Algorithmic Robotics and Motion Planning

Motion planning and arrangements I: General considerations

Dan Halperin
School of Computer Science
Tel Aviv University

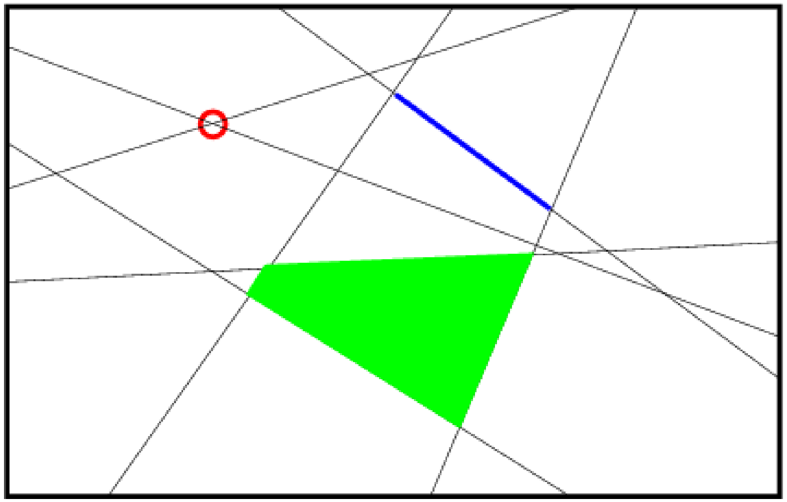
Fall 2019-2020

Overview

- Arrangements, reminder
- Arrangements and configuration spaces
- Examples
- General exact algorithms for motion planning

Reminder
What are arrangements?

Example: an arrangement of lines



vertex

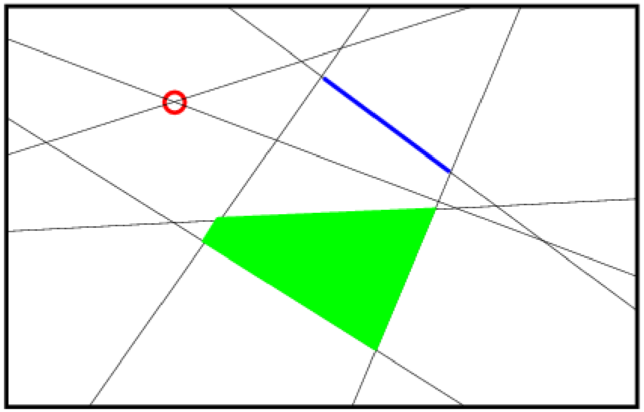
edge

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
- possibly non-linear objects (parabolas), bounded objects (segments, circles), higher dimensional (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



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: $n(n-1)/2$

number of edges: n^2

number of faces:

using Euler's formula $|V|-|E|+|F|=2$
we get $n^2 + n^2/2 + 1$

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 $\Omega(n^2)$

more generally

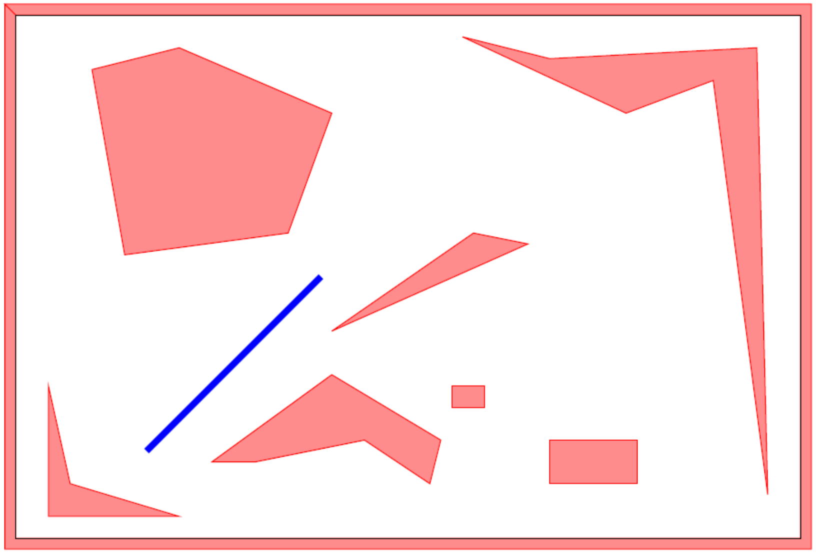
the maximum combinatorial complexity of an arrangement of n well-behaved (hyper)surfaces in $\mathbb{R}^d$ for a fixed d is $O(n^d)$; there are such arrangements whose complexity is $\Omega(n^d)$

