Lecture 11: Arrangements and Duality

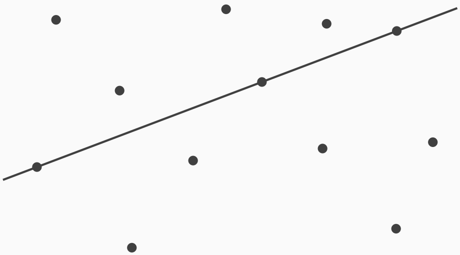
Computational Geometry

Utrecht University

Introduction

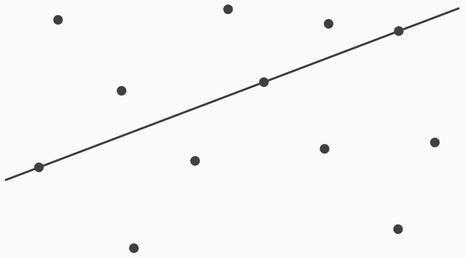
Three Points on a Line

Question: In a set of n points, are there 3 points on a line?



Three Points on a Line

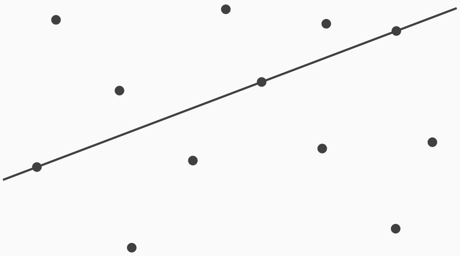
Question: In a set of n points, are there 3 points on a line?



Naive algorithm: tests all triples in $O(n^3)$ time

Three Points on a Line

Question: In a set of n points, are there 3 points on a line?

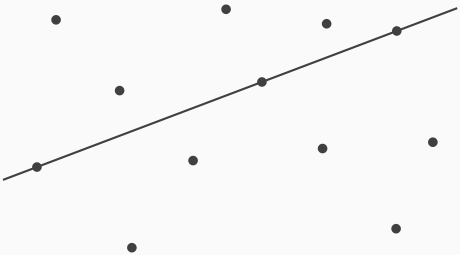


Naive algorithm: tests all triples in $O(n^3)$ time

Faster algorithm: uses duality and arrangements

Three Points on a Line

Question: In a set of n points, are there 3 points on a line?



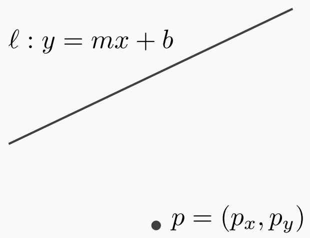
Naive algorithm: tests all triples in $O(n^3)$ time

Faster algorithm: uses duality and arrangements

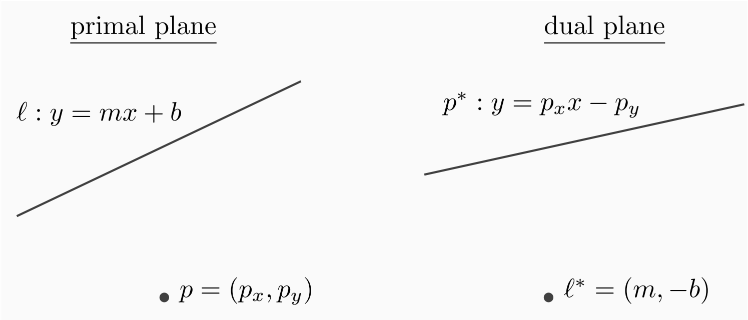
Note: other motivation in chapter 8 of the book

