CS 164 & CS 266: Computational Geometry

Lecture 14

Motion planning

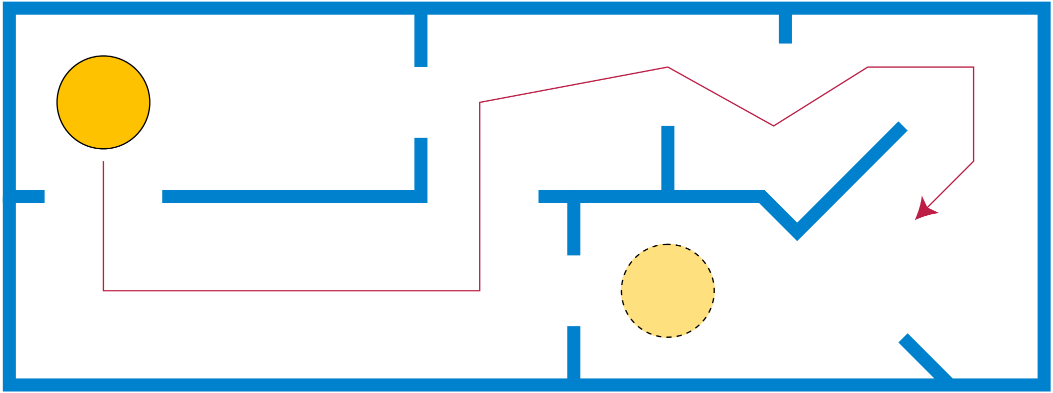
David Eppstein
University of California, Irvine

Fall Quarter, 2025

Overview

The basic problem

Input: a robot within a cluttered space



Goal: find a path to a target location

(Here: “Robot” = any movable mechanical object)

Complications

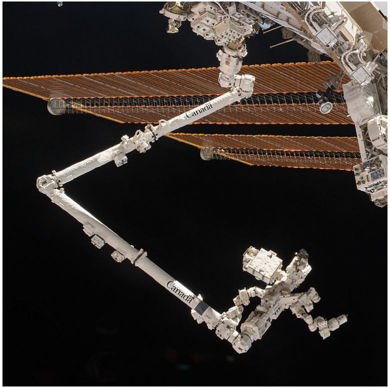
Robot might not be circular, and rotating it could help

Space is three-dimensional

Additional degrees of freedom for robot joints

Limited direction of travel

(Example: Parallel parking)



Point robots

Simplifying assumptions

The robot is just a point!

The environment is two-dimensional and polygonal

This could be a reasonable approximation for very small robots in large environments

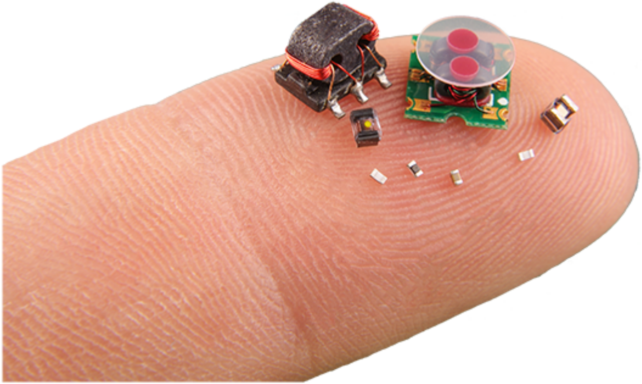
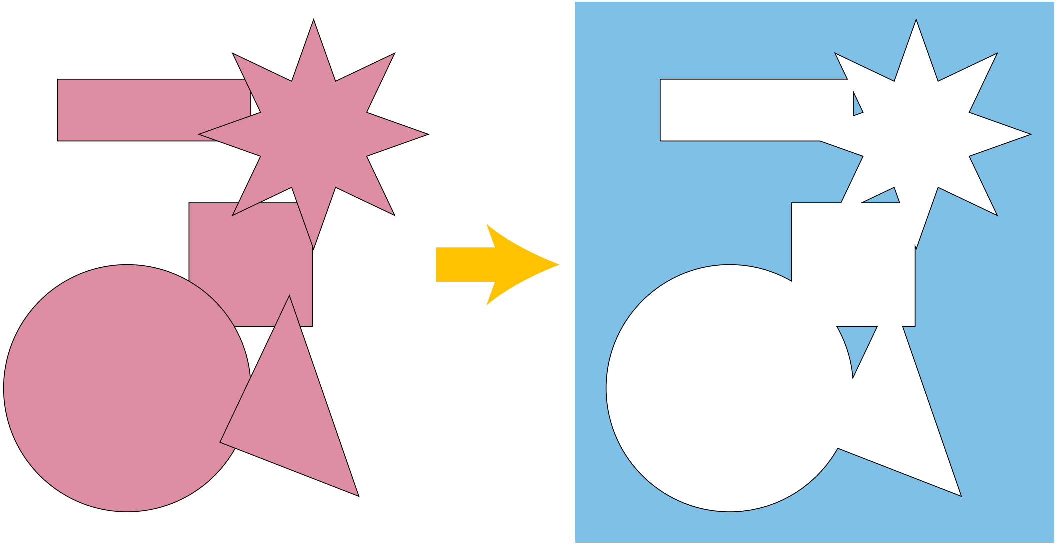


