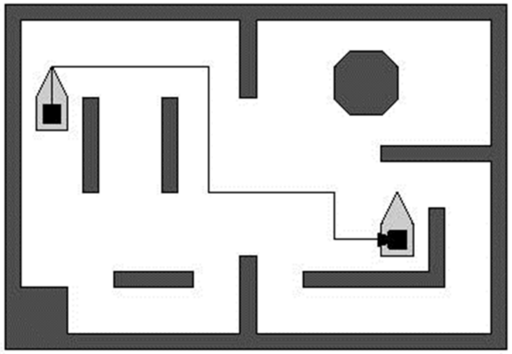
CMPS 3130/6130 Computational Geometry
Spring 2015

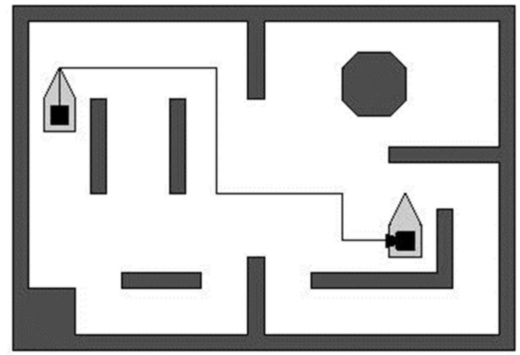


Motion Planning

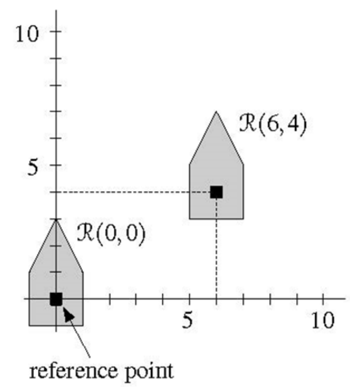
Carola Wenk

Robot motion planning

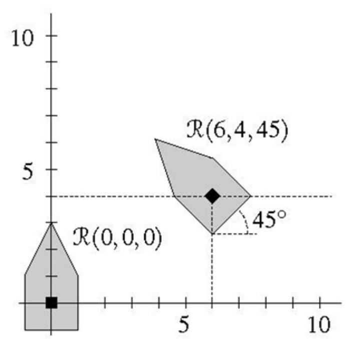
- **Given:** A floor plan (2d polygonal region with obstacles), and a robot (2D simple polygon)
- **Task:** Find a collision-free path from start to end



- Robot R is a simple polygon; $R=R(0,0)$
- Let $R(x,y) = R+\binom{x}{y}$
  R translated by $\vec{x} = \binom{x}{y}$

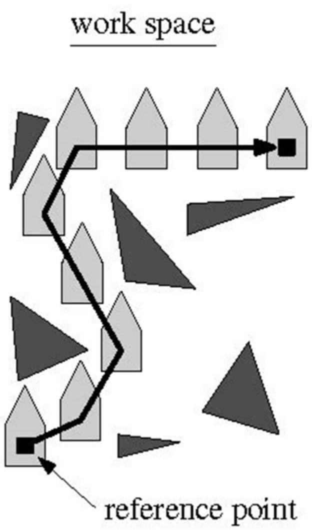


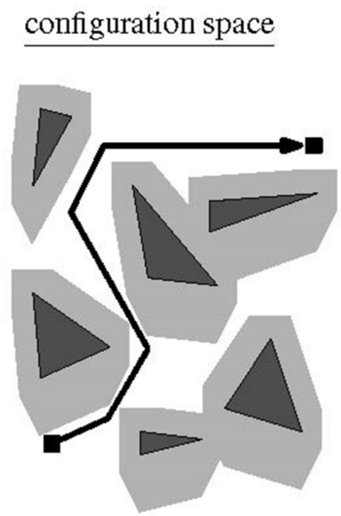
- Add rotations: $R(x,y,\varphi)$

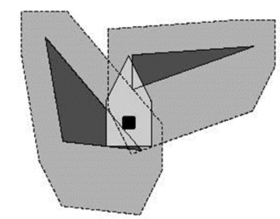


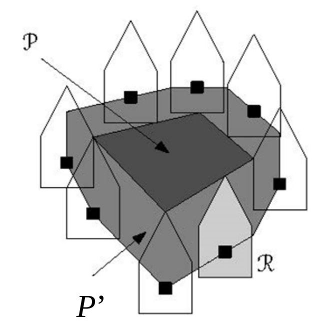
Configuration space

- # parameters = degrees of freedom (DOF)
- Parameter space = “Configuration space” **C**:
  - 2D translating: configuration space is **C** = **R**^{2}
  - 2D translating and rotating: configuration space is **C** = **R**^{2} x $[0,2\pi)$