Duality

Duality



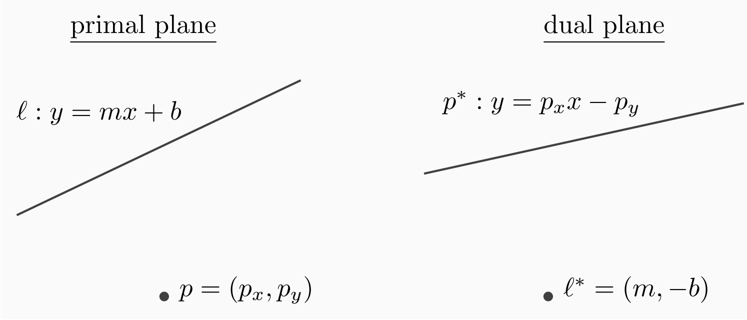
Duality



point $p = (p_x, p_y) \mapsto$ line $p^* : y = p_x x - p_y$

line $\ell : y = mx + b \mapsto$ point $\ell^* = (m, -b)$

Duality

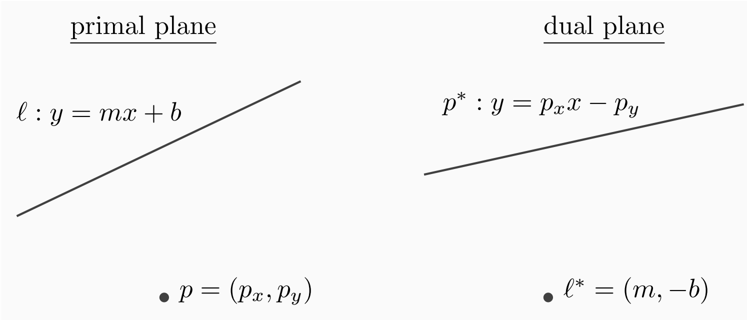


$$\text{point } p = (p_x, p_y) \mapsto \text{line } p^* : y = p_x x - p_y$$

$$\text{line } \ell : y = mx + b \mapsto \text{point } \ell^* = (m, -b)$$

Note: self inverse $(p^*)^* = p, \quad (\ell^*)^* = \ell$

Duality



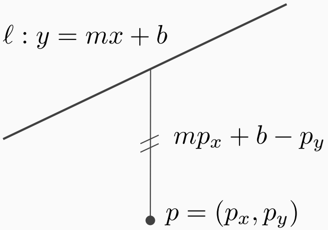
point $p = (p_x, p_y) \mapsto$ line $p^* : y = p_x x - p_y$

line $\ell : y = mx + b \mapsto$ point $\ell^* = (m, -b)$

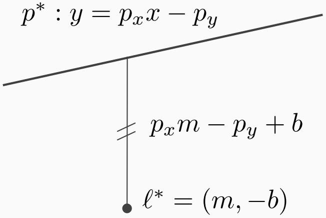
Note: does not handle vertical lines

Duality

primal plane

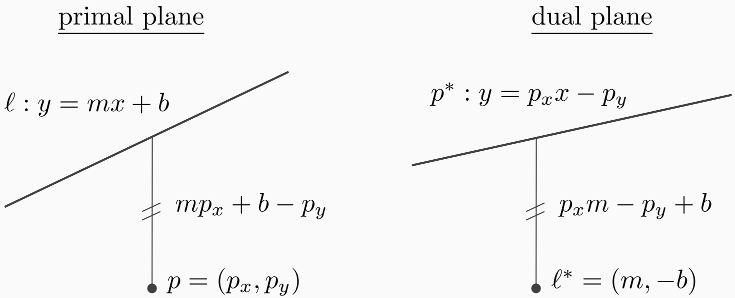


dual plane



Duality preserves vertical distances

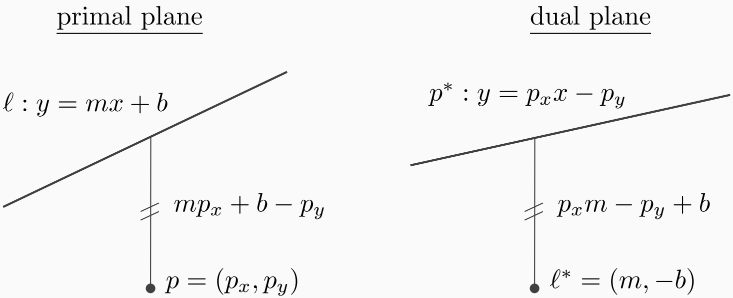
Duality



Duality preserves vertical distances

$\Rightarrow$ incidence preserving: $p \in \ell$ if and only if $\ell^* \in p^*$

Duality



Duality preserves vertical distances

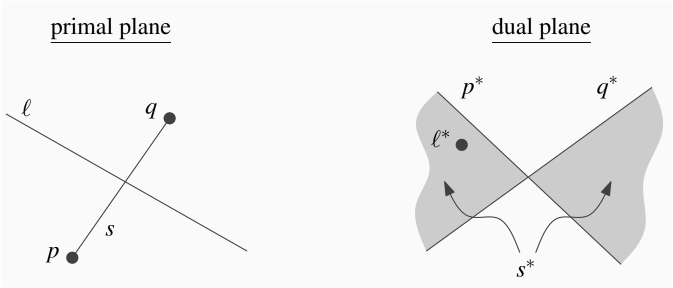
$\Rightarrow$ incidence preserving: $p \in \ell$ if and only if $\ell^* \in p^*$

