$ intersect in a common point

What is the dual object for a line segment $s = \overline{pq}$?
What dual property holds for a line ℓ, intersecting s?





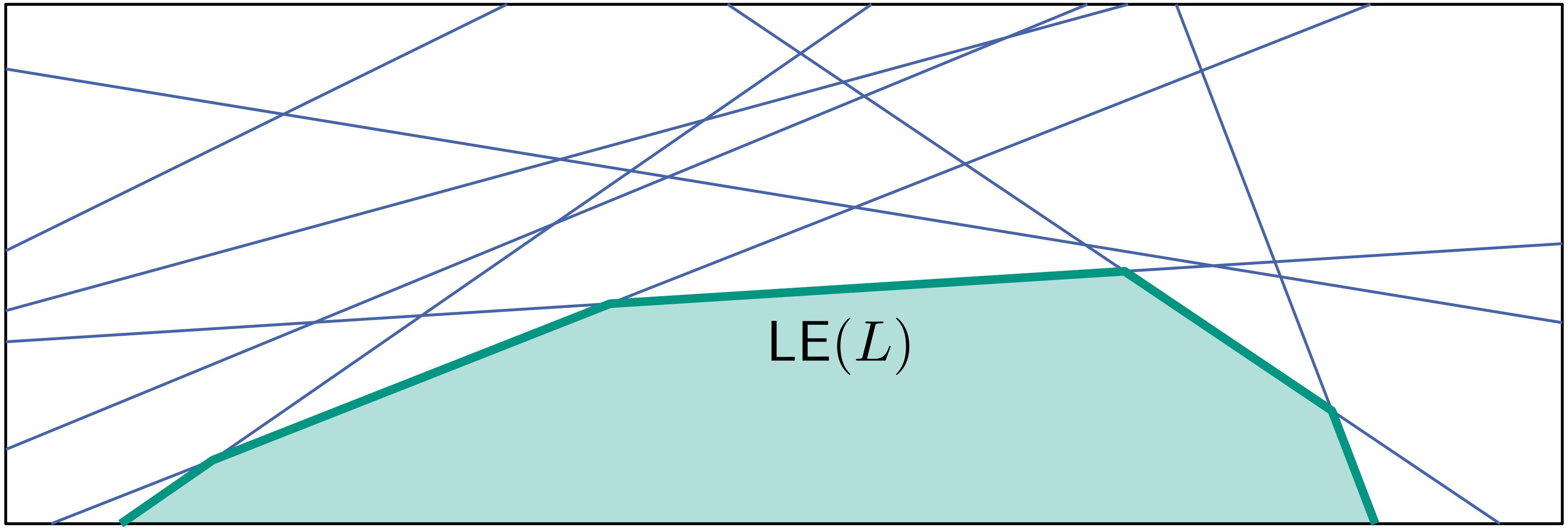
Applications of Duality

Duality does not make geometric problems easier or harder; it simply provides a different (but often helpful) perspective!

We will look at two examples in more detail:

- upper/lower envelopes of line arrangements
- minimum-area triangle in a point set

Lower Envelope

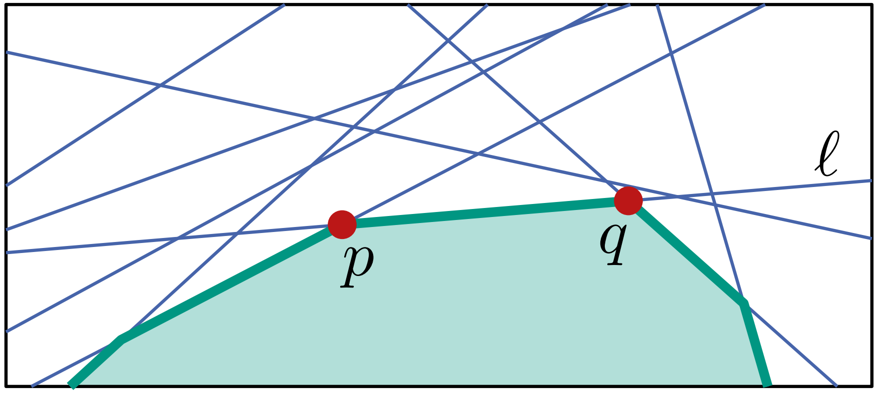


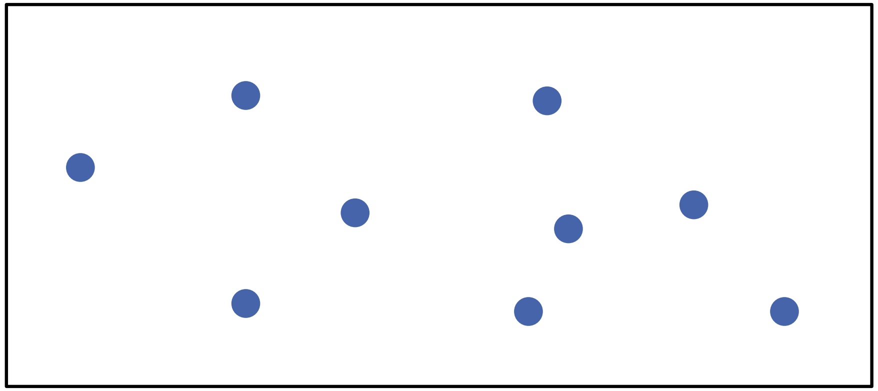
Def: For a set L of lines the **lower envelope** LE(L) of L is the set of all points in $\cup_{\ell \in L} \ell$ that are not above any line in the set L (boundary of the intersection of all lower halfplanes).

Several possibilities for computing lower envelopes

- divide&conquer or sweep-line half-plane intersection algorithms (see Chapter 4.2 in [BCKO08])
- consider the dual problem for $L^* = \{\ell^* \mid \ell \in L\}$

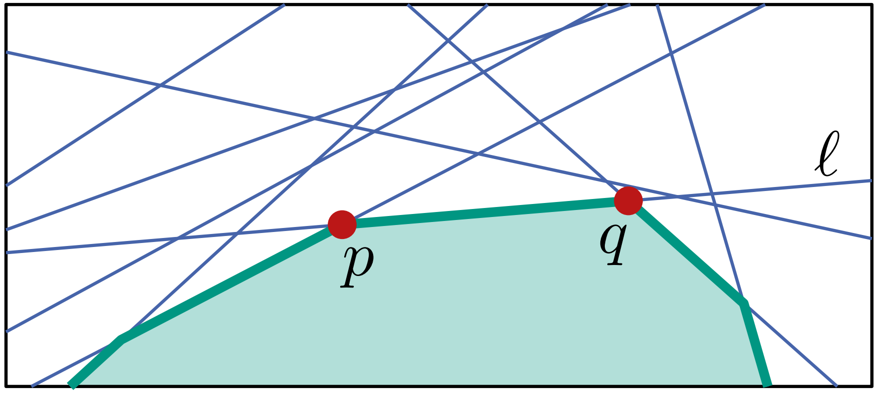
Envelopes and Duality

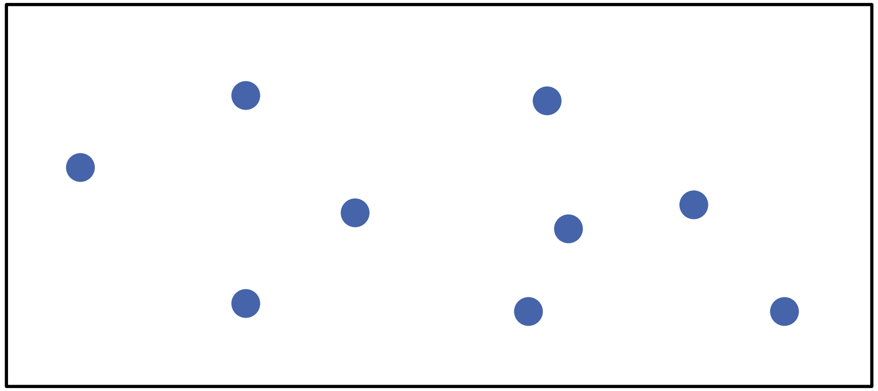




When does an edge $\overline{pq}$ of ℓ appear as a segment on $\mathsf{LE}(L)$?

