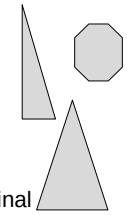
Motion Planning

Thanks to
Piotr Indyk

Piano Mover's Problem



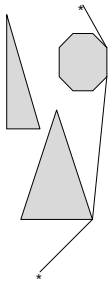
- Given:
  - A set of obstacles
  - The initial position of a robot
  - The final position of a robot
- Goal: find a path that
  - Moves the robot from the initial to final position
  - Avoids the obstacles (at all times)

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

Point case

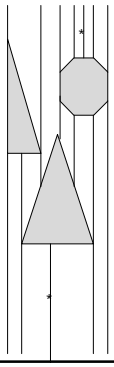
- Assume that the robot is a point
- Then the work space=configuration space
- Free space = the obstacles



Finding a path

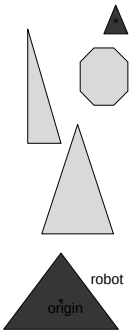
- Compute the trapezoidal map to represent the free space
- Place a node
  - At the center of each trapezoid
  - At each endpoint of each edge of the trapezoid
- Put graph edges between the vertices in the same trapezoids.
- Path finding=BFS in the graph

Note – the size of the graph is linear, but the path is probably not the shortest.



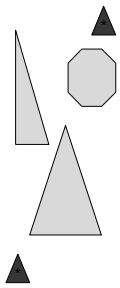
Non-point robots

- Assume a convex robot
- Assume each obstacle is convex (by triangulating the obstacles)
- We specify a point on the robot, called its **origin.**
- We specify the position of the robot by specify the location of the origin



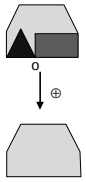
Non-point robots - cont

- 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 ?



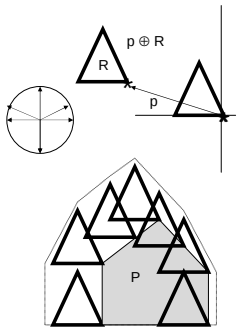
Minkowski Sum

- Minkowski Sum of two sets P and Q is defined as $P \oplus Q = \{p+q: p \in P, q \in Q\}$



Properties of P⊕R

- $P \oplus R = \{p+r: p \in P, r \in R\}$
- Theorem: If P⊕R has m and n edges, and they are convex, then P⊕R has at most n+m edges.
- Proof:
  - P⊕R is convex (next slide)
  - Consider the space of directions – each edge of P⊕R is paralel to either and edge of P or an edge of R.

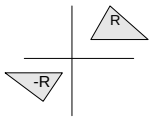


Convexity

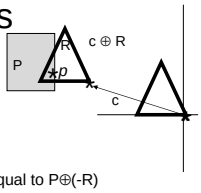
- Assume P,R convex, with n (resp. m) edges
- Theorem: P⊕R is convex:
- Proof:
  - Consider $t_1, t_2 \in P \oplus R$. We know $t_i = p_i + r_i$ for $p_i \in P$, $r_i \in R$
  - P,Q convex: $\lambda p_1 + (1-\lambda) p_2 \in P$, $\lambda r_1 + (1-\lambda) r_2 \in R$
  - Therefore:

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

R vs (-R)



C-obstacles



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

More Properties of P⊕R

A point $p \in Q$ is **extrme** (I.e. corner of Q) if there is some vercor (direction) d such that $p*d = max\ \{ q*d \mid q \in Q\}$

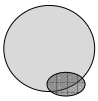
- Observation: an extreme point of P⊕R in direction d is a sum of extreme points of P and R in direction d
- Proof: for p ranging in P and r ranging in R:

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

More complex obstacles

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

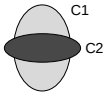
Yes



No

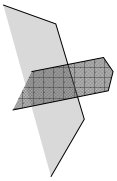
Minkowski sums are pseudo-discs

- Consider convex P,Q,R, such that P and Q are disjoint. Then C_1=P⊕R and C_2=Q⊕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', C_2 is more extreme than C_1.
  - By properties of ⊕, direction d is more extreme for C_1=P⊕R than C_2=Q⊕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



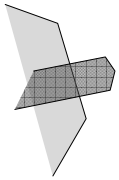
Union of pseudo-discs

- 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



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



Ananlysis: Convex R⊕ Non-convex P

- Given $|P|=n$, $|R|=m$
- Triangulate P into $T_1,\ldots,T_n$. Time O(n log n)
- Compute $R\oplus T_1,\ldots, R \oplus T_n$ Time O(nm)
- Compute their union $O(mn \log^2 (mn))$:
  - divide-and-conqure+line sweep,
  - similar to computing the union of squres shown in hw
  - (can be done faster)
- Trapezoidatoin, and compute the graph and finding a path - Complexity: $O(mn \log (mn))$

Higher dim – randomized planner

- Usually the complexity of the free space for a robot with *d* degrees of freedom in an environment of complexity n is $\Theta(n^d)$
- It is not practicle to construct the free space.
- Instead, we (very raughly) do
  - create a sample *S* of positions of *R*
  - For each position, check if is free. If yes, it is a node of the graph.
  - For every pair of free positions, chech if the segment connecting them is free. If yes connect them by an edge.
  - Find a path from *s* to *t* in this graph.

- Works well in practice
- Problem: narrow passage.
- Applicatoin (one of many): protein docking.