Algorithmic Robotics and Motion Planning

Fall 2006/7

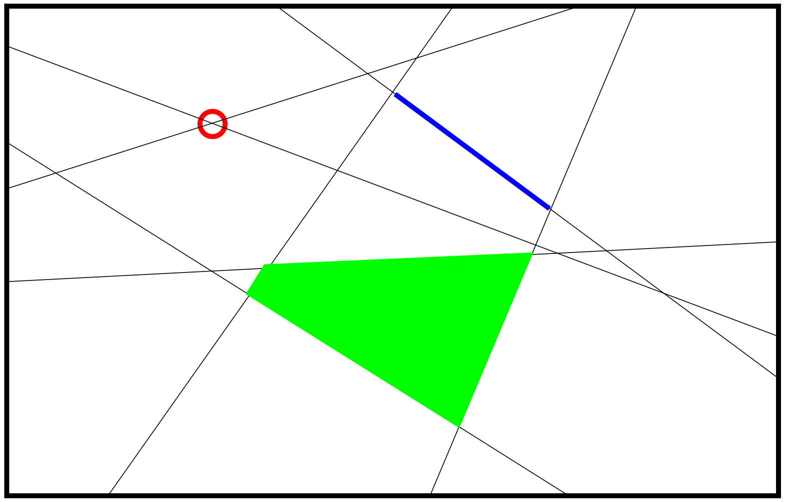
Arrangements, an Overview

Dan Halperin

Tel Aviv University

What are arrangements?

Example: an arrangement of lines



vertex

edge

face

What are arrangements, cont'd

- an arrangement of a set S of geometric objects is the subdivision of space where the objects reside induced by S

What are arrangements, cont'd

- an arrangement of a set S of geometric objects is the subdivision of space where the objects reside induced by S
- possibly non-linear objects (parabolas), bounded objects (segments, circles), higher dimensions (planes, simplices)

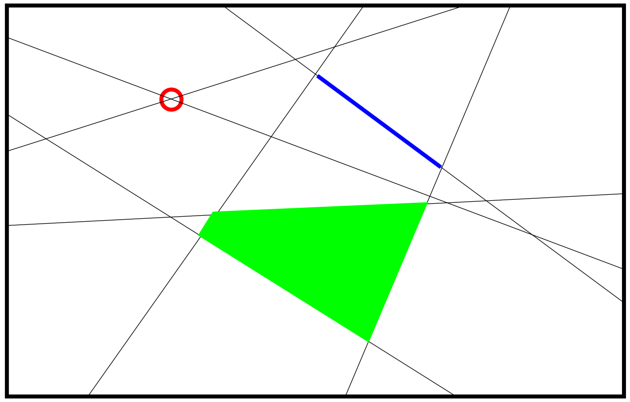
What are arrangements, cont'd

- an arrangement of a set S of geometric objects is the subdivision of space where the objects reside induced by S
- possibly non-linear objects (parabolas), bounded objects (segments, circles), higher dimensions (planes, simplices)
- numerous applications in robotics, molecular biology, vision, graphics, CAD/CAM, statistics, GIS

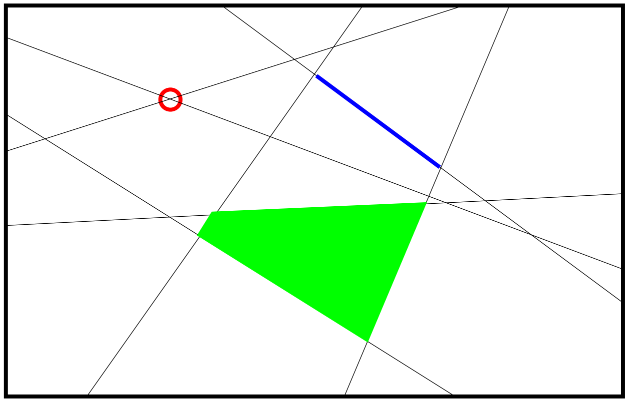
What are arrangements, cont'd

- an arrangement of a set S of geometric objects is the subdivision of space where the objects reside induced by S
- possibly non-linear objects (parabolas), bounded objects (segments, circles), higher dimensions (planes, simplices)
- numerous applications in robotics, molecular biology, vision, graphics, CAD/CAM, statistics, GIS
- have been studied for decades, originally mostly combinatorics
  nowadays mainly studied in combinatorial and computational geometry