Envelopes and Duality

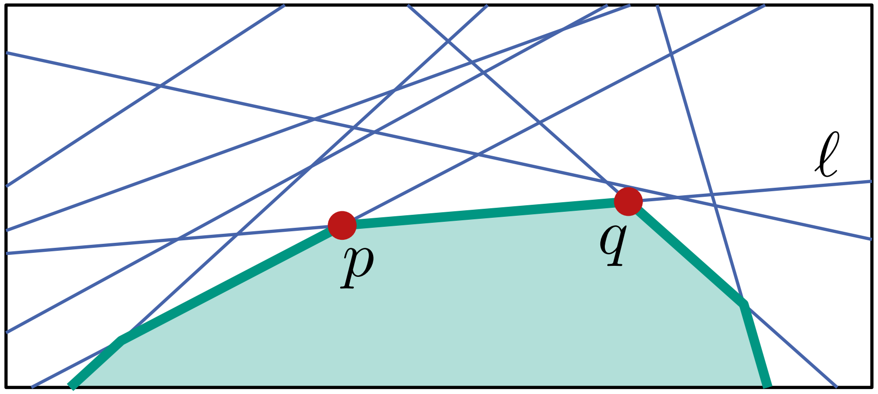


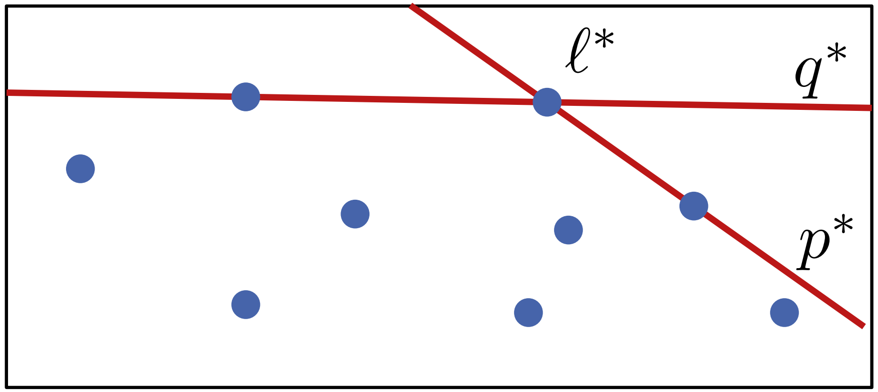


When does an edge $\overline{pq}$ of ℓ appear as a segment on LE(L)?

- p and q are not above any line in L

Envelopes and Duality

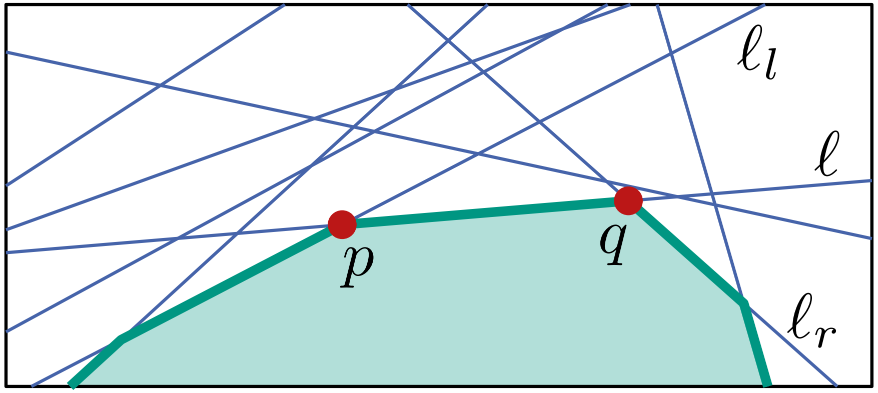


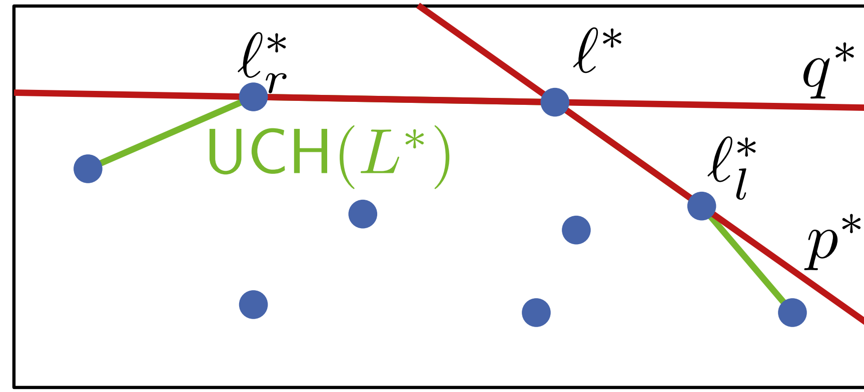


When does an edge $\overline{pq}$ of ℓ appear as a segment on LE(L)?

- p and q are not above any line in L
- p^* and q^* are not below any point in L^*

Envelopes and Duality

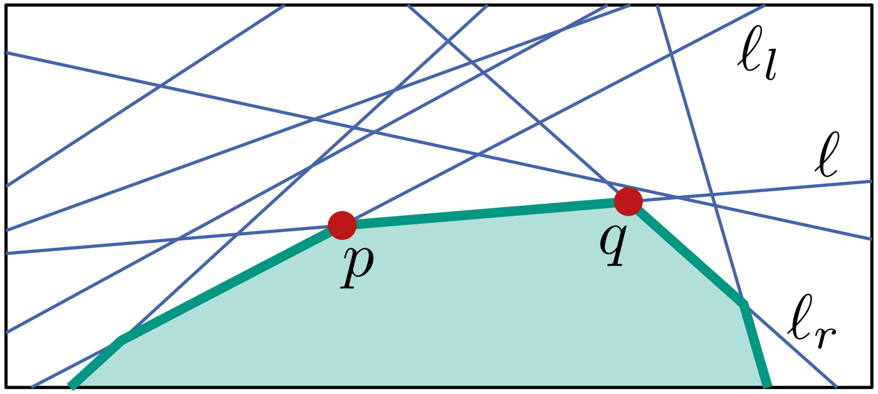


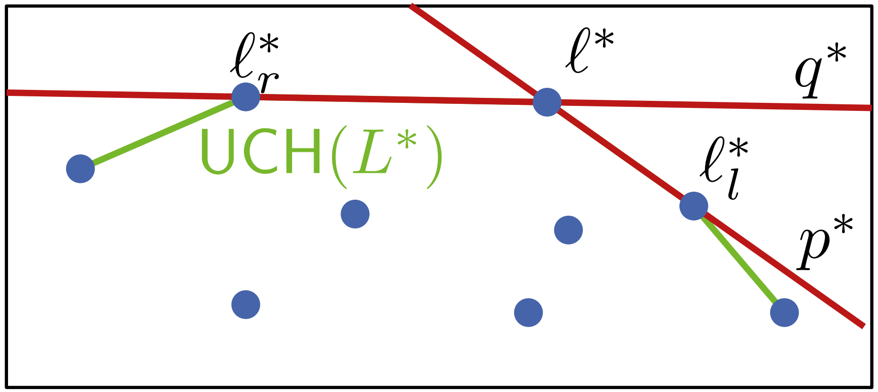


When does an edge $\overline{pq}$ of ℓ appear as a segment on LE(L)?

- p and q are not above any line in L
- p^* and q^* are not below any point in L^*
  $\Rightarrow$ must be neighbors on upper convex hull UCH(L^*)
- intersection point of p^* and q^* is ℓ^*, a vertex of UCH(L^*)

Envelopes and Duality





When does an edge $\overline{pq}$ of ℓ appear as a segment on LE(L)?

- p and q are not above any line in L
- p^* and q^* are not below any point in L^*
  $\Rightarrow$ must be neighbors on upper convex hull UCH(L^*)
- intersection point of p^* and q^* is ℓ^*, a vertex of UCH(L^*)

Lemma 2: The lines on LE(L) ordered from right to left correspond to the vertices of UCH(L^*) ordered from left to right.

Computing the Envelope

- algorithm for computing upper convex hull in time $O(n \log n)$ (see Lecture 1 on convex hulls)

Computing the Envelope

- algorithm for computing upper convex hull in time $O(n \log n)$ (see Lecture 1 on convex hulls)

- primal lines of the points on UCH(L^*) in reverse order form LE(L)

Computing the Envelope

- algorithm for computing upper convex hull in time $O(n \log n)$ (see Lecture 1 on convex hulls)

- primal lines of the points on UCH(L^*) in reverse order form LE(L)

- analogously: upper envelope of $L \hat{=}$ lower convex hull of L^*

Computing the Envelope

- algorithm for computing upper convex hull in time $O(n \log n)$ (see Lecture 1 on convex hulls)

- primal lines of the points on UCH(L^*) in reverse order form LE(L)

- analogously: upper envelope of $L \hat{=}$ lower convex hull of L^*

When does this approach work faster?