Image from https://www.darpa.mil/news-events/2018-07-17 under DARPA User Agreement

Configuration space

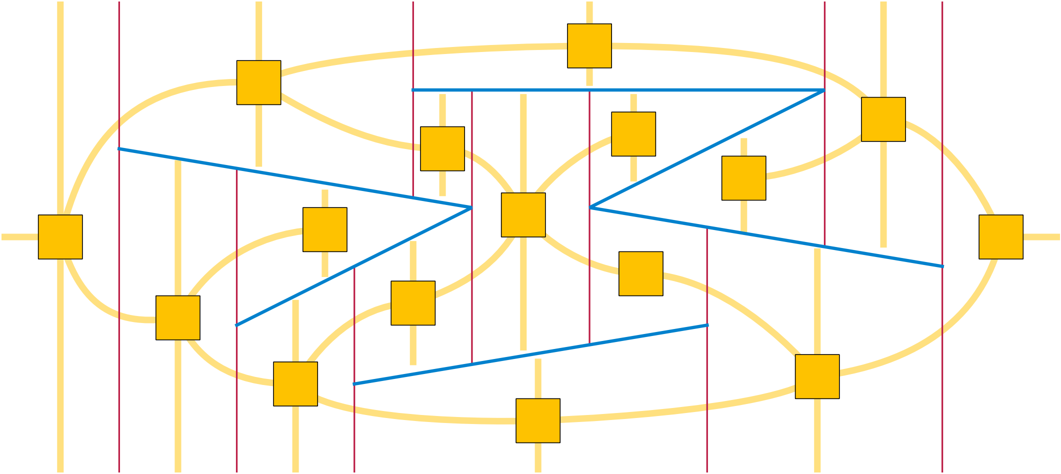
The space of places the point robot can be
$=$ the complement of the union of the obstacles



Finding a path

Assuming we don't care about the length of the path:

Find a path in the graph of adjacent trapezoids of the trapezoidal decomposition of the obstacles



Larger robots with translation

Assumptions

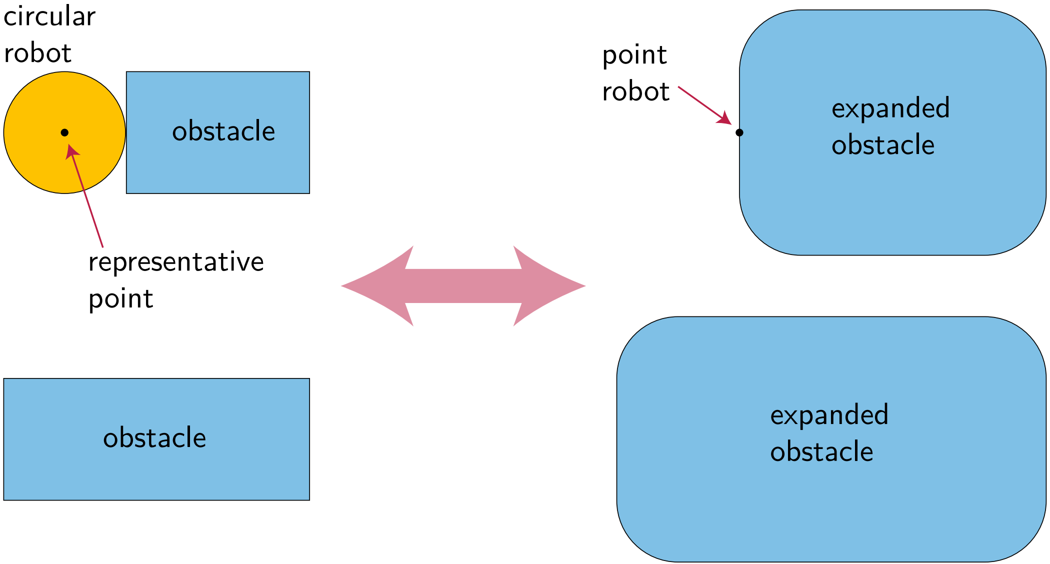
Still two-dimensional, but with nonzero size

Either rotation doesn't help (robot is circular) or is disallowed

Expanded obstacles

Choose a representative point within your robot

Free space = representative points of unobstructed placements



How to expand?

