Motion Planning

Piotr Indyk

Piano Mover's Problem

- Given:
  – A set of obstacles
  – The initial position of a robot
  – The final position of a robot
- Goal: find a path that
  – Moves the robot from the initial to final position
  – Avoids the obstacles (at all times)

Basic notions

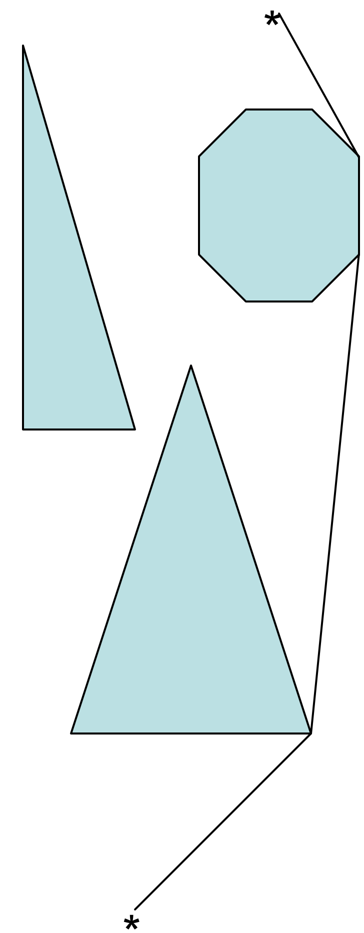
- Work space – the space with obstacles
- Configuration space:
  - The robot (position) is a point
  - Forbidden space = positions in which robot collides with an obstacle
  - Free space: the rest
- Collision-free path in the work space = path in the free part of configuration space

Demo

- http://www.diku.dk/hjemmesider/studerende/palu/start.html

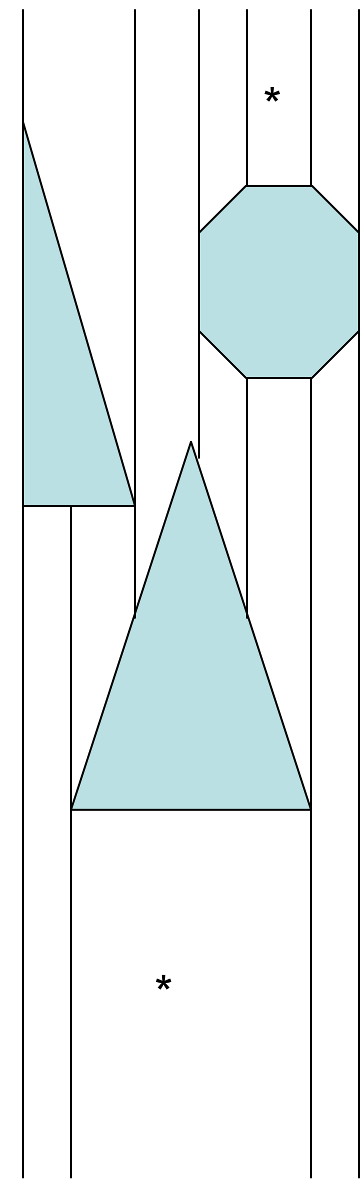
Point case

- Assume that the robot is a point
- Then the work space=configuration space
- Free space = the bounding box – the obstacles