Arrangements of lines: Combinatorics

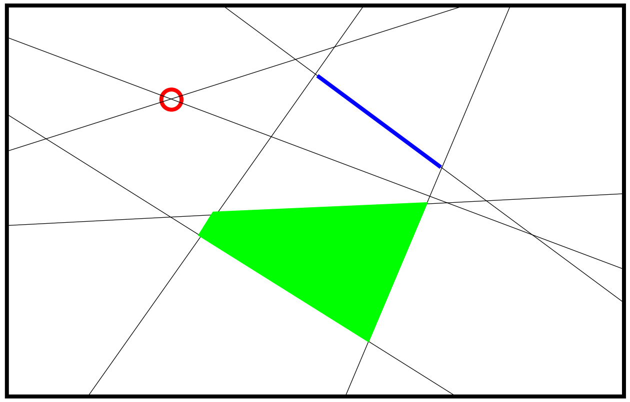


Arrangements of lines: Combinatorics



the complexity of an arrangement is the overall number of cells of all dimensions comprising the arrangement

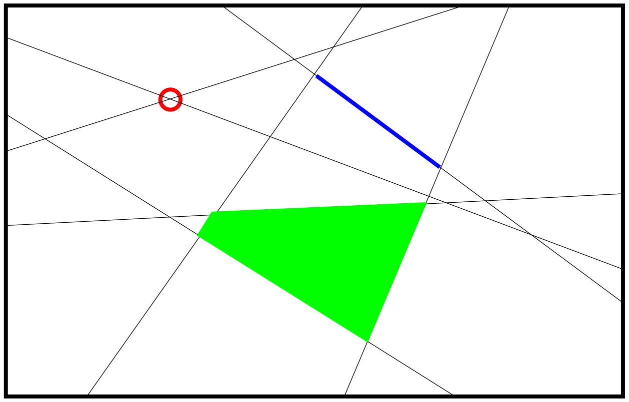
Arrangements of lines: Combinatorics



the complexity of an arrangement is the overall number of cells of all dimensions comprising the arrangement

for planar arrangements we count: vertices, edges, and faces

Arrangements of lines: Combinatorics



the complexity of an arrangement is the overall number of cells of all dimensions comprising the arrangement

for planar arrangements we count: vertices, edges, and faces

the general position assumption: two lines meet in a single point, three lines have no point in common

In an arrangements of n lines

number of vertices:

In an arrangements of n lines

number of vertices: $n(n-1)/2$

In an arrangements of n lines

number of vertices: $n(n-1)/2$

number of edges:

In an arrangements of n lines

number of vertices: $n(n-1)/2$

number of edges: n^2

In an arrangements of n lines

number of vertices: $n(n-1)/2$

number of edges: n^2

number of faces:

In an arrangements of n lines

number of vertices: $n(n-1)/2$

number of edges: n^2

number of faces:
using Euler's formula $\#V - \#E + \#F = 2$
we get $n^2/2 + n/2 + 1$

Basic theorem of arrangement complexity

the maximum combinatorial complexity of an arrangement of n well-behaved curves in the plane is $O(n^2)$; there are such arrangements whose complexity is $\Omega(n^2)$

Basic theorem of arrangement complexity

the maximum combinatorial complexity of an arrangement of n well-behaved curves in the plane is $O(n^2)$; there are such arrangements whose complexity is $\Omega(n^2)$

more generally

Basic theorem of arrangement complexity

the maximum combinatorial complexity of an arrangement of n well-behaved curves in the plane is $O(n^2)$; there are such arrangements whose complexity is $\Omega(n^2)$

more generally

the maximum combinatorial complexity of an arrangement of n well-behaved (hyper)surfaces in $\mathbb{R}^d$ is $O(n^d)$; there are such arrangements whose complexity is $\Omega(n^d)$

