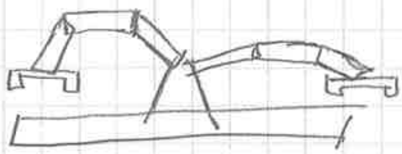
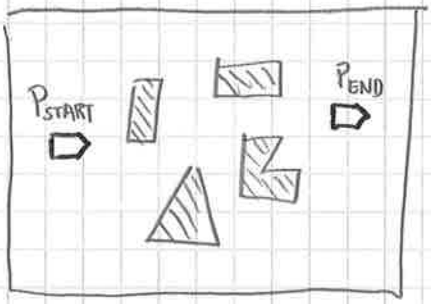
Today: Robot Motion Planning



Problem: Can a robot move from an initial position P_{START} to a final position P_{END}, without colliding with an obstacle?

If yes, compute a path.

In this lecture: simplified problem in 2D:



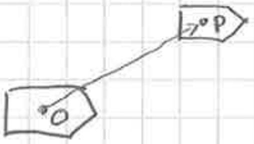
We assume:

- The set of obstacles is known. ("floor plan")
- Obstacles are polygons (and static)
- Robot can either only translate, or translate and rotate.

Configuration spaces:

Let R be a robot (a usually polygon) and $S = \{P_1, \dots, P_t\}$ be a set of obstacles (polygons)

(A) R only translates:



Define a reference point o (some interior point of R)

Define a configuration $p \in \mathbb{R}^2$ as the placement of the robot where o is translated to p.

The configuration space or working space is the set of all possible configurations: $\mathbb{R}^2$ In this case.

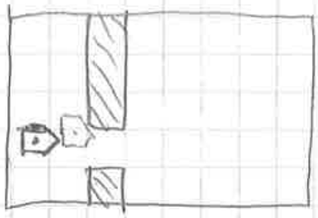
There are two types of configurations $\begin{cases} \text{Free} \\ \text{Forbidden} \end{cases}$

A configuration c is free if the robot at c does not collide with any obstacle. Otherwise, it is called forbidden

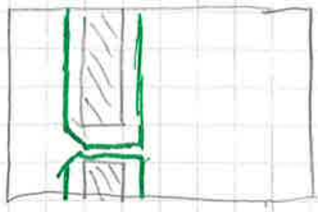
The set of all free configurations is the free space C_{FREE}

Question: How does C_{FREE} look like?
How do we compute it?

Example:

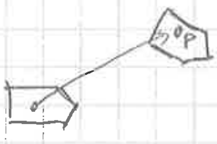


Working space.



Free space C_{FREE}

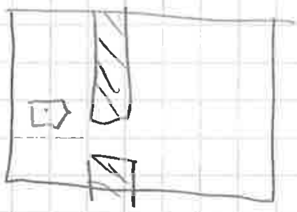
(B) Rotating and translating robot:



position encoded by reference point (translation) plus reference orientation (angle)

configuration:

(P_x, P_y, ϕ) with ϕ angle in $[0, 2\pi]$ determining ccw rotation.



Working space.

Free space $C_{FREE} \subseteq \mathbb{R} \times \mathbb{R} \times [0, 2\pi]$
with $(x,y,0) \sim (x,y,2\pi)$

↳ Interesting topological space.

Generic solution for motion planning:

(1) Compute a representation of C_{FREE}

((2) Check if P_{START} and P_{END} are free)

(3) Check whether there is a path in C_{FREE} from P_{START} to P_{END}.
If yes, return a path.

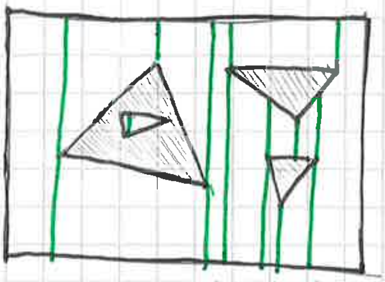
Warm up situation: Assume the robot R is just a point $\bullet$

In this case: Free space C_{FREE} = Working space.

Obstacles are polygons (maybe even with holes) with n edges in total.

We proceed as follows:

(i) Compute a trapezoidal map of the free space C_{FREE}



Free space subdivided into trapezoids.

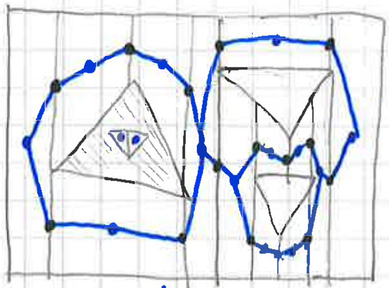
⇝ this can be done in expected $O(n \log n)$ time

see Computational Geometry book Chapter 6.1

(ii) Compute the road map: straight line graph

one node in the interior of each trapezoid ("center")
one node in each vertical boundary
edges between center and boundary vertices

⇝ $O(n)$ time



road map

(iii) Find path between P_{START} and P_{END}.

- Find trapezoids $\Delta_{START}, \Delta_{END}$ containing P_{START} and P_{END} (see Computational geometry book Chapter 6)
- If $\Delta_{START} = \Delta_{END}$, the line segment from P_{START} to P_{END} is a valid path.
- Otherwise, let C_{START}, C_{END} be the centers of $\Delta_{START}, \Delta_{END}$. Find path from C_{START} to C_{END} by graph traversal (BFS/DFS), and extend it by segments P_{START}, C_{START} and C_{END}, P_{END}

  ⇝ $O(n)$

Complexity: In $O(n \log n)$ expected time we can build a data structure such that for every pair P_{START}, P_{END} we can decide if they are connected in linear time $O(n)$, and also compute the path.