Computing the Envelope

- algorithm for computing upper convex hull in time $O(n \log n)$ (see Lecture 1 on convex hulls)

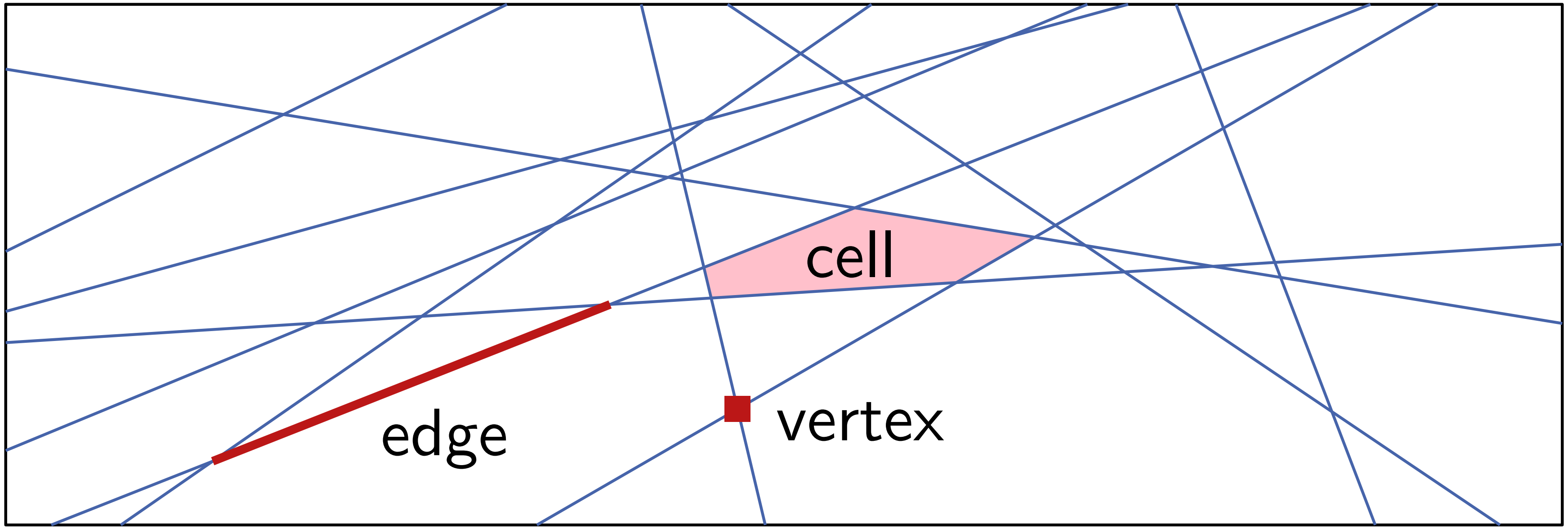
- primal lines of the points on UCH(L^*) in reverse order form LE(L)

- analogously: upper envelope of $L \hat{=}$ lower convex hull of L^*

When does this approach work faster?

- output sensitive algorithm for computing convex hull with h points with time complexity $O(n \log h)$

Line Arrangements



Def: A set L of lines defines a subdivision $\mathcal{A}(L)$ of the plane (the **line arrangement**) composed of vertices, edges, and cells (poss. unbounded).
$\mathcal{A}(L)$ is called **simple** if no three lines share a point and no two lines are parallel.

Complexity of $\mathcal{A}(L)$

The combinatorial complexity of $\mathcal{A}(L)$ is the number of vertices, edges, and cells.

Theorem 1: Let $\mathcal{A}(L)$ be a simple line arrangement for n lines. Then $\mathcal{A}(L)$ has $\binom{n}{2}$ vertices, n^2 edges, and $n^2/2 + n/2 + 1$ cells.

Complexity of $\mathcal{A}(L)$

The combinatorial complexity of $\mathcal{A}(L)$ is the number of vertices, edges, and cells.

Theorem 1: Let $\mathcal{A}(L)$ be a simple line arrangement for n lines. Then $\mathcal{A}(L)$ has $\binom{n}{2}$ vertices, n^2 edges, and $n^2/2 + n/2 + 1$ cells.

Data structure for $\mathcal{A}(L)$:

- create bounding box of all vertices (s. exercise) $\rightarrow$ obtain planar embedded Graph G
- doubly-connected edge list for G

Complexity of $\mathcal{A}(L)$

The combinatorial complexity of $\mathcal{A}(L)$ is the number of vertices, edges, and cells.

Theorem 1: Let $\mathcal{A}(L)$ be a simple line arrangement for n lines. Then $\mathcal{A}(L)$ has $\binom{n}{2}$ vertices, n^2 edges, and $n^2/2 + n/2 + 1$ cells.

Data structure for $\mathcal{A}(L)$:

- create bounding box of all vertices (s. exercise) $\rightarrow$ obtain planar embedded Graph G
- doubly-connected edge list for G

Do we already know a way to compute $\mathcal{A}(L)$?

Complexity of $\mathcal{A}(L)$

The combinatorial complexity of $\mathcal{A}(L)$ is the number of vertices, edges, and cells.

Theorem 1: Let $\mathcal{A}(L)$ be a simple line arrangement for n lines. Then $\mathcal{A}(L)$ has $\binom{n}{2}$ vertices, n^2 edges, and $n^2/2 + n/2 + 1$ cells.

Data structure for $\mathcal{A}(L)$:

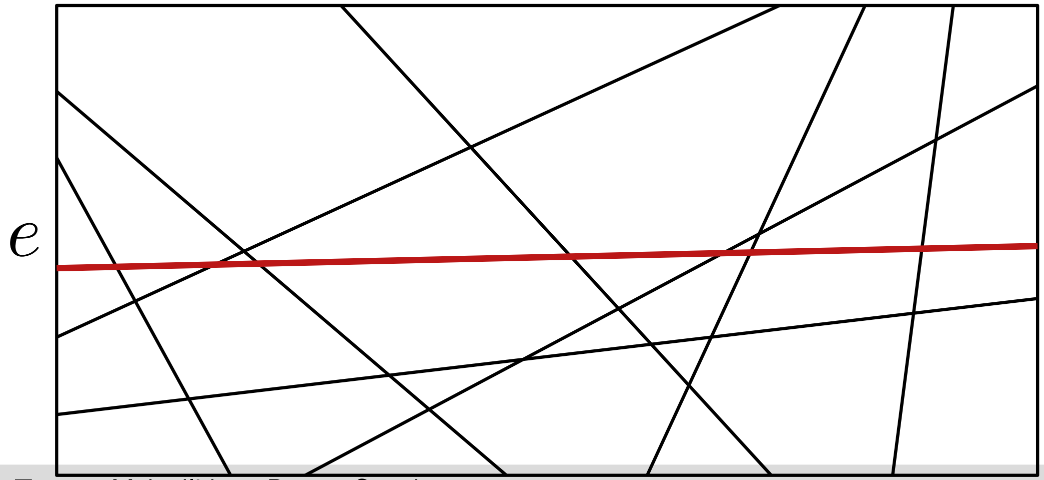
- create bounding box of all vertices (s. exercise)
  $\rightarrow$ obtain planar embedded Graph G
- doubly-connected edge list for G

Do we already know a way to compute $\mathcal{A}(L)$?

$\rightarrow$ could use line segment intersection plane sweep in $O(n^2 \log n)$

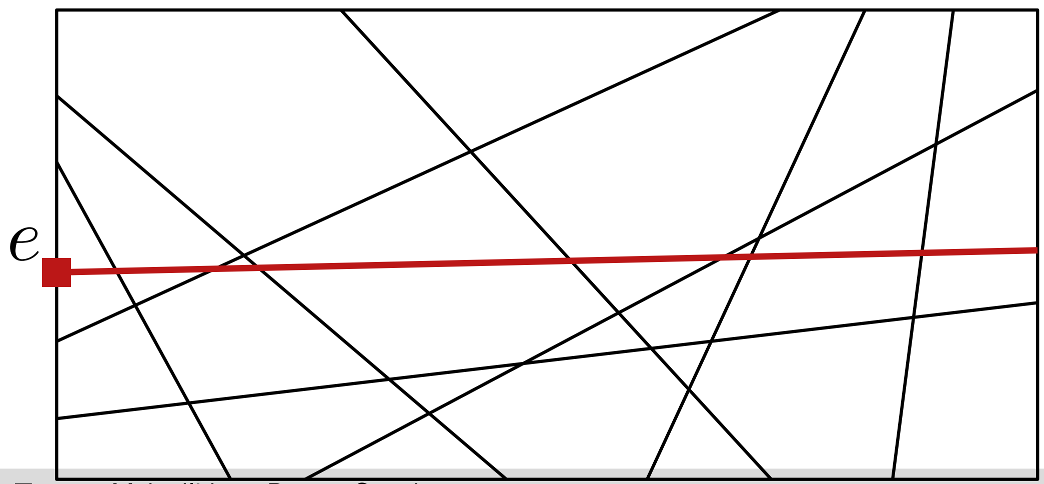
Incrementally Constructing $\mathcal{A}(L)$

Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



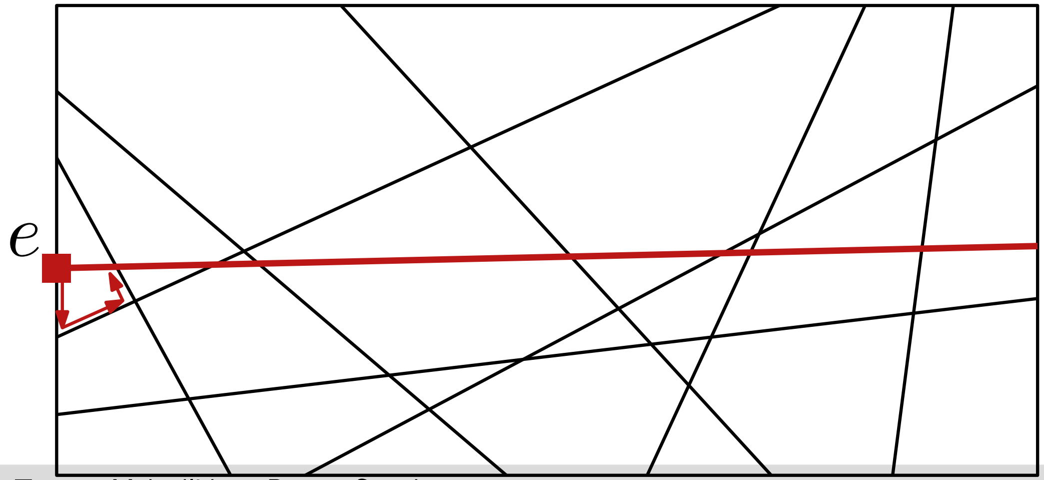
Incrementally Constructing $\mathcal{A}(L)$

Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



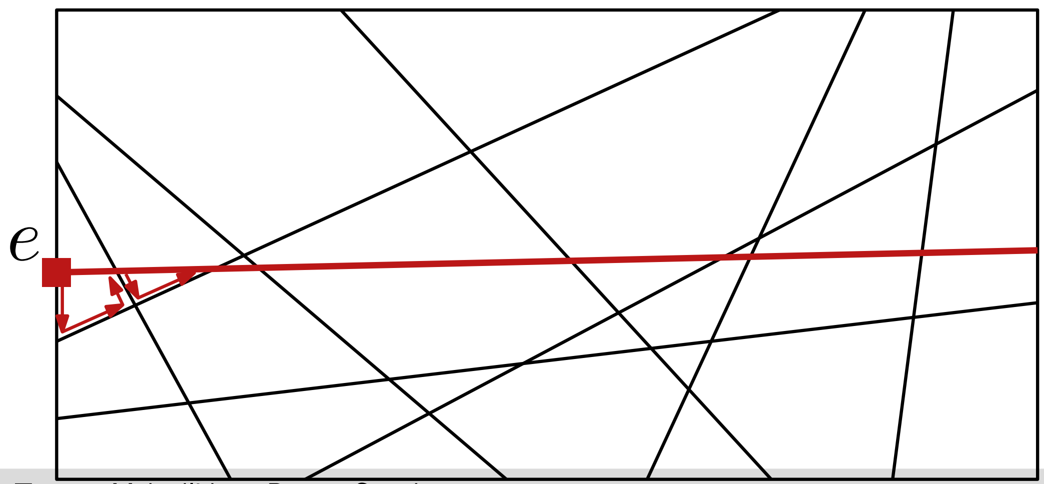
Incrementally Constructing $\mathcal{A}(L)$

Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



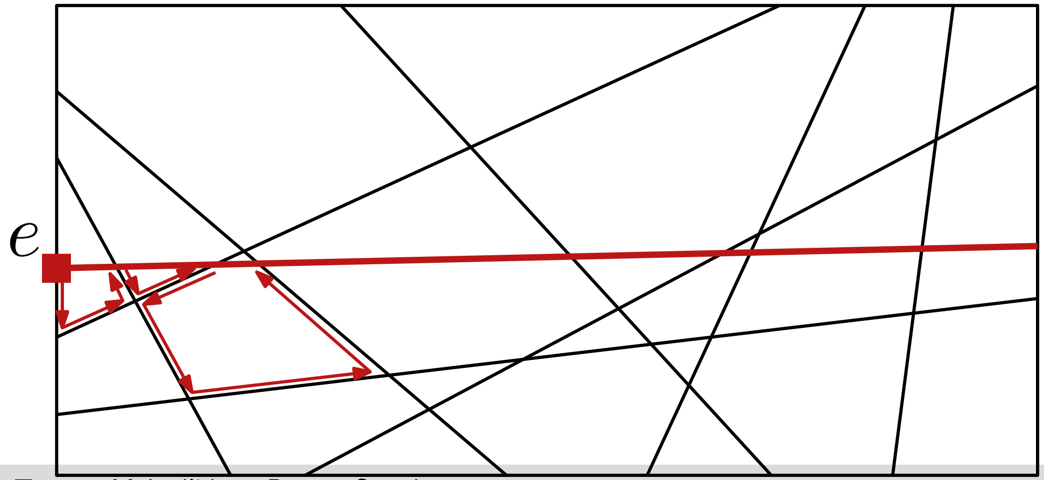
Incrementally Constructing $\mathcal{A}(L)$

Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
    find leftmost edge e of B intersecting ℓ_i
    $f \leftarrow$ inner cell incident to e
    while $f \neq$ outer cell **do**
        split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



Incrementally Constructing $\mathcal{A}(L)$

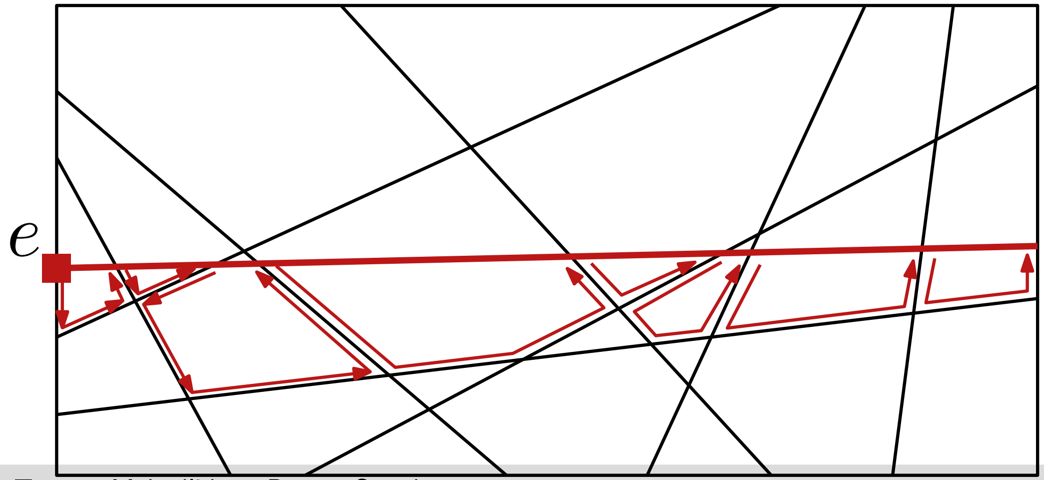
Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



Incrementally Constructing $\mathcal{A}(L)$

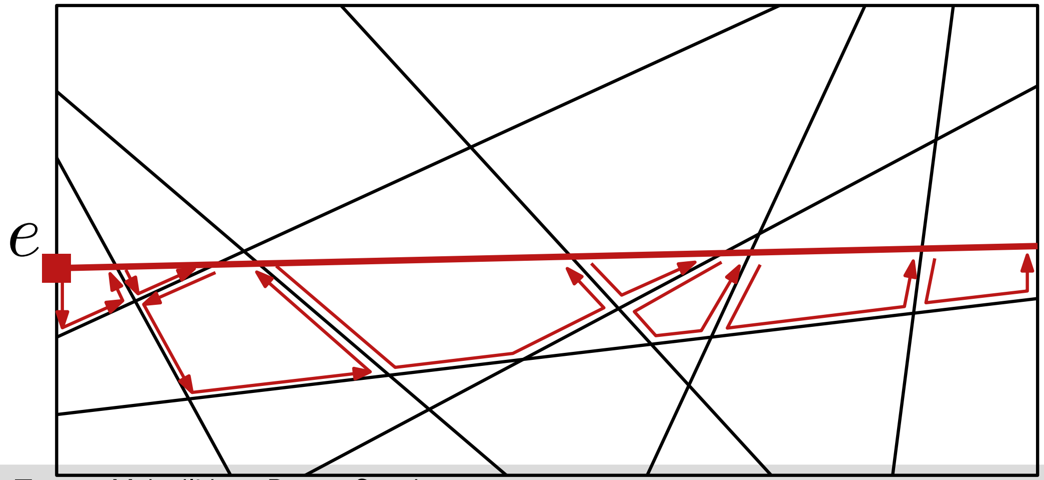
Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$

$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



Incrementally Constructing $\mathcal{A}(L)$

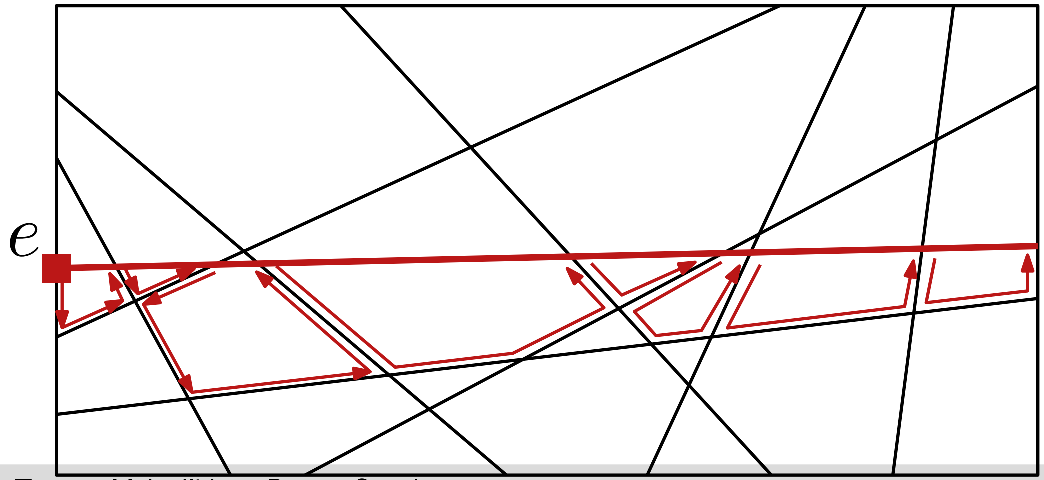
Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



Running time?