$\Rightarrow$ order preserving: p lies below ℓ if and only if ℓ^* lies below p^*

It can be applied to other objects, like segments

Duality

It can be applied to other objects, like segments



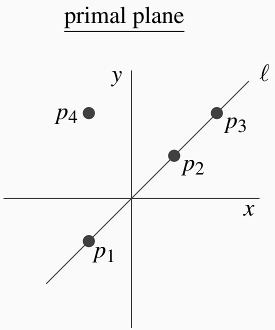
The dual of a segment is a double wedge

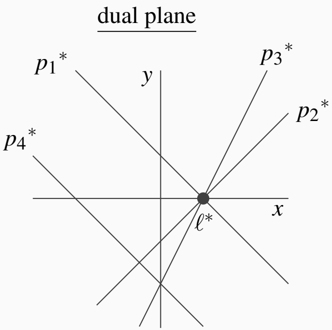
Question: What line would dualize to a point in the right part of the double wedge?

Usefulness of Duality

Why use duality? It gives a new perspective!

Detecting three points on a line dualizes to detecting three lines intersecting in a point

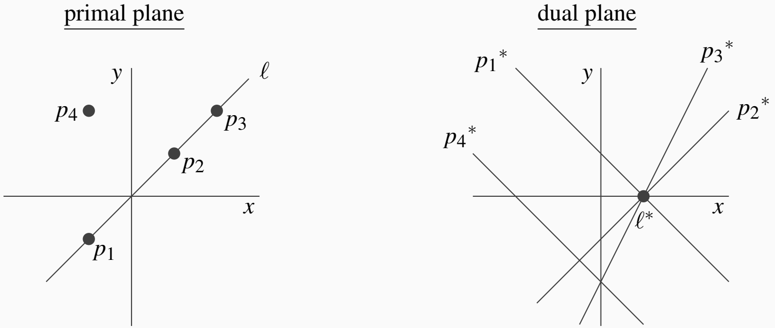




Usefulness of Duality

Why use duality? It gives a new perspective!

Detecting three points on a line dualizes to detecting three lines intersecting in a point

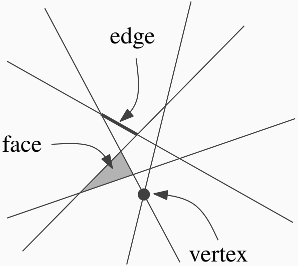


Next we use arrangements

Arrangements

Arrangement $\mathcal{A}(L)$: subdivision induced by a set of lines L

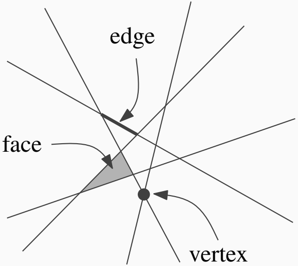
- consists of *faces*, *edges* and *vertices* (some unbounded)



Arrangements of Lines

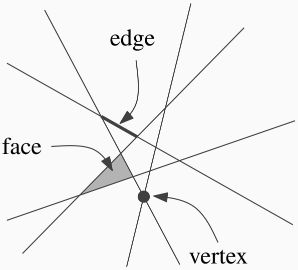
Arrangement $\mathcal{A}(L)$: subdivision induced by a set of lines L

- consists of *faces, edges* and *vertices* (some unbounded)
- arrangements consist of other geometric objects too, like line segments, circles, higher-dimensional objects



Combinatorial Complexity:

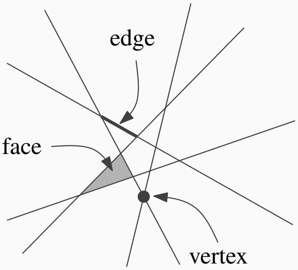
- $\leq n(n-1)/2$ vertices



Arrangements of Lines

Combinatorial Complexity:

- $\leq n(n-1)/2$ vertices
- $\leq n^2$ edges

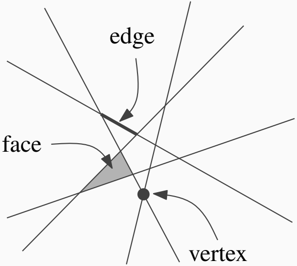


Combinatorial Complexity:

- $\leq n(n-1)/2$ vertices
- $\leq n^2$ edges
- $\leq n^2/2 + n/2 + 1$ faces:

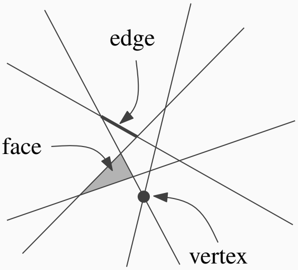
  add lines incrementally

$$1 + \sum_{i=1}^{n} i = n(n+1)/2 + 1$$



Combinatorial Complexity:

- $\leq n(n-1)/2$ vertices
- $\leq n^2$ edges
- $\leq n^2/2 + n/2 + 1$ faces:
  add lines incrementally
  $$1 + \sum_{i=1}^{n} i = n(n+1)/2 + 1$$
- equality holds in *simple* arrangements



Arrangements of Lines

Combinatorial Complexity:

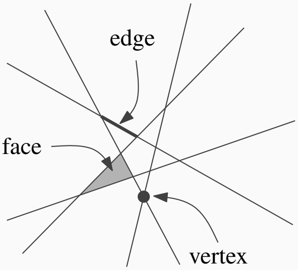
- $\leq n(n-1)/2$ vertices
- $\leq n^2$ edges
- $\leq n^2/2+n/2+1$ faces:

  add lines incrementally

$$1+\sum_{i=1}^{n} i = n(n+1)/2+1$$

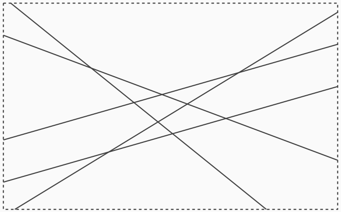
- equality holds in *simple* arrangements

Overall $O(n^2)$ complexity



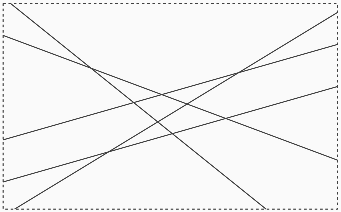
Constructing Arrangements

Goal: Compute $\mathcal{A}(L)$ in bounding box in DCEL representation



Constructing Arrangements

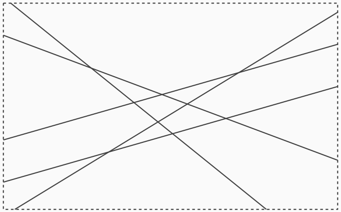
Goal: Compute $\mathcal{A}(L)$ in bounding box in DCEL representation



- plane sweep for line segment intersection: $O((n+k)\log n) = O(n^2 \log n)$

Constructing Arrangements

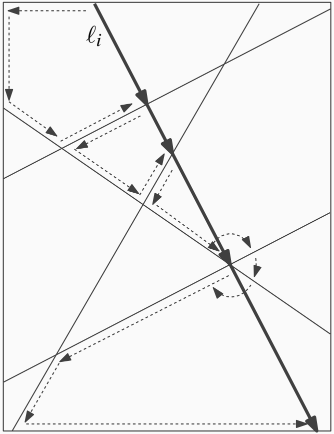
Goal: Compute $\mathcal{A}(L)$ in bounding box in DCEL representation



- plane sweep for line segment intersection: $O((n+k)\log n) = O(n^2 \log n)$
- faster: incremental construction

Arrangements

Incremental Construction



Algorithm ConstructArrangement(L)

Input. Set L of n lines

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$

1. Compute bounding box $\mathcal{B}(L)$
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$
3. **for** $i \leftarrow 1$ **to** n
4.   **do** insert ℓ_i

Incremental Construction

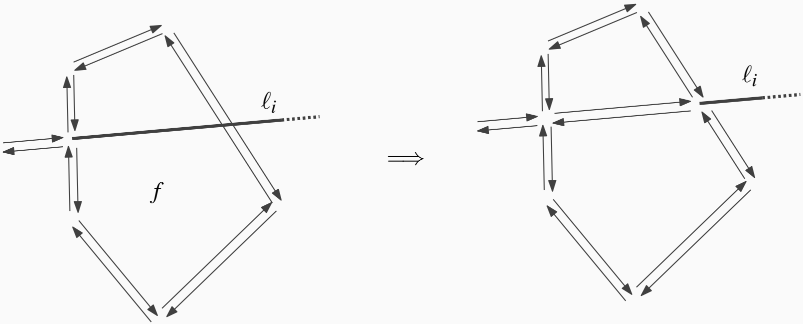
Algorithm ConstructArrangement(L)