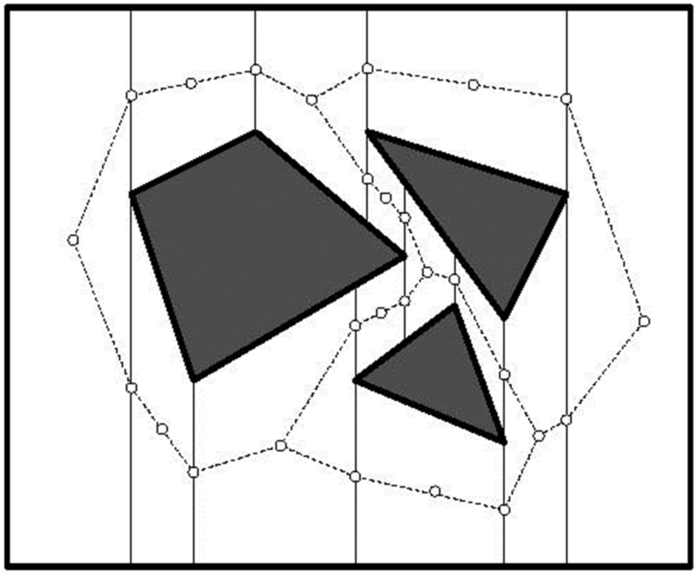




- Obstacle P. Its corresponding **C-obstacle** P’=$\{\vec{x} \in \mathbf{C} \mid R(\vec{x}) \cap P \neq \emptyset\}$
- Free space: Placements (= subset of configuration space) where robot does not intersect any obstacle.

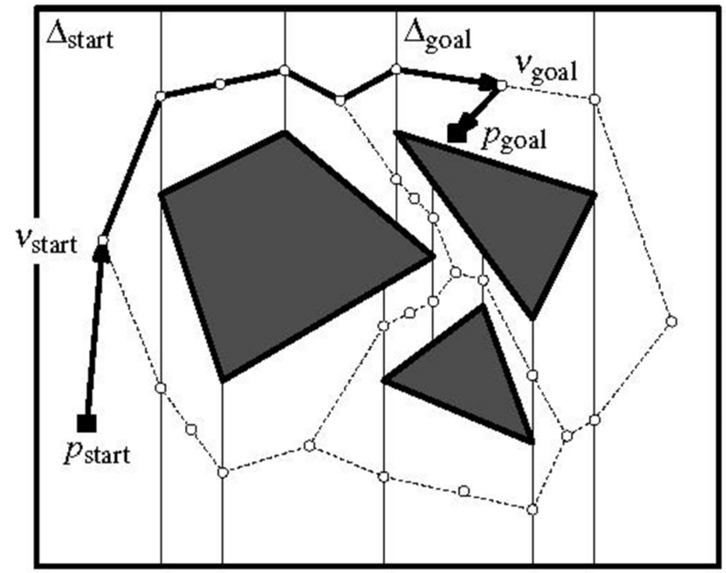
Translating a point robot

- Work space = configuration space
- Compute trapezoidal map of disjoint polygonal obstacles in $O(n \log n)$ expected time (where n = total # edges), including point location data structure
- Construct road map in trapezoidal map:
  - One vertex on each vertical edge
  - One vertex in center of each trapezoid
  - Edges between center-vertex and edge-vertex of same trapezoid
  - $O(n)$ time and space



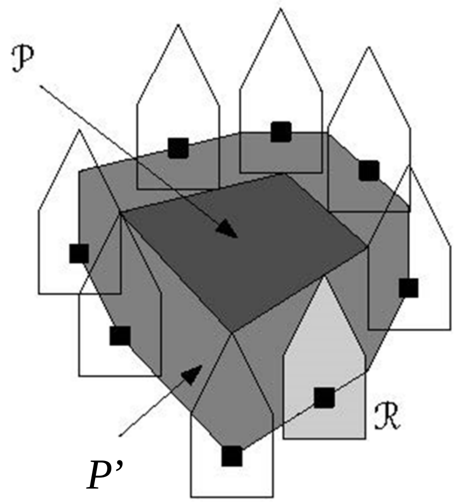
Translating a point robot

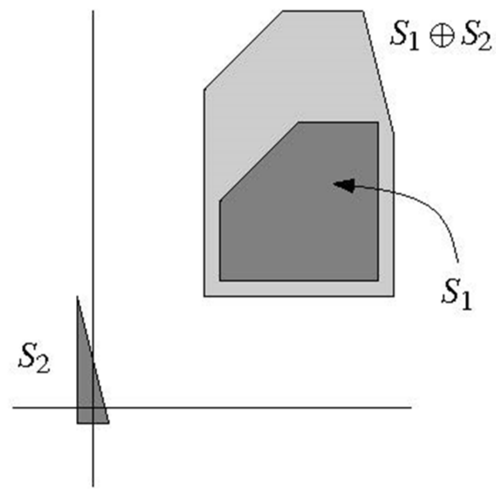
- Work space = configuration space
- Compute trapezoidal map of disjoint polygonal obstacles in O(n log n) expected time (where n = total # edges), including point location data structure
- Construct road map in trapezoidal map:
  - One vertex on each vertical edge
  - One vertex in center of each trapezoid
  - Edges between center-vertex and edge-vertex of same trapezoid
  - O(n) time and space
- Compute path:
  - Locate trapezoids containing start and end
  - Traverse this road map using DFS or BFS to find path from start to end