Incrementally Constructing $\mathcal{A}(L)$

Input: lines $L = \{\ell_1, \ldots, \ell_n\}$
Output: DCEL $\mathcal{D}$ for $\mathcal{A}(L)$
$\mathcal{D} \leftarrow$ bounding box B of the vertices of $\mathcal{A}(L)$
for $i \leftarrow 1$ **to** n **do**
  find leftmost edge e of B intersecting ℓ_i
  $f \leftarrow$ inner cell incident to e
  while $f \neq$ outer cell **do**
    split f, update $\mathcal{D}$ and set f to the next cell intersected by ℓ_i



Running time?

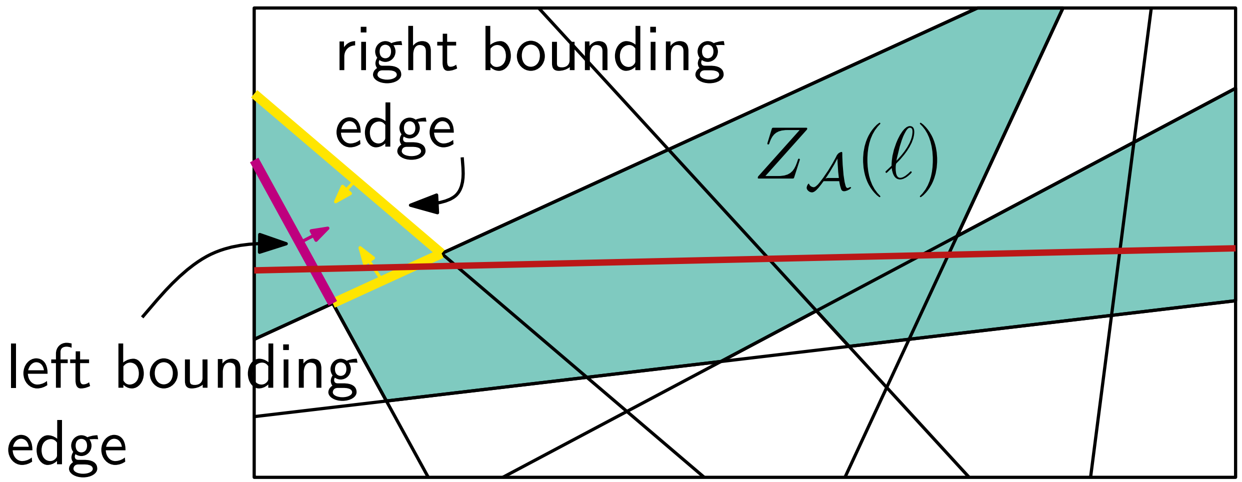
- bounding box: $O(n^2)$
- start point of ℓ_i: $O(i)$
- **while**-loop: $O(|\text{red path}|)$

Zone Theorem

Def: For an arrangement $\mathcal{A}(L)$ and a line $\ell \notin L$ the **zone** $Z_{\mathcal{A}}(\ell)$ is defined as the set of all cells of $\mathcal{A}(L)$ whose closure intersects ℓ.

Zone Theorem

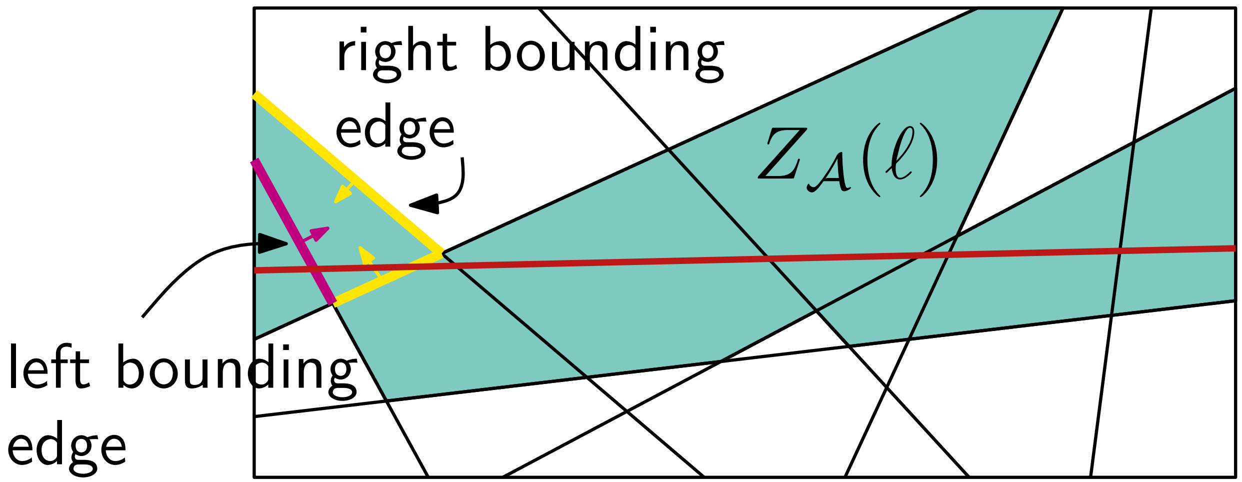
Def: For an arrangement $\mathcal{A}(L)$ and a line $\ell \notin L$ the **zone** $Z_{\mathcal{A}}(\ell)$ is defined as the set of all cells of $\mathcal{A}(L)$ whose closure intersects ℓ.



How many edges are in $Z_{\mathcal{A}}(\ell)$?

Zone Theorem

Def: For an arrangement $\mathcal{A}(L)$ and a line $\ell \notin L$ the **zone** $Z_{\mathcal{A}}(\ell)$ is defined as the set of all cells of $\mathcal{A}(L)$ whose closure intersects ℓ.

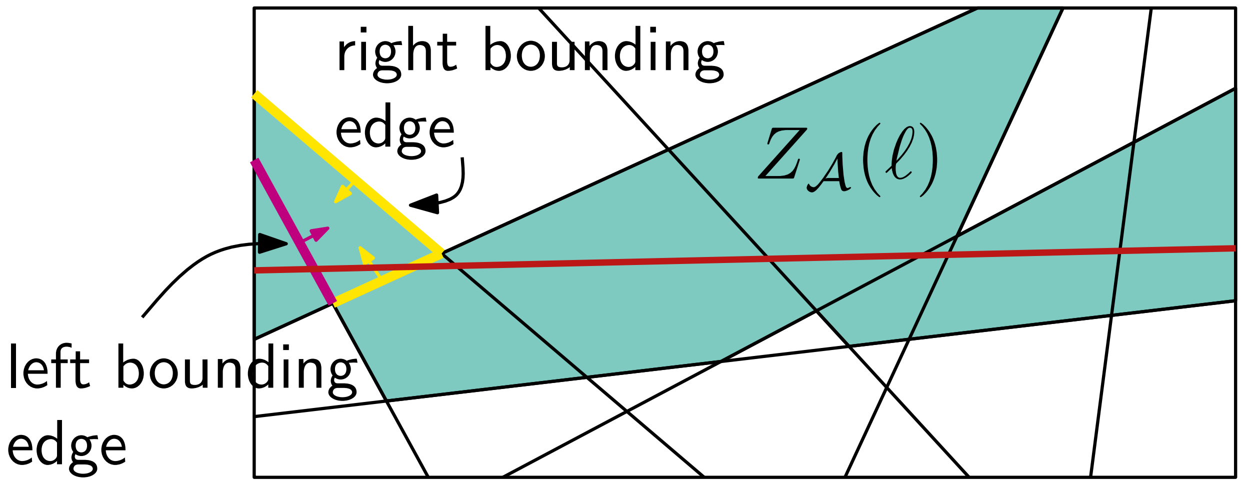


How many edges are in $Z_{\mathcal{A}}(\ell)$?

Theorem 2: For an arrangement $\mathcal{A}(L)$ of n lines in the plane and a line $\ell \notin L$ the zone $Z_{\mathcal{A}}(\ell)$ consist of at most $6n$ edges.

Zone Theorem

Def: For an arrangement $\mathcal{A}(L)$ and a line $\ell \notin L$ the **zone** $Z_{\mathcal{A}}(\ell)$ is defined as the set of all cells of $\mathcal{A}(L)$ whose closure intersects ℓ.