What about general polygonal robots that can only translate?

→ If we already computed C_{FREE} then the rest of the process remains the same

→ So, the problem reduces to computing the free space C_{FREE}

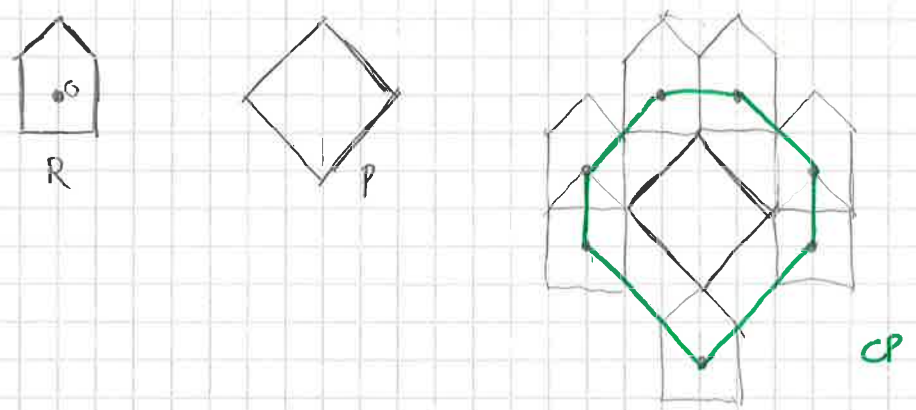
For this we use Minkowski sums.

Minkowski sums

Let R be a convex robot (convex polygon) and P be a convex obstacle (convex polygon)

Define

$$CP := \{ (x,y) : \underbrace{R(x,y)}_{\text{robot translated at position } (x,y)} \cap P \neq \emptyset \} \Rightarrow \text{"Forbidden configuration with respect to obstacle } P\text{"}$$

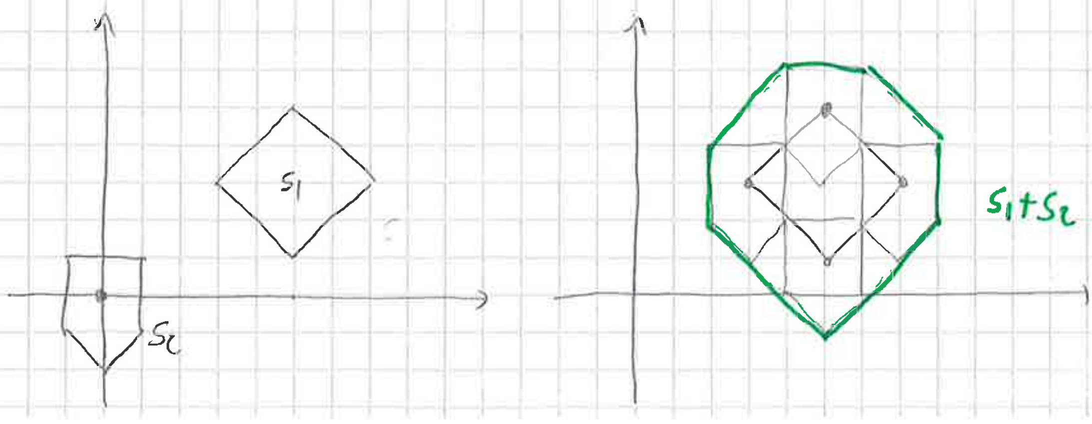


Slide R around the boundary of P. The curve traced by the reference point o is the boundary of CP.

The Minkowski sum of two sets $S_1, S_2 \subseteq \mathbb{R}^2$ is

$$S_1 \oplus S_2 := \{ p+q : p \in S_1, q \in S_2 \}.$$

Example:



Let $-S := \{ -s : s \in S \}$

Lemma: $CP = P \oplus (-R)$

Proof:

If $\vec{q} \in CP$ then $R(\vec{q})$ intersects P

So $\exists \vec{r} \in R$ and $\exists \vec{p} \in P$ such that

$$\vec{r} + \vec{q} = \vec{p}$$

Then $\vec{q} = \vec{p} - \vec{r} \in P \oplus (-R)$

Viceversa, if $\vec{q} \in P \oplus (-R)$ then $\vec{q} = \vec{p} - \vec{r}$ for some $\vec{p} \in P$ and $\vec{r} \in R$.

then

$$\vec{p} = \vec{r} + \vec{q}$$

which means that P intersects $R(\vec{q})$ ∎

Question What is the complexity of the forbidden space CP?

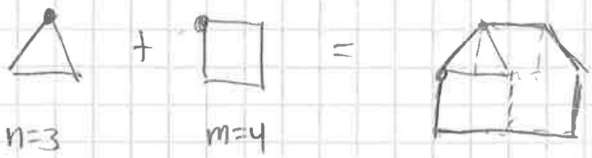
Theorem For P, R convex polygons with n, m edges, $P \oplus R$ is a convex polygon with at most $n+m$ edges, and can be computed in $O(n+m)$ time

Proof: Exercise.

Examples:

$n=3$ $m=3$ $n+m=6$

note: independent of reference points.



$n=3$ $m=4$ 6 edges $\leq n+m=7$