Constructing the arrangement

- representation (data structure)
  - DCEL - the doubly connected edge list

Constructing the arrangement

- representation (data structure)
  - DCEL - the doubly connected edge list

- algorithm
  - incremental

Constructing the arrangement

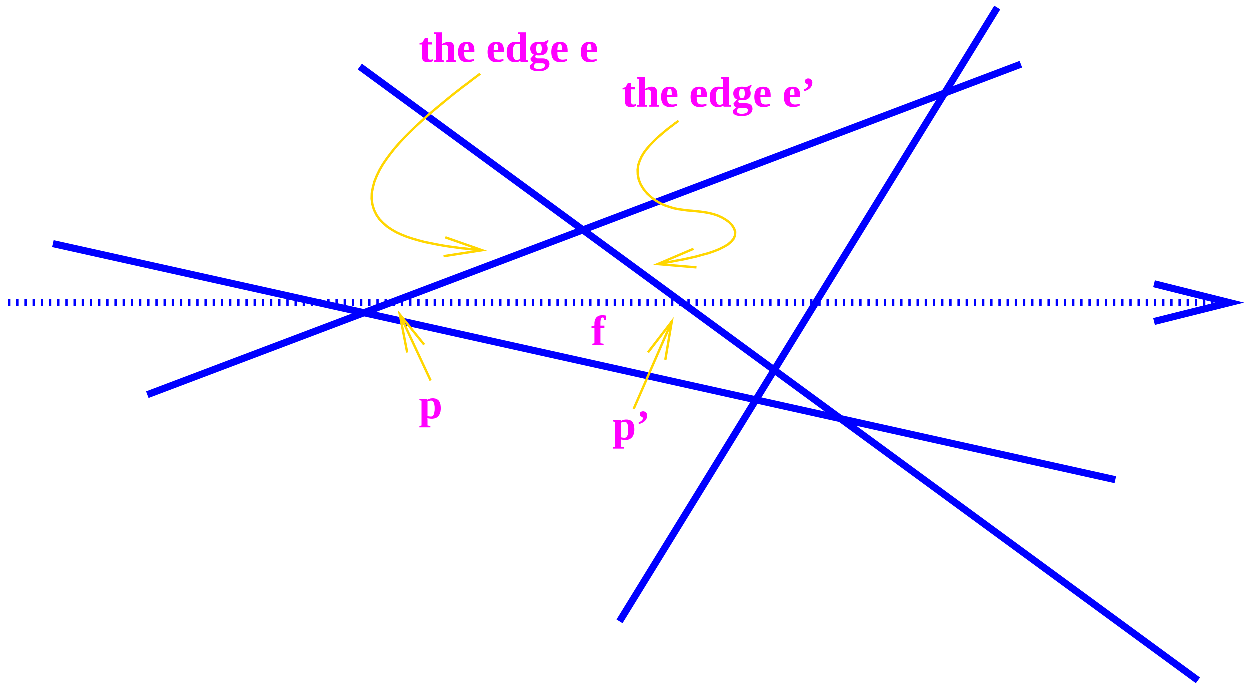
- representation (data structure)
  - DCEL - the doubly connected edge list
- algorithm
  - incremental
- computational model
  - the real RAM

Constructing the arrangement

- representation (data structure)
  - DCEL - the doubly connected edge list
- algorithm
  - incremental
- computational model
  - the real RAM
- the general position assumption

Incremental construction

- computing a bounding box
- inserting the ith line



How much time does it take?

the steps:

- computing the bounding box

How much time does it take?

the steps:

- computing the bounding box

- finding where to insert line i

How much time does it take?

the steps:

- computing the bounding box
- finding where to insert line i
- inserting line i

How much time does it take?