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



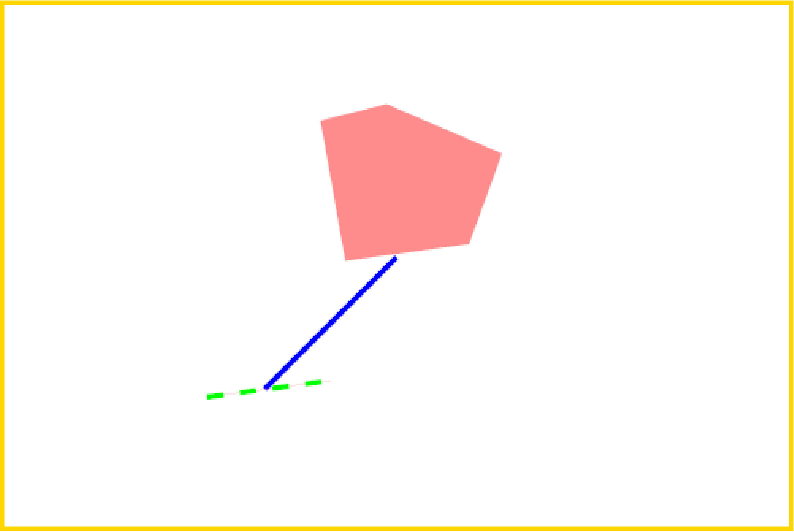
- the reference point: the lower end-point of the rod
- the configuration space is 2 dimensional

Configuration space obstacles

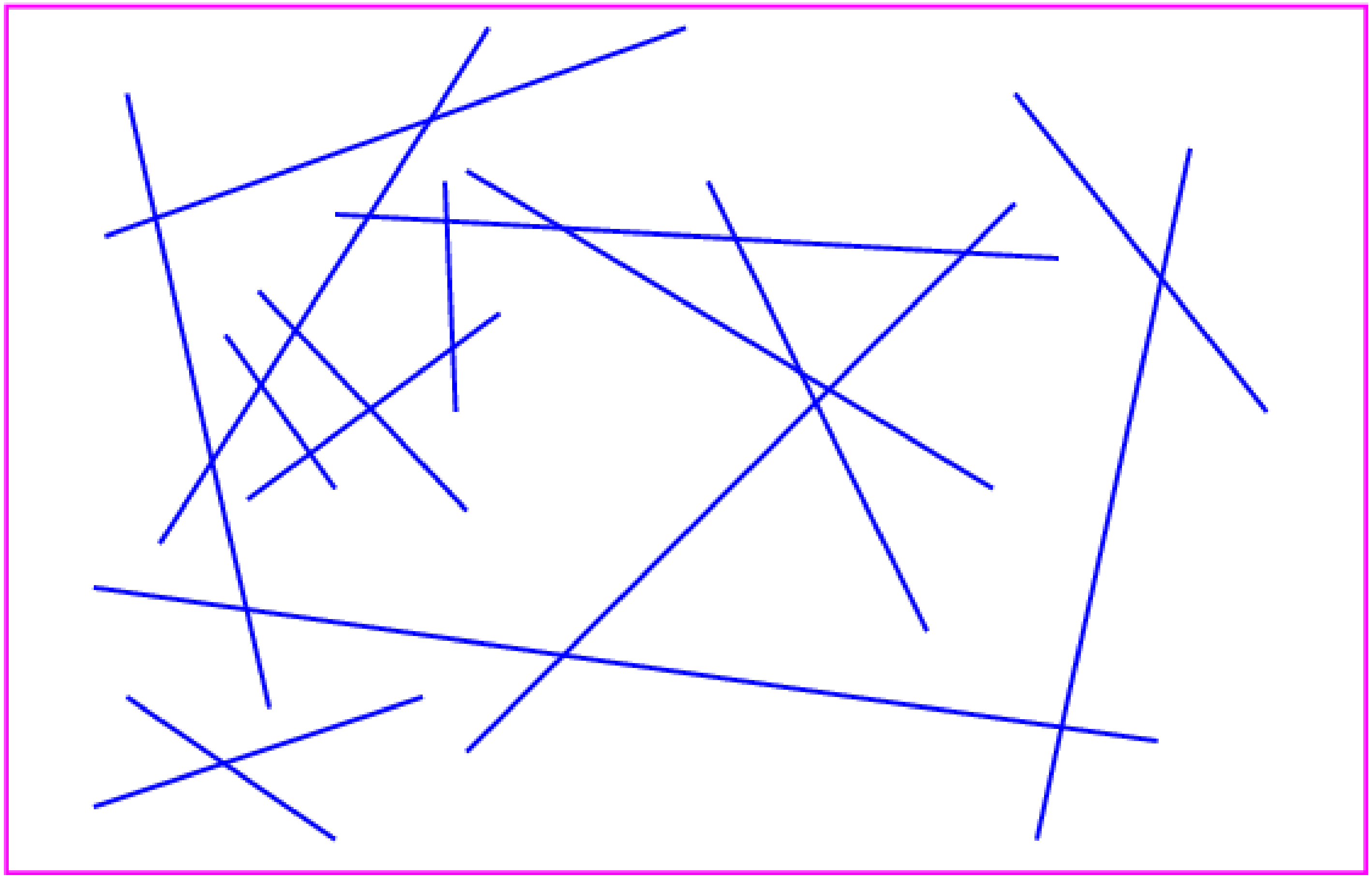
the robot has shrunk to a point
⇒
the obstacles are accordingly expanded