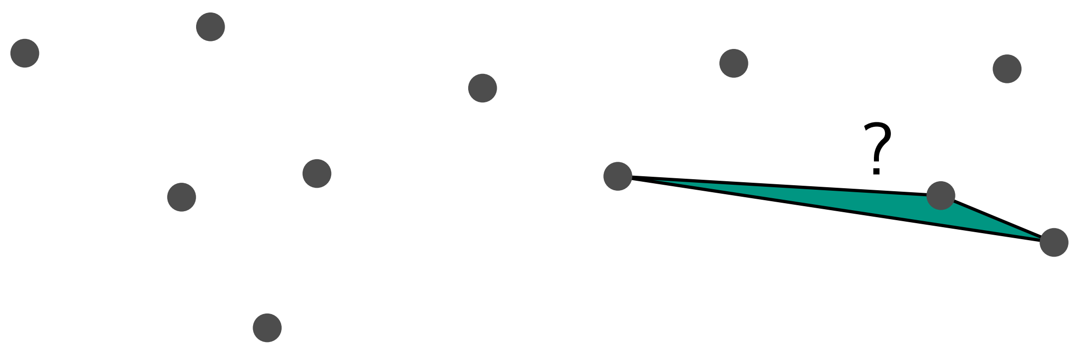
How many edges are in $Z_{\mathcal{A}}(\ell)$?

Theorem 2: For an arrangement $\mathcal{A}(L)$ of n lines in the plane and a line $\ell \notin L$ the zone $Z_{\mathcal{A}}(\ell)$ consist of at most $6n$ edges.

Theorem 3: The arrangement $\mathcal{A}(L)$ of a set of n lines can be constructed in $O(n^2)$ time and space.

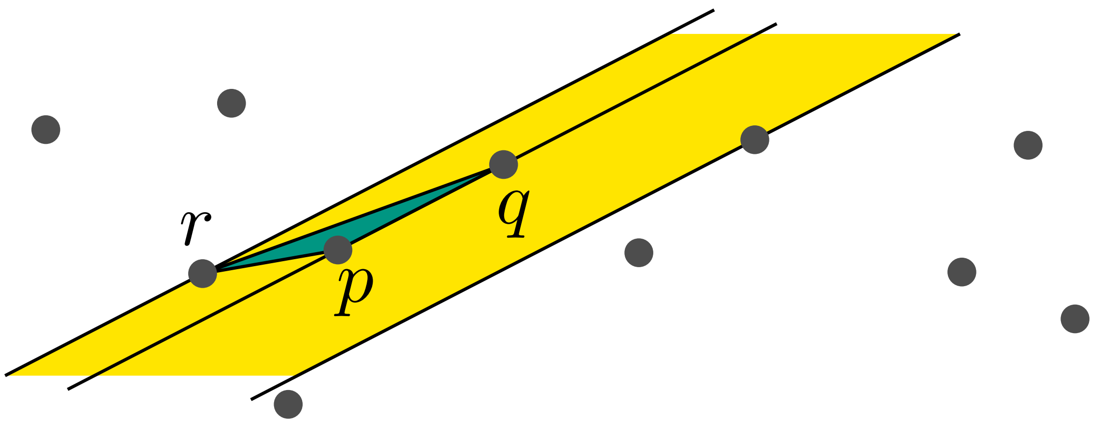
Smallest Triangle

Given a set P of n points in $\mathbb{R}^2$, find a minimum-area triangle Δpqr with $p, q, r \in P$.



Smallest Triangle

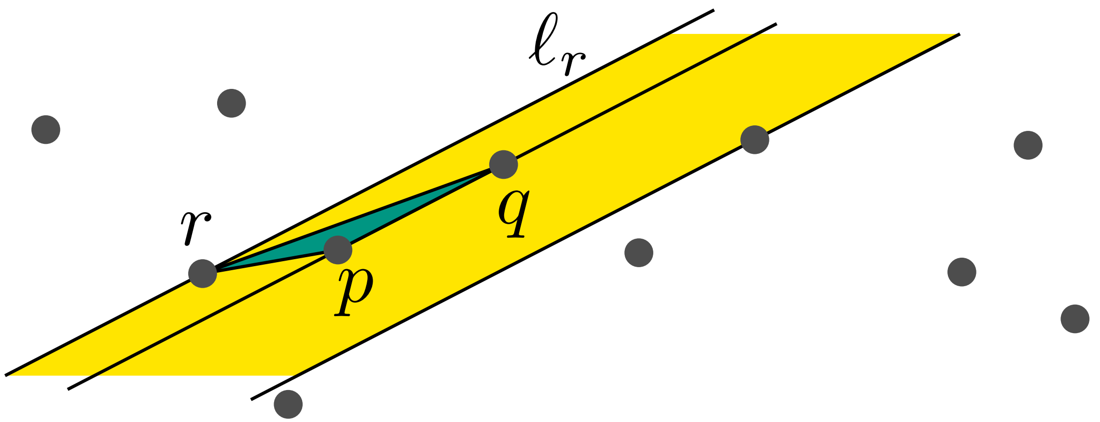
Given a set P of n points in $\mathbb{R}^2$, find a minimum-area triangle Δpqr with $p, q, r \in P$.



Let $p, q \in P$. The point $r \in P \setminus \{p, q\}$ minimizing Δpqr lies on the boundary of the most thin empty corridor parallel to pq.

Smallest Triangle

Given a set P of n points in $\mathbb{R}^2$, find a minimum-area triangle Δpqr with $p, q, r \in P$.

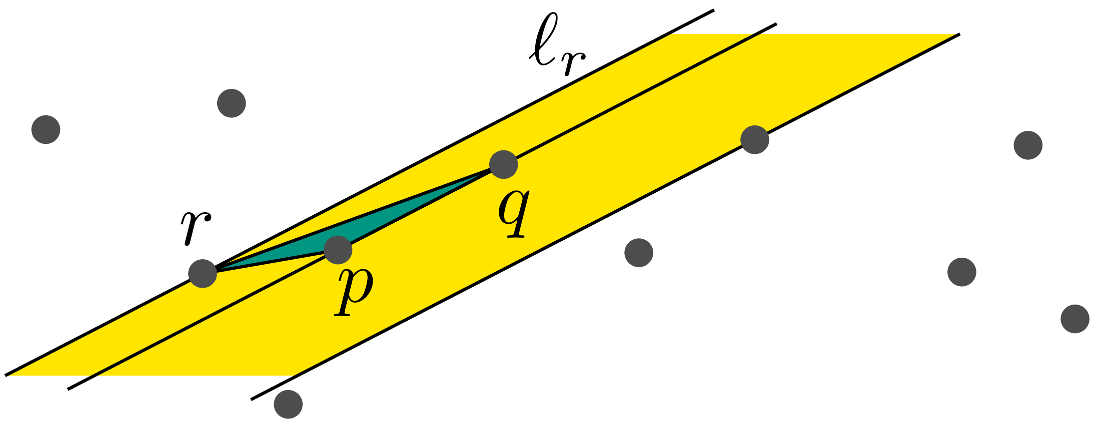


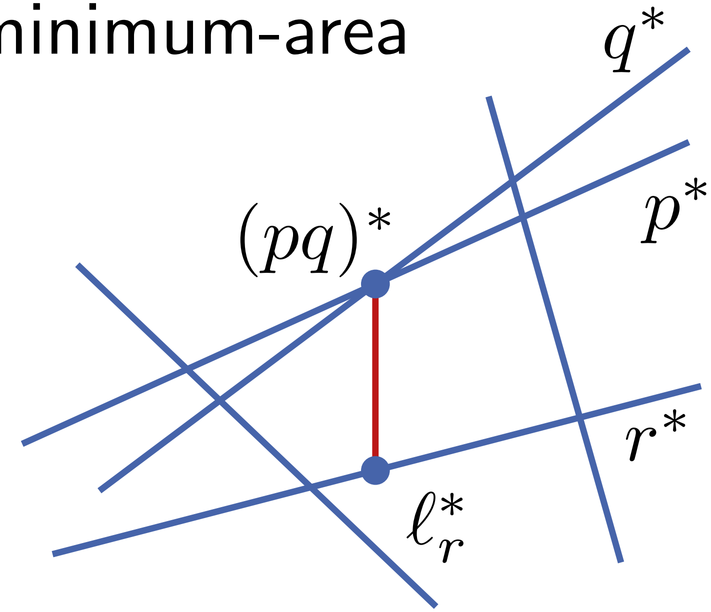
Let $p, q \in P$. The point $r \in P \setminus \{p, q\}$ minimizing Δpqr lies on the boundary of the most thin empty corridor parallel to pq.

There is no other point in P between pq and the line ℓ_r through r and parallel to pq.

Smallest Triangle

Given a set P of n points in $\mathbb{R}^2$, find a minimum-area triangle Δpqr with $p, q, r \in P$.