the steps:

- computing the bounding box
naively $O(n^2)$;
- finding where to insert line i
- inserting line i

How much time does it take?

the steps:

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$, exercise

- finding where to insert line i

- inserting line i

How much time does it take?

the steps:

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$, exercise

- finding where to insert line i

simple, $O(i)$

- inserting line i

How much time does it take?

the steps:

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$, exercise

- finding where to insert line i

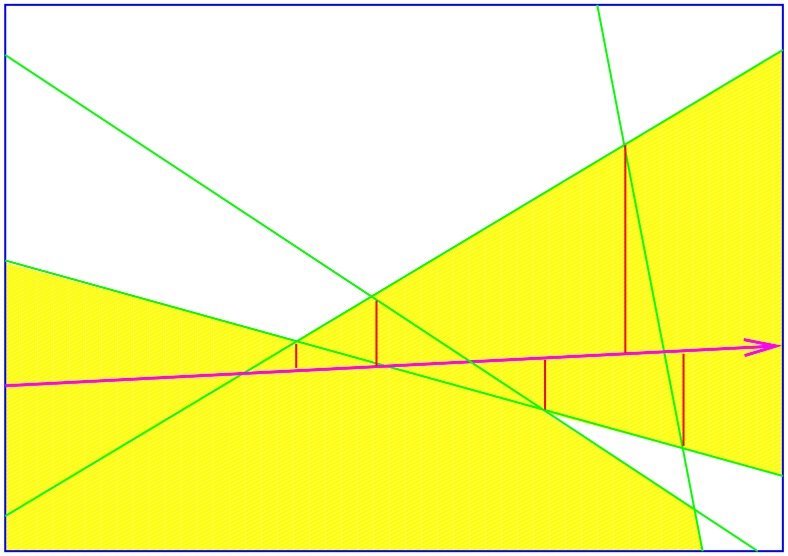
simple, $O(i)$

- inserting line i

$O($zone complexity$)$

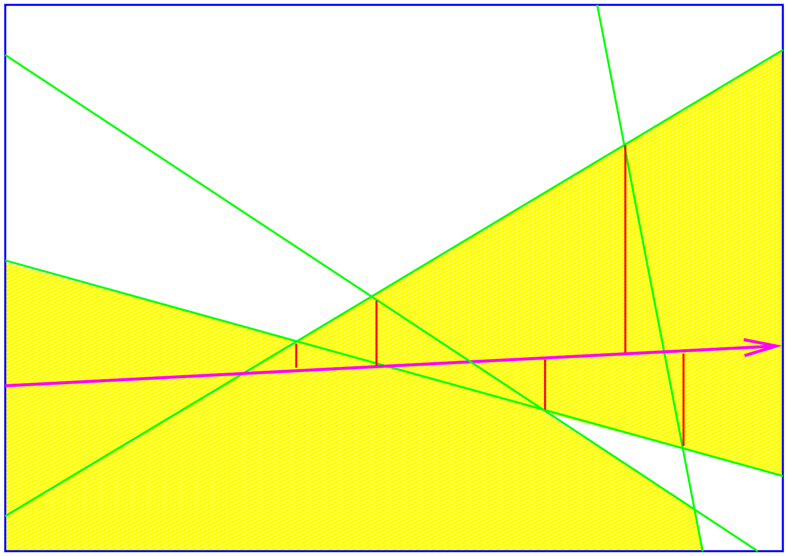
The zone of a curve

the zone of a curve γ in an arrangement $\mathcal{A}$ is the collection of faces of $\mathcal{A}$ intersected by γ