Critical curves in configuration space

the locus of semi-free placements



Making the connection:
The arrangement of critical curves in configuration space

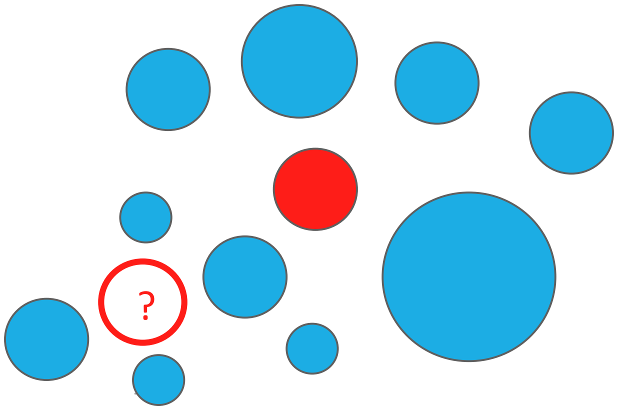


Solving a motion-planning problem
a general framework

- 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

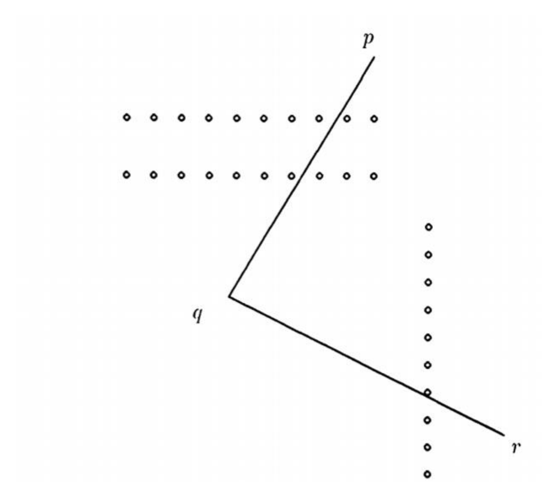
- the critical curves are circles
- how complex is the arrangement of the circles?
- what is the complexity of the free space?
- can we compute the free space efficiently?
- do we need to compute the entire free space? does it matter?



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?

- next, we let the L rotate as well
- what are the critical surfaces?
- how complex is the arrangement of the critical surfaces?
- what is the complexity of the free space?



Complete solutions, I

the Piano Movers series [Schwartz-Sharir 83],

cell decomposition: a doubly-exponential solution, $O((nd)^{3^\wedge k})$ expected time

assuming the robot complexity is constant,

k is the number of degrees of freedom,

n is the complexity of the obstacles and

d is the algebraic complexity of the problem

Complete solutions, II

roadmap [Canny 87]:
a singly exponential solution,
$n^k(\log n)d^{O(k^2)}$ expected time

see also [Basu-Pollack-Roy 06]

Bibliography

References to all the results mentioned in this presentation and more can be found in the following two chapters of the:

Handbook of Discrete and Computational Geometry —Third Edition—edited by Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth CRC Press LLC, Boca Raton, FL, 2018

- Chapter 28, Arrangements, Halperin and Sharir
- Chapter 50, Algorithmic Motion Planning, Halperin-Slazman-Sharir

THE END