Input. A set L of n lines in the plane

Output. DCEL for subdivision induced by $\mathcal{B}(L)$ and the part of $\mathcal{A}(L)$ inside $\mathcal{B}(L)$, where $\mathcal{B}(L)$ is a suitable bounding box

1. Compute a bounding box $\mathcal{B}(L)$ that contains all vertices of $\mathcal{A}(L)$ in its interior
2. Construct DCEL for the subdivision induced by $\mathcal{B}(L)$
3. **for** $i \leftarrow 1$ **to** n
4. **do** Find the edge e on $\mathcal{B}(L)$ that contains the leftmost intersection point of ℓ_i and $\mathcal{A}_i$
5. $f \leftarrow$ the bounded face incident to e
6. **while** f is not the unbounded face, that is, the face outside $\mathcal{B}(L)$
7. **do** Split f, and set f to be the next face intersected by ℓ_i

Incremental Construction



Face split:

Runtime analysis:

Algorithm ConstructArrangement(L)

Input. Set L of n lines

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$

1. Compute bounding box $\mathcal{B}(L)$
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i

Incremental Construction

Runtime analysis:

1. $O(n^2)$

Algorithm ConstructArrangement(L)

Input. Set L of n lines

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$

1. Compute bounding box $\mathcal{B}(L)$
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i

Incremental Construction

Runtime analysis:

1. $O(n^2)$
2. constant

Algorithm ConstructArrangement(L)

Input. Set L of n lines

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$

1. Compute bounding box $\mathcal{B}(L)$
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i

Incremental Construction

Runtime analysis:

1. $O(n^2)$
2. constant

3. & 4. ?

Algorithm ConstructArrangement(L)

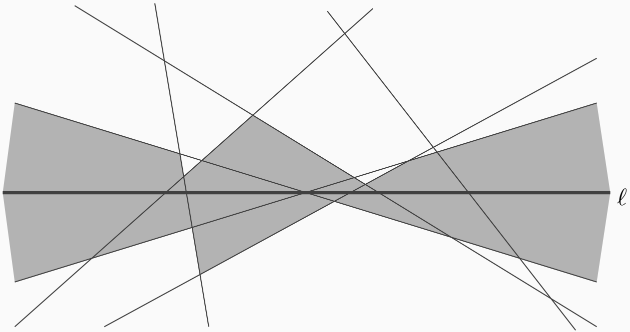
Input. Set L of n lines

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$

1. Compute bounding box $\mathcal{B}(L)$
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i

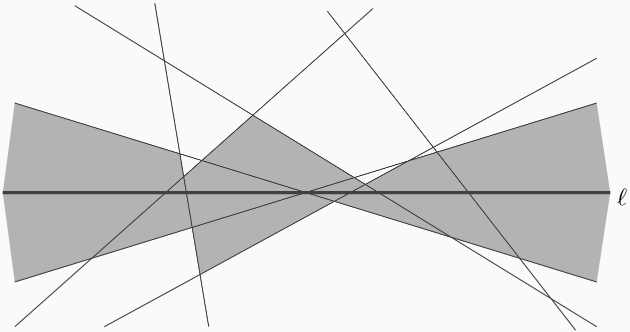
Zone Theorem

The zone of a line ℓ in an arrangement $\mathcal{A}(L)$ is the set of faces of $\mathcal{A}(L)$ whose closure intersects ℓ



Zone Theorem

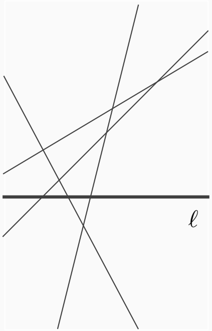
The zone of a line ℓ in an arrangement $\mathcal{A}(L)$ is the set of faces of $\mathcal{A}(L)$ whose closure intersects ℓ



Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$

Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$
Proof:

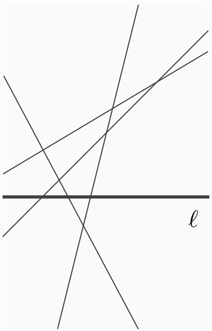
- We can assume ℓ horizontal and no other line is horizontal



Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$

Proof:

- We can assume ℓ horizontal and no other line is horizontal
- We count number of *left-bounding* edges

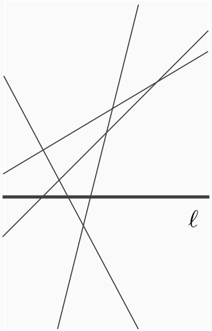


Zone Theorem

Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$

Proof:

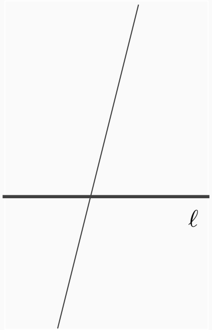
- We can assume ℓ horizontal and no other line is horizontal
- We count number of *left-bounding* edges
- We show by induction on m that this at most $5m$:



Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$

Proof:

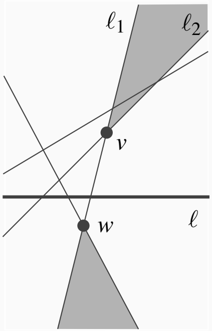
- We can assume ℓ horizontal and no other line is horizontal
- We count number of *left-bounding* edges
- We show by induction on m that this at most $5m$:
  - $m = 1$: trivially true



Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$

Proof:

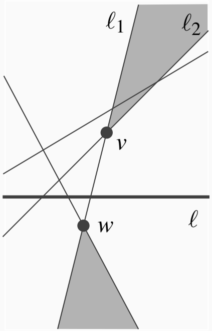
- We can assume ℓ horizontal and no other line is horizontal
- We count number of *left-bounding* edges
- We show by induction on m that this at most $5m$:
  - $m = 1$: trivially true
  - $m > 1$: only at most 3 new edges if ℓ_1 is unique,



Zone Theorem

Theorem: The complexity of the zone of a line in an arrangement of m lines is $O(m)$
Proof:

- We can assume ℓ horizontal and no other line is horizontal
- We count number of *left-bounding* edges
- We show by induction on m that this at most $5m$:
  - $m = 1$: trivially true
  - $m > 1$: only at most 3 new edges if ℓ_1 is unique, at most 5 if ℓ_1 is not unique
    $5(m-1) + 5 = 5m$



Run time analysis:

Algorithm ConstructArrangement(L)

Input. Set L of n lines.

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$.

1. Compute bounding box $\mathcal{B}(L)$.
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$.
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i.

Incremental Construction

Run time analysis:

1. $O(n^2)$

Algorithm ConstructArrangement(L)

Input. Set L of n lines.

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$.

1. Compute bounding box $\mathcal{B}(L)$.
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$.
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i.

Incremental Construction

Run time analysis:

1. $O(n^2)$
2. constant

Algorithm ConstructArrangement(L)

Input. Set L of n lines.

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$.

1. Compute bounding box $\mathcal{B}(L)$.
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$.
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i.

Run time analysis:

1. $O(n^2)$
2. constant

3. & 4. $\sum_{i=1}^{n} O(i) = O(n^2)$

Algorithm ConstructArrangement(L)

Input. Set L of n lines.

Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$.

1. Compute bounding box $\mathcal{B}(L)$.
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$.
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i.

Incremental Construction

Run time analysis:

1. $O(n^2)$
2. constant

3. & 4. $\sum_{i=1}^{n} O(i) = O(n^2)$

In total $O(n^2)$

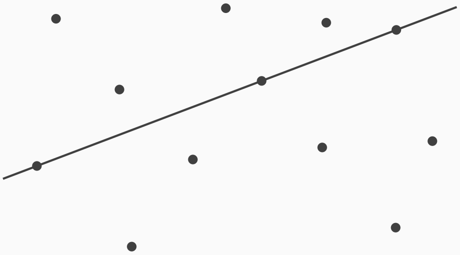
Algorithm ConstructArrangement(L)

Input. Set L of n lines.

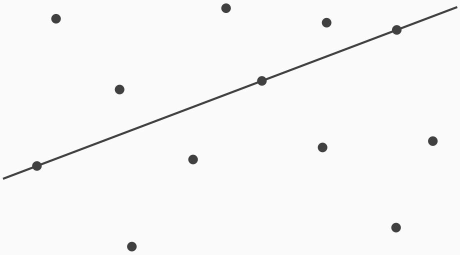
Output. DCEL for $\mathcal{A}(L)$ in $\mathcal{B}(L)$.

1. Compute bounding box $\mathcal{B}(L)$.
2. Construct DCEL for subdivision induced by $\mathcal{B}(L)$.
3. **for** $i \leftarrow 1$ **to** n
4. **do** insert ℓ_i.