The zone of a curve

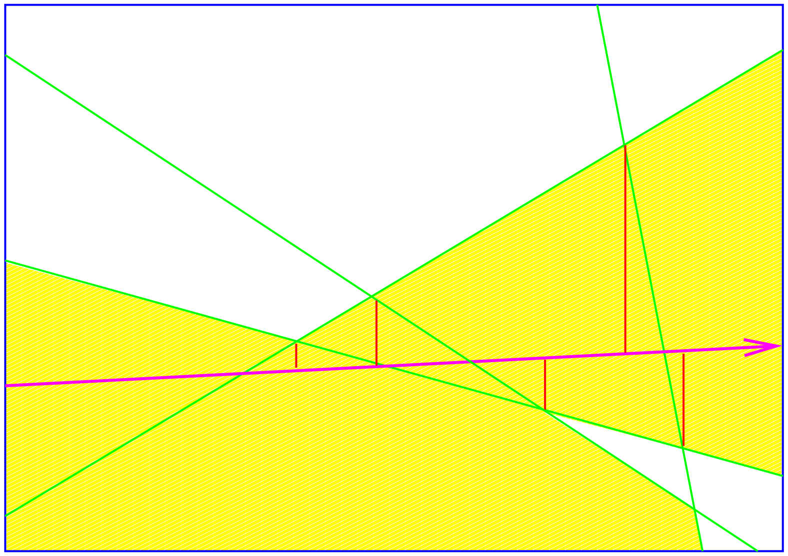
the zone of a curve γ in an arrangement $\mathcal{A}$ is the collection of faces of $\mathcal{A}$ intersected by γ



the complexity of the zone is the overall complexity of cells of various dimensions in the closure of the zone

The zone of a curve

the zone of a curve γ in an arrangement $\mathcal{A}$ is the collection of faces of $\mathcal{A}$ intersected by γ



the complexity of the zone is the overall complexity of cells of various dimensions in the closure of the zone

we need: the complexity of the zone of a line in an arrangement of i lines

Zone theorem

theorem: the complexity of the zone of a line in an arrangement of i lines is $O(i)$

Zone theorem

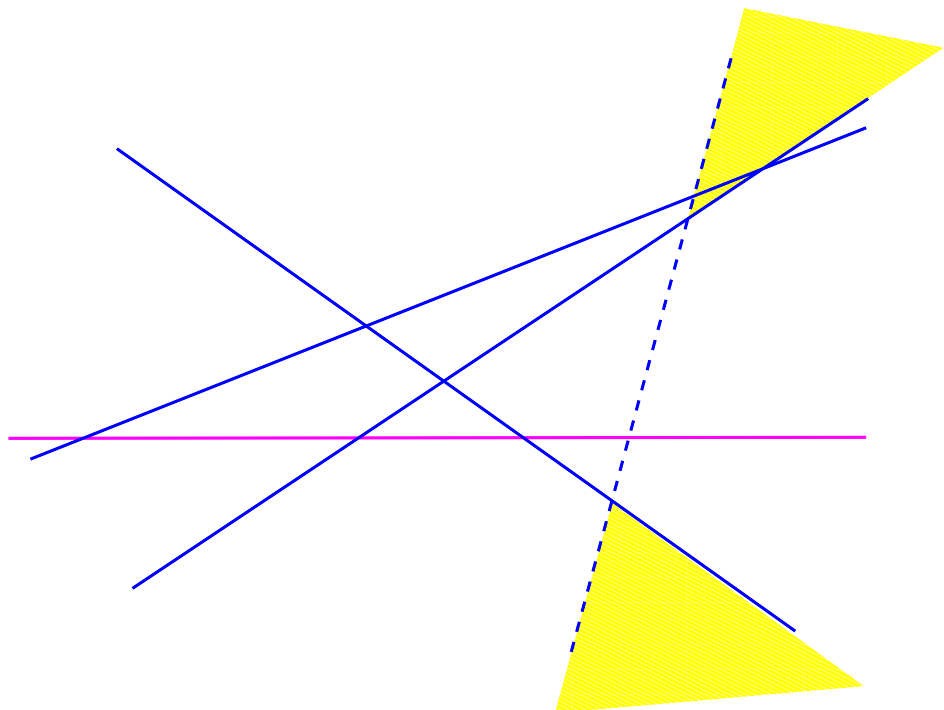
theorem: the complexity of the zone of a line in an arrangement of i lines is $O(i)$

proof:

Zone theorem

theorem: the complexity of the zone of a line in an arrangement of i lines is $O(i)$

proof:



Overall running time of the algorithm

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$

- finding where to insert line i

simple, $O(i)$

- inserting line i

$O(\text{zone complexity})$

Overall running time of the algorithm

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$

- finding where to insert line i

simple, $O(i)$

- inserting line i

$O($zone complexity$)$ = $O(i)$

Overall running time of the algorithm

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$

- finding where to insert line i

simple, $O(i)$

- inserting line i

$O($zone complexity$)$ = $O(i)$

overall $O(n^2)$ time

Basics of arrangements, summary

- the combinatorial complexity of an arrangement
- incremental construction
- the zone of an(other) object in an arrangement
- the basic theorem of arrangement complexity
- the real RAM model,the general position assumption

Configuration spaces

- arrangements $\mathcal{A}(\mathcal{S})$ are used for exact discretization of continuous problems

Configuration spaces

- arrangements $\mathcal{A}(\mathcal{S})$ are used for exact discretization of continuous problems
- a point p in configuration space $\mathcal{C}$ has a property $\Pi(p)$

Configuration spaces

- arrangements $\mathcal{A}(\mathcal{S})$ are used for exact discretization of continuous problems
- a point p in configuration space $\mathcal{C}$ has a property $\Pi(p)$
- if a neighborhood U of p is not intersected by an object in $\mathcal{S}$, the same property $\Pi(q)$ holds for every point $q \in U$
  (the same holds when we restrict the configuration space to an object in $\mathcal{S}$)

Configuration spaces

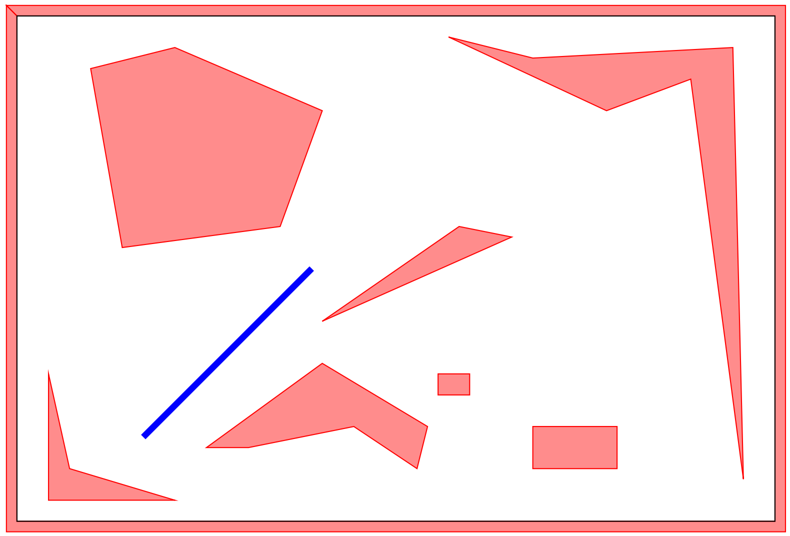
- arrangements $\mathcal{A}(\mathcal{S})$ are used for exact discretization of continuous problems
- a point p in configuration space $\mathcal{C}$ has a property $\Pi(p)$
- if a neighborhood U of p is not intersected by an object in $\mathcal{S}$, the same property $\Pi(q)$ holds for every point $q \in U$
  (the same holds when we restrict the configuration space to an object in $\mathcal{S}$)
- the objects in $\mathcal{S}$ are critical

Configuration spaces

- arrangements $\mathcal{A}(\mathcal{S})$ are used for exact discretization of continuous problems
- a point p in configuration space $\mathcal{C}$ has a property $\Pi(p)$
- if a neighborhood U of p is not intersected by an object in $\mathcal{S}$, the same property $\Pi(q)$ holds for every point $q \in U$
  (the same holds when we restrict the configuration space to an object in $\mathcal{S}$)
- the objects in $\mathcal{S}$ are critical
- the property is invariant in each cell of the arrangement