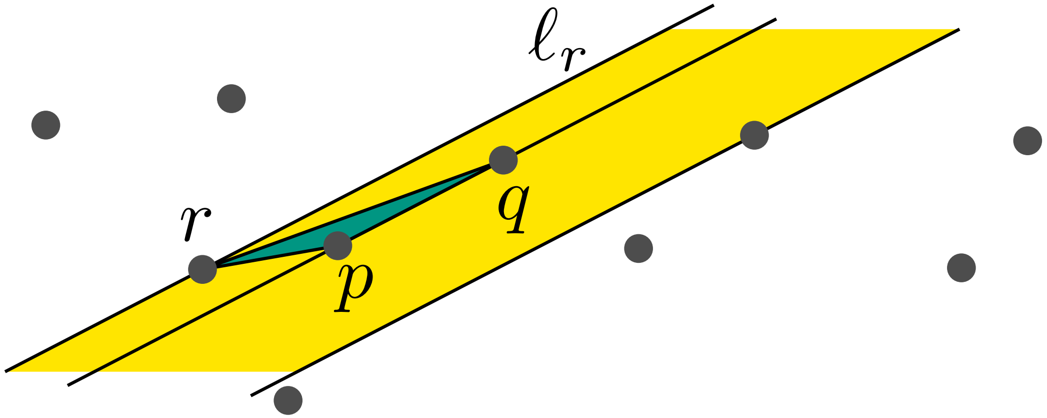
Let $p, q \in P$. The point $r \in P \setminus \{p, q\}$ minimizing Δpqr lies on the boundary of the most thin empty corridor parallel to pq.

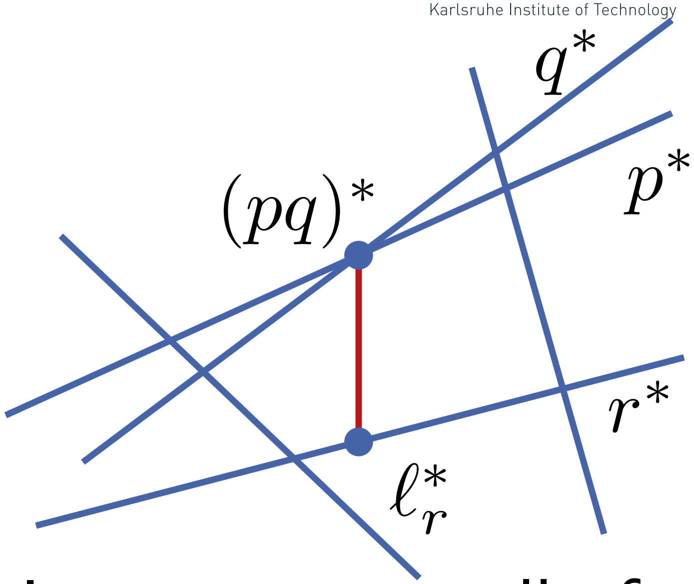
There is no other point in P between pq and the line ℓ_r through r and parallel to pq.

In dual plane:

- ℓ_r^* lies on r^*
- ℓ_r^* and $(pq)^*$ have identical x-coordinate
- no line $p^* \in P^*$ intersects $\overline{\ell_r^* (pq)^*}$

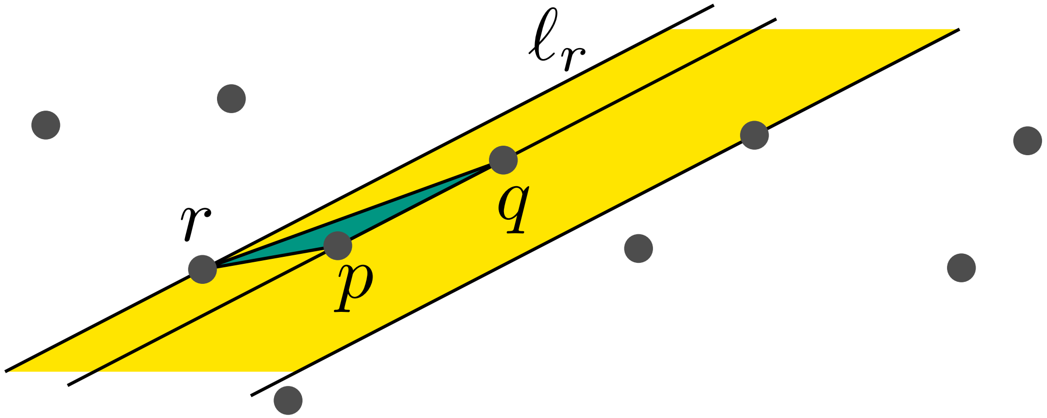
Computing in the Dual

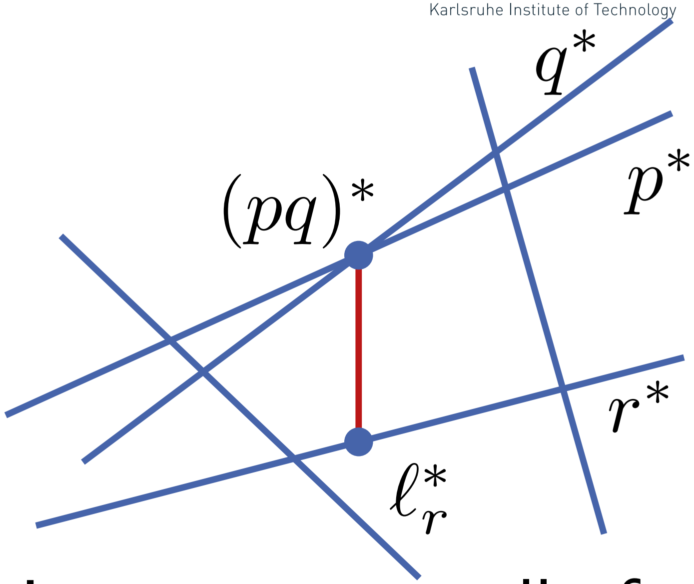




- ℓ_r^* lies vertically above or below $(pq)^*$ in a common cell of $\mathcal{A}(P^*) \Rightarrow$ only two candidates

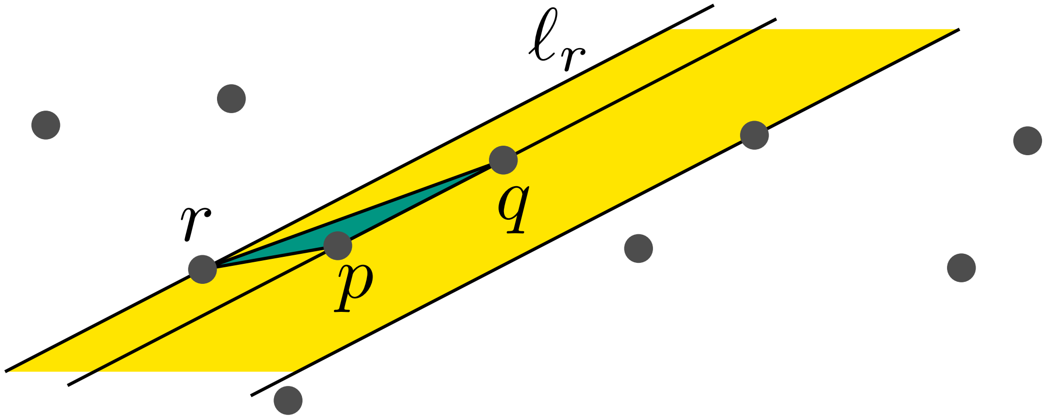
Computing in the Dual

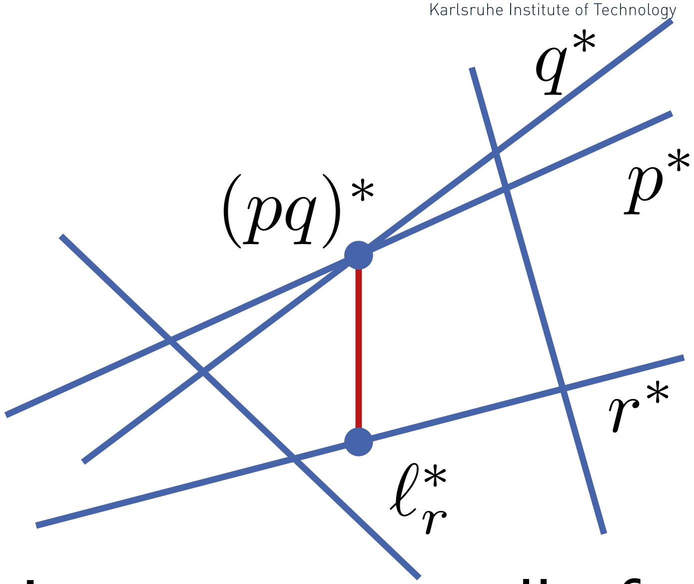




- ℓ_r^* lies vertically above or below $(pq)^*$ in a common cell of $\mathcal{A}(P^*) \Rightarrow$ only two candidates
- Compute in $O(n^2)$ time the arrangement $\mathcal{A}(P^*)$