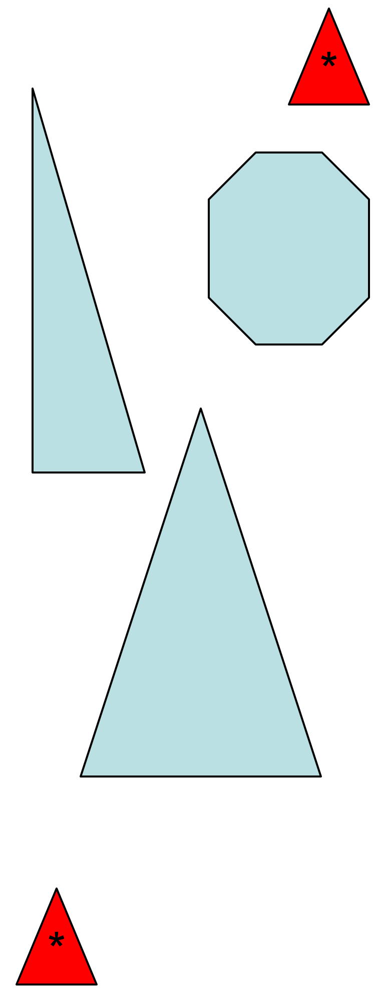
Finding a path

- Compute the trapezoidal map to represent the free space
- Place a node
  - At the center of each trapezoid
  - At each edge of the trapezoid
- Put the “visibility” edges
- Path finding=BFS in the graph



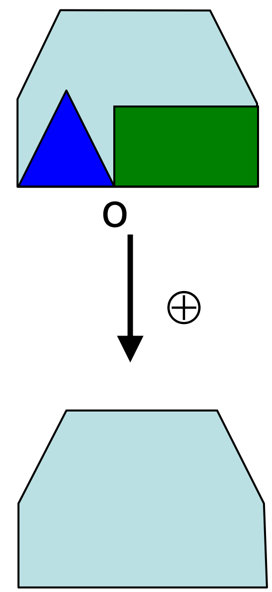
Non-point robots

- C-obstacle = the set of robot positions which overlap an obstacle
- Free space: the bounding box minus all C-obstacles
- Given a robot and obstacles, how to calculate C-obstacles ?

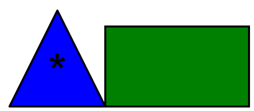


Minkowski Sum

- Minkowski Sum of two sets P and Q is defined as $P \oplus Q = \{p+q:\ p \in P,\ q \in Q\}$
- How to define C-obstacles using Minkowski Sums ?



C-obstacles



C-obstacles

- The C-obstacle of P w.r.t. robot R is equal to $P \oplus (-R)$
- Proof:
  - Assume robot R collides with P at position c
  - I.e., consider $t \in (R+c) \cap P$
  - We have $t-c \in R \rightarrow c-t \in -R \rightarrow c \in t+(-R)$
  - Since $t \in P$, we have $c \in P \oplus (-R)$
- Reverse direction is similar

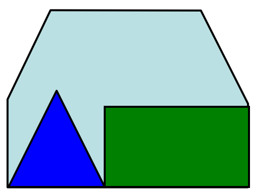
Properties of $P \oplus R$

- Assume P,R convex, with n (resp. m) edges
- Theorem: $P \oplus R$ is convex:
- Proof:
  - Consider $t_1, t_2 \in P \oplus R$. We know $t_i = p_i + r_i$ for $p_i \in P$, $r_i \in R$
  - P,Q convex: $\lambda p_1 + (1- \lambda) p_2 \in P$, $\lambda r_1 + (1- \lambda) r_2 \in R$
  - Therefore:

$$\lambda t_1 + (1- \lambda) t_2 = \lambda (p_1 + r_1) + (1- \lambda)(p_2 + r_2) \in P \oplus R$$

Properties of $P\oplus R$ II

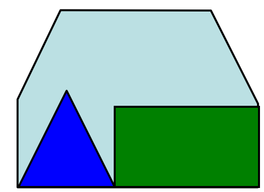
- Observation: an extreme point of $P\oplus R$ in direction d is a sum of extreme points of P and R in direction d
- Proof: for p ranging in P and r ranging in R:

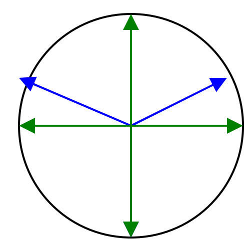


$$\max (p+r)*d$$
$$= \max p*d + r*d$$
$$= \max p*d + \max r*d$$

Properties of P⊕R III

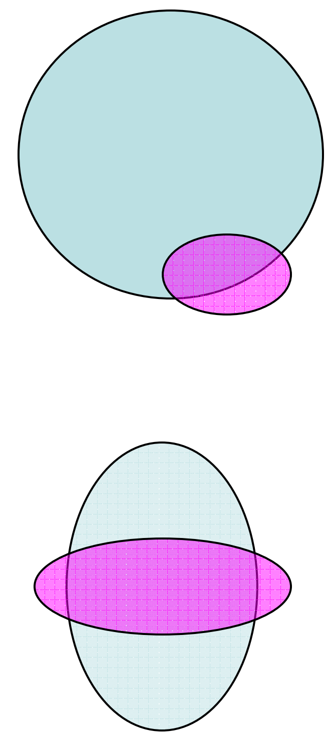
- Theorem: P⊕R has at most n+m edges.
- Proof:
  - Consider the space of directions





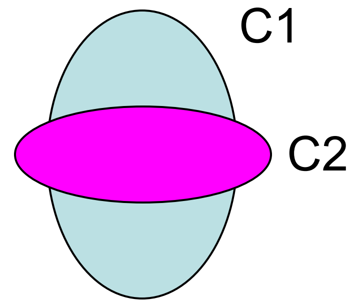
More complex obstacles

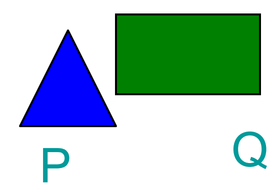
- Pseudo-disc pairs: O_1 and O_2 are in pd position, if O_1-O_2 and O_2-O_1 are connected
- At most two proper intersections of boundaries



Minkowski sums are pseudo-discs

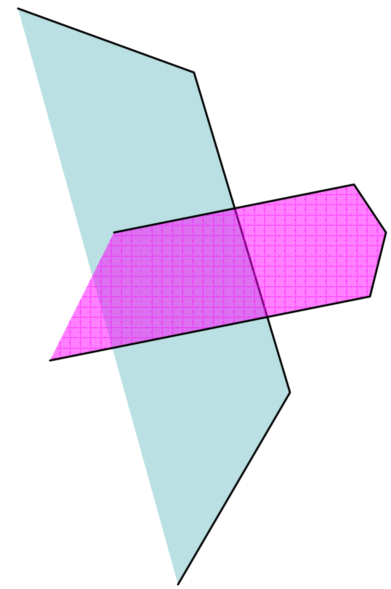
- Consider convex P,Q,R, such that P and Q are disjoint. Then $C_1=P\oplus R$ and $C_2=Q\oplus R$ are in pd position.
- Proof:
  - Consider C_1-C_2, assume it has 2 connected components
  - There are two different directions d and d' :
    - In which C_1 is more extreme than C_2
    - Somewhere in between d and d' , as well as d' and d, C_2 is more extreme than C_1
  - By properties of $\oplus$, direction d is more extreme for $C_1=P\oplus R$ than $C_2=Q\oplus R$ iff it is more extreme for P than for Q
  - Thus, there are two different directions d and d' :
    - In which P is more extreme than Q
    - Somewhere in between d and d' ,as well as d' and d, Q is more extreme than P
  - Configuration impossible for disjoint, convex P,Q





Union of pseudo-discs

- Let $P_1, \ldots, P_k$ be polygons in pd position. Then their union has complexity $|P_1| + \ldots + |P_k|$
- Proof:
  - Suffices to bound the number of vertices
  - Each vertex either original or induced by intersection
  - Charge each intersection vertex to the next original vertex in the interior of the union
  - Each vertex charged at most twice



Convex R$\oplus$ Non-convex P

- Triangulate P into $T_1,\ldots,T_n$
- Compute $R\oplus T_1,\ldots, R\oplus T_n$
- Compute their union
- Complexity: $|R|\ n$
- Similar algorithmic complexity