Configuration space for translational motion planning

the rod is translating in the room



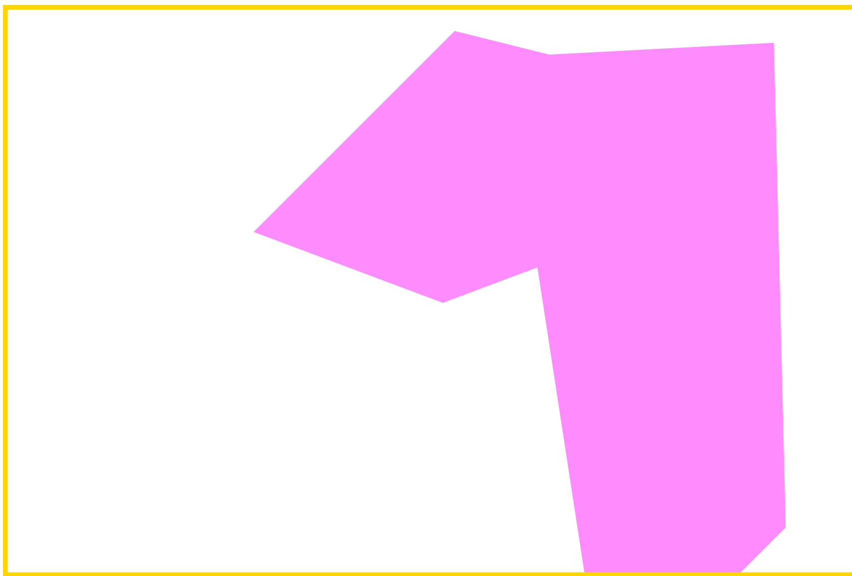
the reference point: the lower end-point of the rod
the configuration space is 2 dimensional