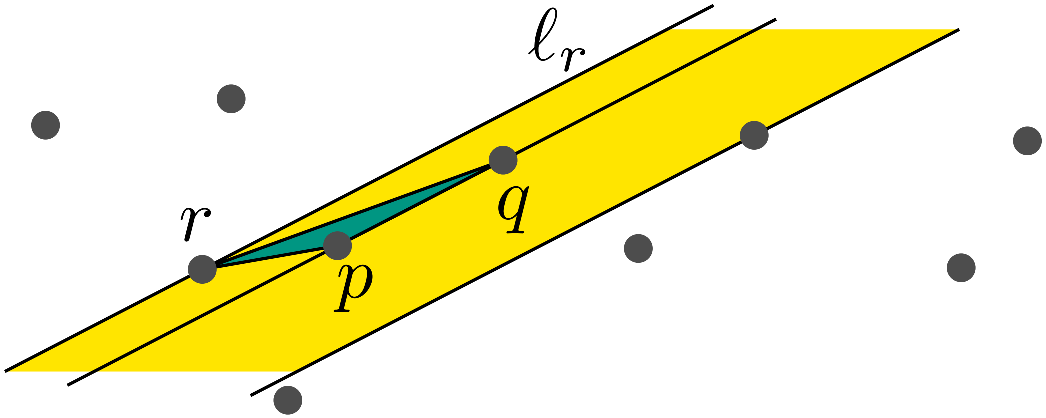
Computing in the Dual

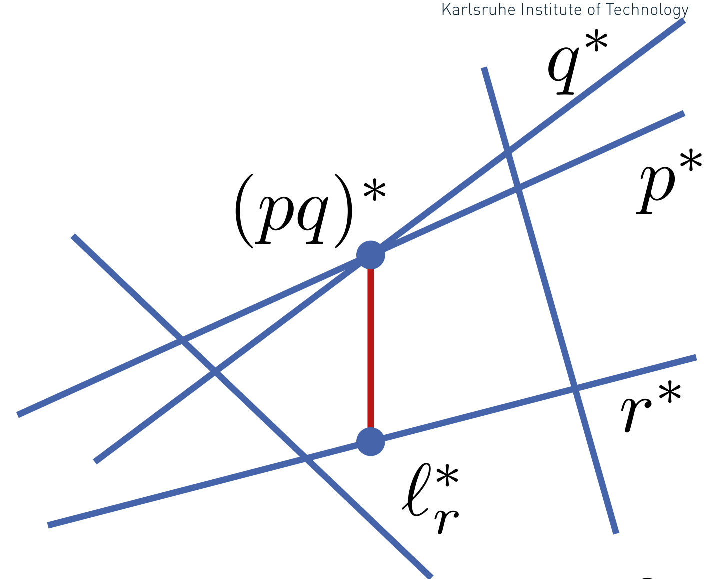




- ℓ_r^* lies vertically above or below $(pq)^*$ in a common cell of $\mathcal{A}(P^*)$ $\Rightarrow$ only two candidates
- Compute in $O(n^2)$ time the arrangement $\mathcal{A}(P^*)$
- With a single traversal of a cell (left-to-right) compute the vertical neighbors of the vertices $\rightarrow$ time linear in cell size

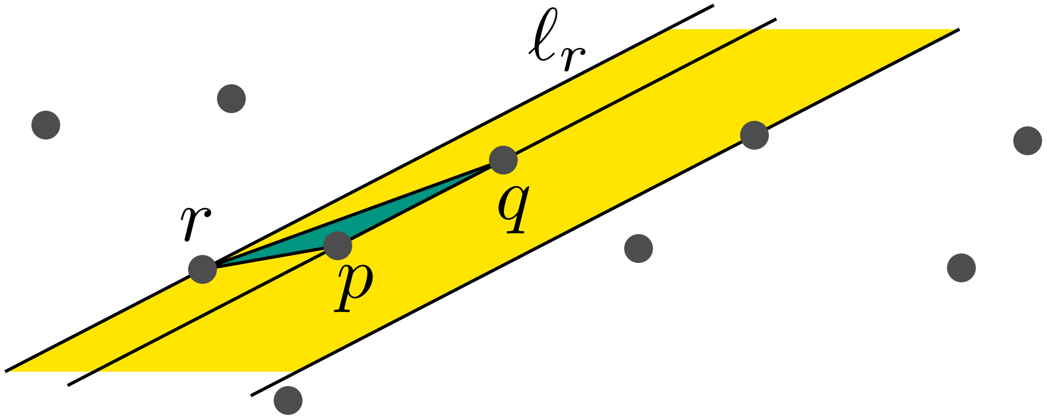
Computing in the Dual

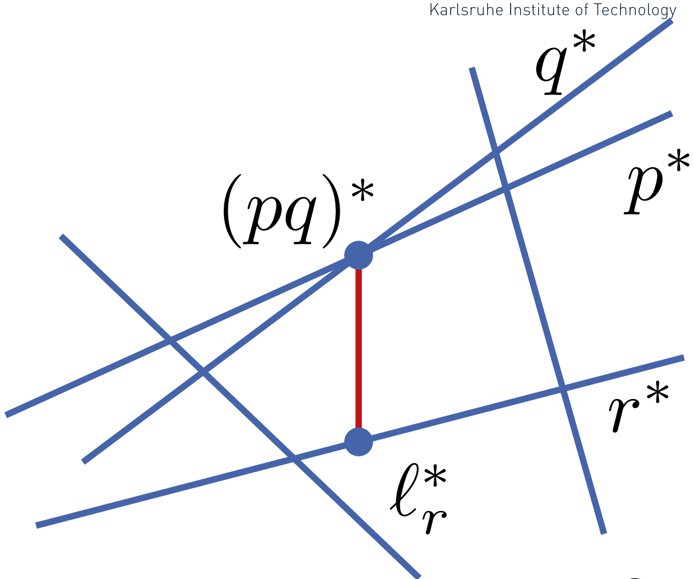




- ℓ_r^* lies vertically above or below $(pq)^*$ in a common cell of $\mathcal{A}(P^*)$ $\Rightarrow$ only two candidates
- Compute in $O(n^2)$ time the arrangement $\mathcal{A}(P^*)$
- With a single traversal of a cell (left-to-right) compute the vertical neighbors of the vertices $\rightarrow$ time linear in cell size
- for all $O(n^2)$ candidate triples $(pq)^* r^*$ compute in $O(1)$ time the area of Δpqr

Computing in the Dual





- ℓ_r^* lies vertically above or below $(pq)^*$ in a common cell of $\mathcal{A}(P^*) \Rightarrow$ only two candidates
- Compute in $O(n^2)$ time the arrangement $\mathcal{A}(P^*)$
- With a single traversal of a cell (left-to-right) compute the vertical neighbors of the vertices $\rightarrow$ time linear in cell size
- for all $O(n^2)$ candidate triples $(pq)^*r^*$ compute in $O(1)$ time the area of Δpqr
- finds minimum in $O(n^2)$ time in total

Further Duality Applications

- Two thieves have stolen a necklace of diamonds and emeralds. They want to share fairly without destroying the necklace more than necessary. How many cuts do they need?

Further Duality Applications

Karlsruhe Institute of Technology

- Two thieves have stolen a necklace of diamonds and emeralds. They want to share fairly without destroying the necklace more than necessary. How many cuts do they need?

Theorem 4: Let D, E be two finite sets of points in $\mathbb{R}^2$. Then there is a line ℓ that divides S and D in half simultaneously.

Further Duality Applications

- Two thieves have stolen a necklace of diamonds and emeralds. They want to share fairly without destroying the necklace more than necessary. How many cuts do they need?

Theorem 4: Let D, E be two finite sets of points in $\mathbb{R}^2$. Then there is a line ℓ that divides S and D in half simultaneously.

- Given n segments in the plane, find a maximum stabbing-line, i.e., a line intersecting as many segments as possible.

Discussion

Duality is a very useful tool in algorithmic geometry!

Discussion

Duality is a very useful tool in algorithmic geometry!

Check: "Monotone Simultaneous Embeddings of Upward Planar Digraphs" Journal of Algorithms and Applications

Discussion

Karlsruhe Institute of Technology

Duality is a very useful tool in algorithmic geometry!

Check: "Monotone Simultaneous Embeddings of Upward Planar Digraphs" Journal of Algorithms and Applications

Can we use duality in higher dimensions?

Discussion

Karlsruhe Institute of Technology

Duality is a very useful tool in algorithmic geometry!

Check: "Monotone Simultaneous Embeddings of Upward Planar Digraphs" Journal of Algorithms and Applications

Can we use duality in higher dimensions?

Yes, you can define incidence- and order-preserving duality transforms between d-dimensional points and hyperplanes.

Discussion

Duality is a very useful tool in algorithmic geometry!

Check: "Monotone Simultaneous Embeddings of Upward Planar Digraphs" Journal of Algorithms and Applications

Can we use duality in higher dimensions?

Yes, you can define incidence- and order-preserving duality transforms between d-dimensional points and hyperplanes.

What about higher-dimensional arrangements?

Discussion

Karlsruhe Institute of Technology

Duality is a very useful tool in algorithmic geometry!

Check: "Monotone Simultaneous Embeddings of Upward Planar Digraphs" Journal of Algorithms and Applications

Can we use duality in higher dimensions?

Yes, you can define incidence- and order-preserving duality transforms between d-dimensional points and hyperplanes.

What about higher-dimensional arrangements?

The arrangement of n hyperplanes in $\mathbb{R}^d$ has complexity $\Theta(n^d)$. A generalization of the Zone Theorem yields an $O(n^d)$-time algorithm.