Place representative point of robot at the origin of the plane, and let R be all points covered by the robot

Let X be the set of points covered by any obstacle

Then placing the representative point at r (instead of the origin) translates all of its points: it now covers $R + r$.

We have a collision when R contains a point y such that $y + r \in X$

Equivalently, bad placements for r occur at points $X - y$

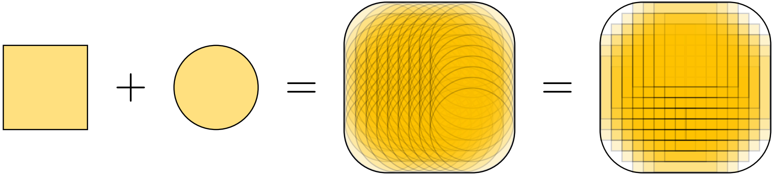
Expanded obstacle is set of all points $x - y$ for $x \in X$, $y \in R$

Minkowski sums

If X and Y are two subsets of the plane, define

$$X + Y = \{x + y \mid x \in X \text{ and } y \in Y\}$$

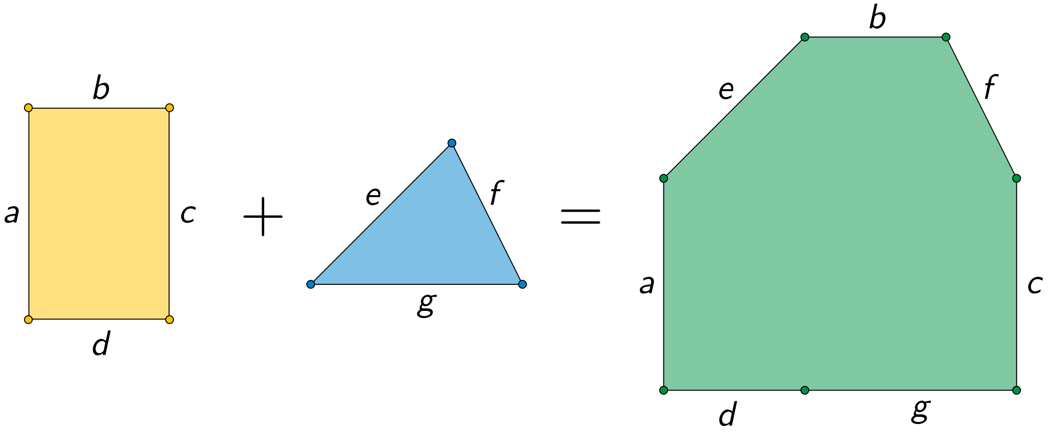
$$X - Y = \{x - y \mid x \in X \text{ and } y \in Y\}$$



Expanded obstacle = obstacle − robot

Minkowski sums of convex polygons

Merge the two cyclic sequences of polygon edges
in order by slope (adding lengths when slopes are equal)



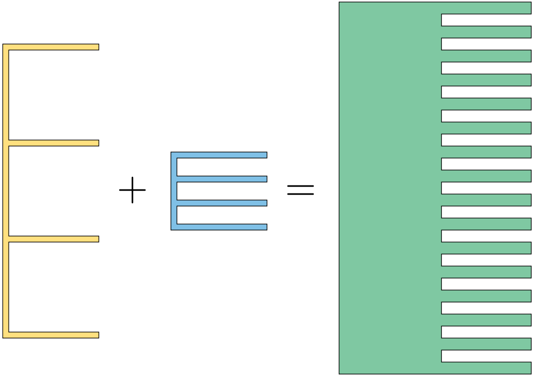
Linear time; output has complexity $O(n)$

Minkowski sums of non-convex polygons

Triangulate each polygon

Compute Minkowski sum for each pair of triangles

Form union of $O(n^2)$ convex hexagons

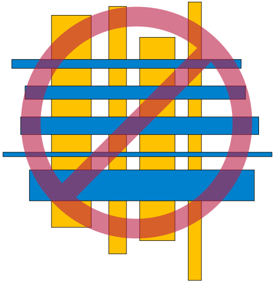


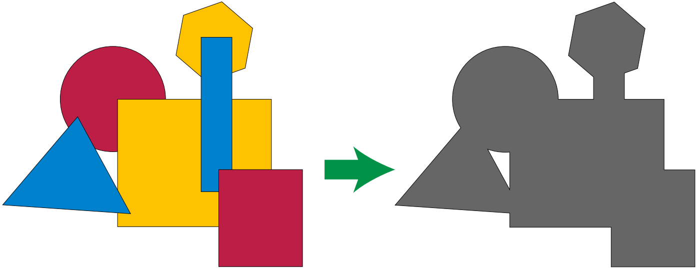
But output complexity can be $\Theta(n^2)$

Improvement for convex robots

Triangles of triangulated obstacles are non-overlapping!