Configuration space obstacles

the robot has shrunk to a point

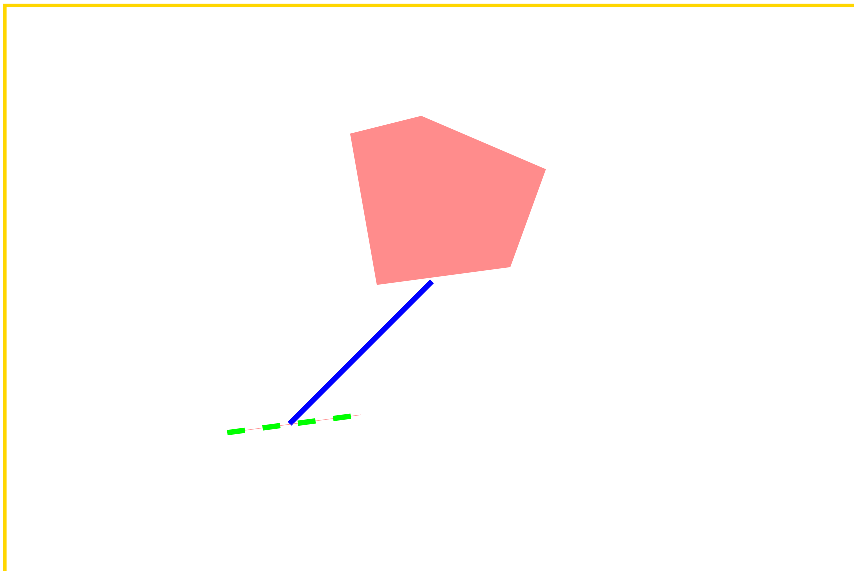
$\Rightarrow$

the obstacles are accordingly expanded

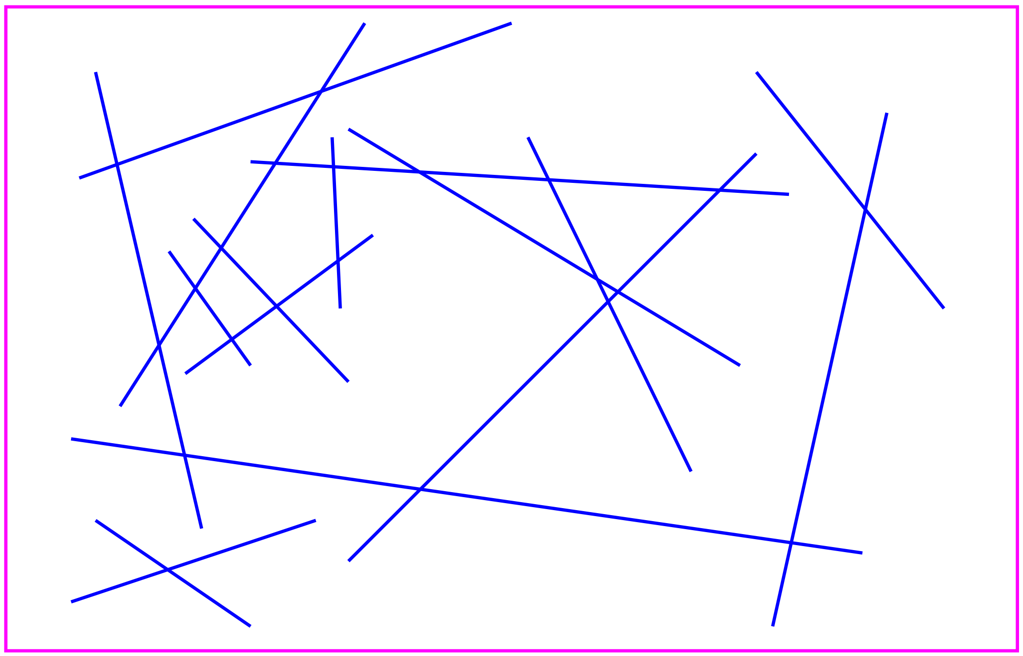


Critical curves in configuration space

the locus of semi-free placements



The arrangement of critical curves in configuration space



Solving a motion-planning problem

- what are the critical curves

Solving a motion-planning problem

- what are the critical curves
- how complex is the arrangement of the critical curves

Solving a motion-planning problem

- what are the critical curves
- how complex is the arrangement of the critical curves
- constructing the arrangement and filtering out the forbidden cells

Solving a motion-planning problem

- what are the critical curves
- how complex is the arrangement of the critical curves
- constructing the arrangement and filtering out the forbidden cells
- what is the complexity of the free space

Solving a motion-planning problem

- what are the critical curves
- how complex is the arrangement of the critical curves
- constructing the arrangement and filtering out the forbidden cells
- what is the complexity of the free space
- can we compute the free space efficiently

Solving a motion-planning problem

- what are the critical curves
- how complex is the arrangement of the critical curves
- constructing the arrangement and filtering out the forbidden cells
- what is the complexity of the free space
- can we compute the free space efficiently
- do we need to compute the entire free space?

Example: a disc moving among discs

- the critical curves are circles

Example: a disc moving among discs

- the critical curves are circles
- how complex is the arrangement of the circles?

Example: a disc moving among discs

- the critical curves are circles
- how complex is the arrangement of the circles?
- what is the complexity of the free space?

Example: a disc moving among discs

- the critical curves are circles
- how complex is the arrangement of the circles?
- what is the complexity of the free space?
- can we compute the free space efficiently?

Example: an L-shaped robot moving among points

- what are the critical curves?

Example: an L-shaped robot moving among points

- what are the critical curves?
- how complex is the arrangement of the critical curves?

Example: an L-shaped robot moving among points

- what are the critical curves?
- how complex is the arrangement of the critical curves?
- what is the complexity of the free space?

Example: an L-shaped robot moving among points

- what are the critical curves?
- how complex is the arrangement of the critical curves?
- what is the complexity of the free space?
- how to compute the free space efficiently?

END