3 Points on a Line



3 Points on a Line



Algorithm:

- run incremental construction algorithm for dual problem
- stop when 3 lines pass through a point

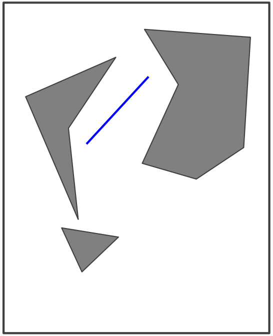
Run time: $O(n^2)$

Arrangements

Motion Planning

Example: Motion Planning

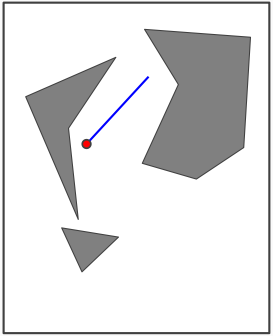
Where can the rod move by translation (no rotations) while avoiding obstacles?



Example: Motion Planning

Where can the rod move by translation (no rotations) while avoiding obstacles?

- pick a reference point:
  lower end-point of rod



Example: Motion Planning

Where can the rod move by translation (no rotations) while avoiding obstacles?

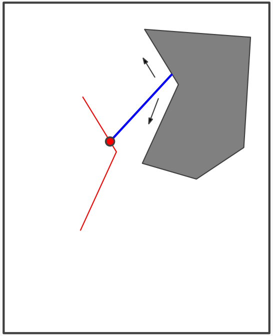
- pick a reference point:

  lower end-point of rod

- shrink rod to a point,

  expand obstacles accordingly:

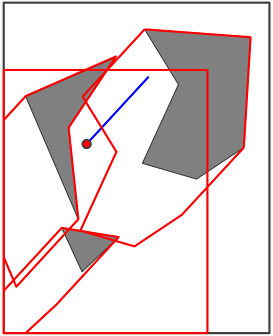
  locus of semi-free placements



Example: Motion Planning

Where can the rod move by translation (no rotations) while avoiding obstacles?

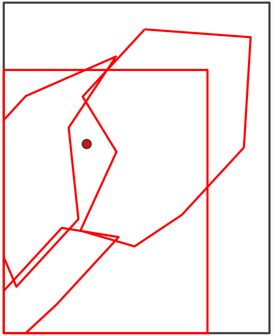
- pick a reference point:
  lower end-point of rod
- shrink rod to a point,
  expand obstacles accordingly:
  locus of semi-free placements



Example: Motion Planning

Where can the rod move by translation (no rotations) while avoiding obstacles?

- pick a reference point:
  lower end-point of rod
- shrink rod to a point,
  expand obstacles accordingly:
  locus of semi-free placements
- reachable configurations:
  cell of initial configuration in
  arrangement of line segments

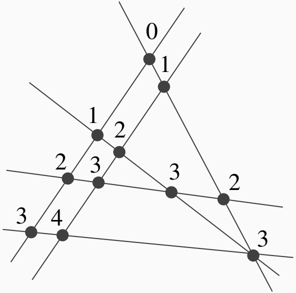


Arrangements

k-Levels

k-levels in Arrangements

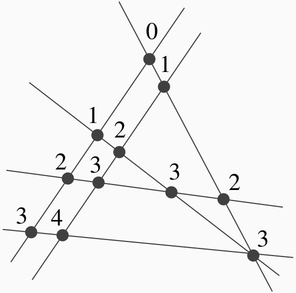
The level of a point in an arrangement of lines is the number of lines strictly above it



k-levels in Arrangements

The level of a point in an arrangement of lines is the number of lines strictly above it

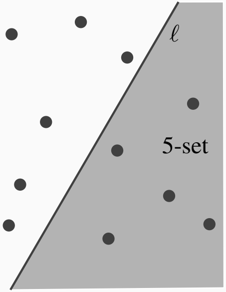
Open problem: What is the complexity of k-levels?



The level of a point in an arrangement of lines is the number of lines strictly above it

Open problem: What is the complexity of k-levels?

Dual problem: What is the number of k-sets in a point set?



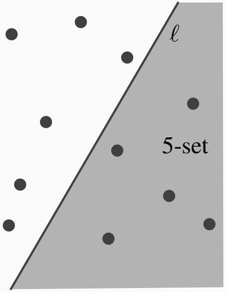
The level of a point in an arrangement of lines is the number of lines strictly above it

Open problem: What is the complexity of k-levels?

Dual problem: What is the number of k-sets in a point set?