When P and Q are disjoint convex sets, and R is convex,
$P+R$ and $P+R$ are pseudo-disks:
like circles, they cross each other only at two points





Union of pseudo-disks with n total vertices has complexity $O(n)$

Intuition for pseudo-disk property

If two convex obstacles are far apart, robot can pass between them $\Rightarrow$ expanded obstacles do not cross at all

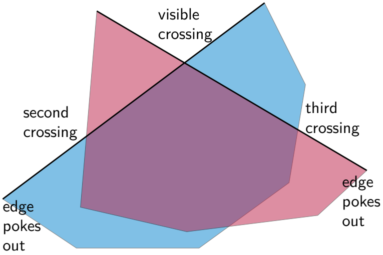
If two convex obstacles are close enough to block the corridor between them, expanded robot gets stopped partway into corridor

Two crossing points of Minkowski sums $=$ two positions at either end of corridor, where robot gets stopped

Why unions of pseudo-disks are linear

Suppose we have a union of pseudo-disks, and the pseudo-disks have n total vertices

We get a new vertex whenever the crossing point of two pseudo-disks lies on the boundary of the union



If both edges of crossing poke out of other polygons, they would cross three times

Not possible!

For one of the crossing edges, this is the closest visible crossing to its endpoint

$$\# \text{ visible crossing points} \leq 2 \# \text{ edges} = O(n)$$

To construct unions of pseudo-disks

Divide and conquer!

- Split input into two subsets of half the size
  (size = total number of vertices)
- Recursively construct their unions
- Merge using output-sensitive arrangement construction
  All crossings are visible so output complexity is $O(n)$

$$\text{time } T(n) = 2T(\frac{n}{2}) + O(n \log n) = O(n \log^2 n)$$

Overall algorithm

For motion planning with translation, for robots whose shape is a convex polygon R with $O(1)$ sides:

- Triangulate the obstacles
- Compute $\Delta_i - R$ for each triangle Δ_i
(constant time, produces convex polygon of constant complexity)
- Construct union of pseudodisks
- Construct trapezoidal decomposition of free space outside of union
- Use a graph search in the trapezoidal decomposition

Total time $O(n \log^2 n)$; slowest step is union of pseudodisks

Probabilistic roadmaps

More realistic models of motion

Translation and rotation in 3d (six degrees of freedom)

Parts of robots that can bend or rotate (more degrees of freedom)

Multiple robots working together (sum their degrees of freedom)

Every degree of freedom is a dimension of the free space

Expanded obstacles become hard-to-describe shapes with curved boundaries

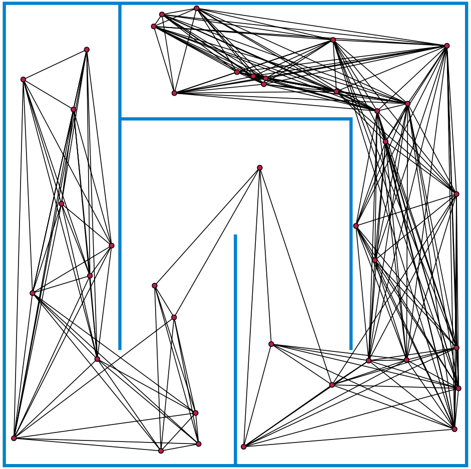
Probabilistic roadmaps

Instead of constructing whole free space in high dimensions, find a random sample of its points

Connect nearby points with edges when direct motion from one position to the other is unobstructed

Plan motions using paths in the resulting graph

[Kavraki et al. 1996]



How to sample

Easiest: choose placements randomly, test whether they are free or obstructed, keep the free ones

May oversample easy-to-route parts of the free space, undersample narrow corridors where routing is harder

Better to choose more samples near expanded-obstacle boundaries

One possibility: choose random lines, then binary search along lines for points near boundary

(⇒ uniform distribution on boundary points of free space)

[Yeh et al. 2012]

How to connect nearby points

Straight line in parameter space = vary all degrees of freedom linearly from starting to ending position

Just need to check this motion against each obstacle

Bigger issue: sampling enough points to ensure that these straight-line connections are enough to get through the whole free space

References and image credits

Kārlis Dambrāns. iRobot Roomba 870. Licensed under the Creative Commons Attribution 2.0 Generic license, May 31 2014. URL `https://commons.wikimedia.org/wiki/File:IRobot_Roomba_870_(15860914940).jpg`.

L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996. doi: 10.1109/70.508439.

Hsin-Yi Yeh, Shawna L. Thomas, David Eppstein, and Nancy M. Amato. UOBPRM: A uniformly distributed obstacle-based PRM. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 2655–2662. IEEE, 2012. doi: 10.1109/IROS.2012.6385875.