Known bounds:

- Erdös et al. ’73: $\Omega(n \log k)$ and $O(nk^{1/2})$



k-levels in Arrangements

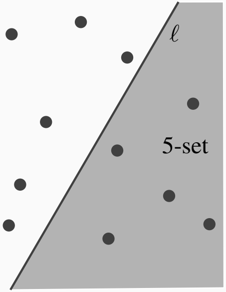
The level of a point in an arrangement of lines is the number of lines strictly above it

Open problem: What is the complexity of k-levels?

Dual problem: What is the number of k-sets in a point set?

Known bounds:

- Erdös et al. '73: $\Omega(n \log k)$ and $O(nk^{1/2})$
- Dey '97: $O(nk^{1/3})$



Three dimensions

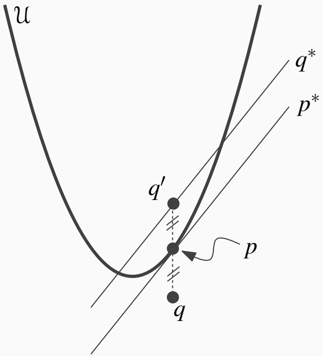
In 3D, we have point-plane duality; lines dualize to other lines

An arrangement induced by n planes in 3D has complexity $O(n^3)$

Deciding whether a set of points in 3D has four or more co-planar points can be done in $O(n^3)$ time (dualize and construct the arrangement)

A geometric interpretation:

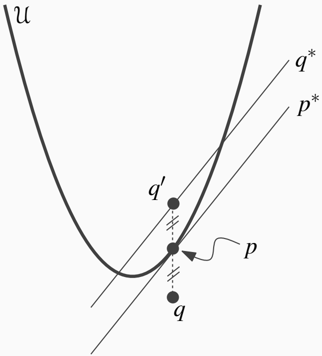
- parabola $\mathcal{U} : y = x^2/2$



More Duality

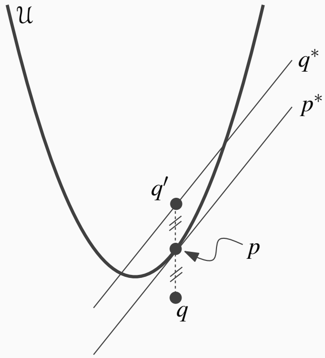
A geometric interpretation:

- parabola $\mathcal{U} : y = x^2/2$
- point $p = (p_x, p_y)$ on $\mathcal{U}$



A geometric interpretation:

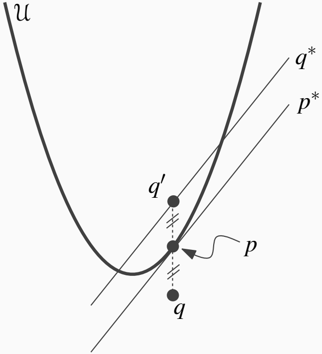
- parabola $\mathcal{U} : y = x^2/2$
- point $p = (p_x, p_y)$ on $\mathcal{U}$
- derivative of $\mathcal{U}$ at p is p_x, i.e., p^* has same slope as the tangent line



More Duality

A geometric interpretation:

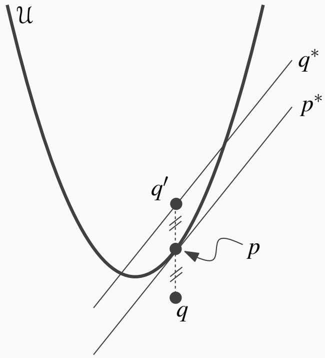
- parabola $\mathcal{U} : y = x^2/2$
- point $p = (p_x, p_y)$ on $\mathcal{U}$
- derivative of $\mathcal{U}$ at p is p_x, i.e., p^* has same slope as the tangent line
- the tangent line intersects y-axis at $(0, -p_x^2/2)$



More Duality

A geometric interpretation:

- parabola $\mathcal{U} : y = x^2/2$
- point $p = (p_x, p_y)$ on $\mathcal{U}$
- derivative of $\mathcal{U}$ at p is p_x, i.e., p^* has same slope as the tangent line
- the tangent line intersects y-axis at $(0, -p_x^2/2)$
- $\Rightarrow$ p^* is the tangent line at p



Summary

Duality is a useful tool to reformulate certain problems on points in the plane to lines in the plane, and vice versa

Dualization of line segments is especially useful

Arrangements, zones of lines in arrangements, and levels in arrangements are useful concepts in computational geometry

All of this exists in three and higher dimensional spaces too