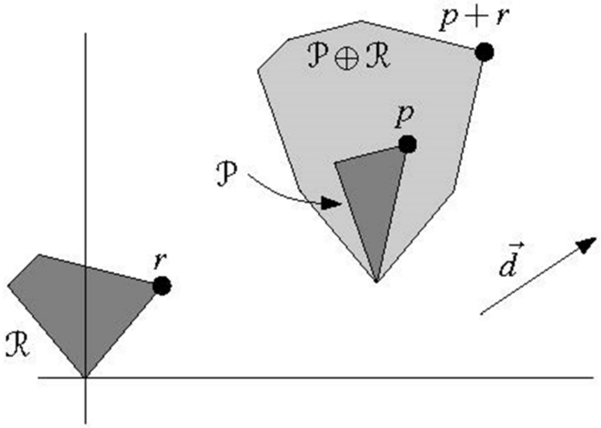
Theorem: One can preprocess a set of obstacles (with n = total # edges) in O(n log n) expected time, such that for any (start/end) query a collision-free path can be computed in O(n) time.

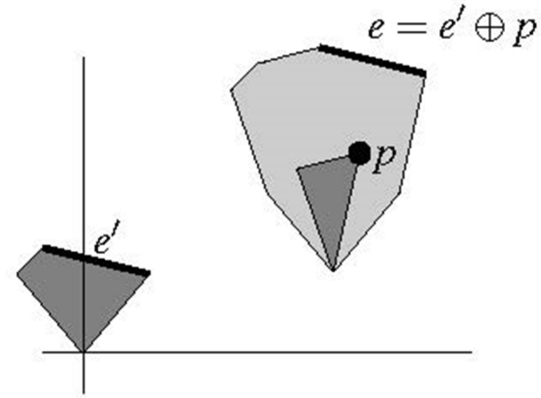
Minkowski sums



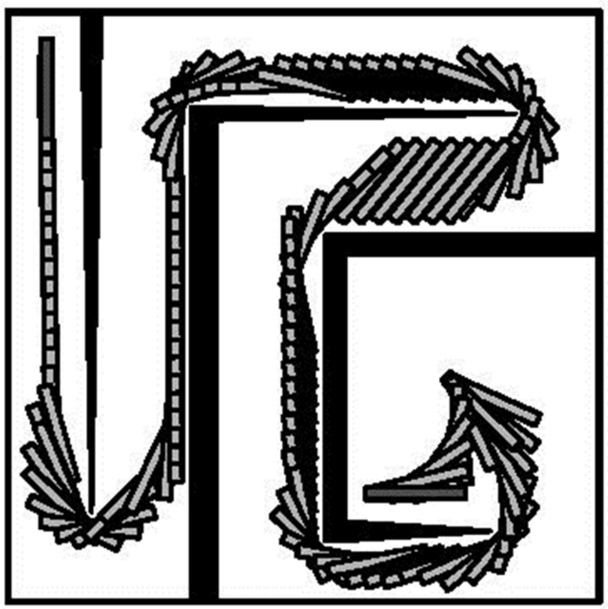


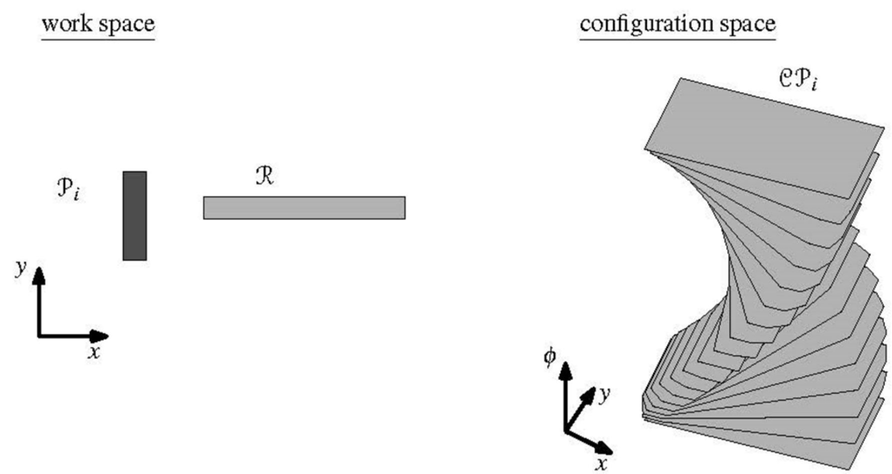
Extreme points



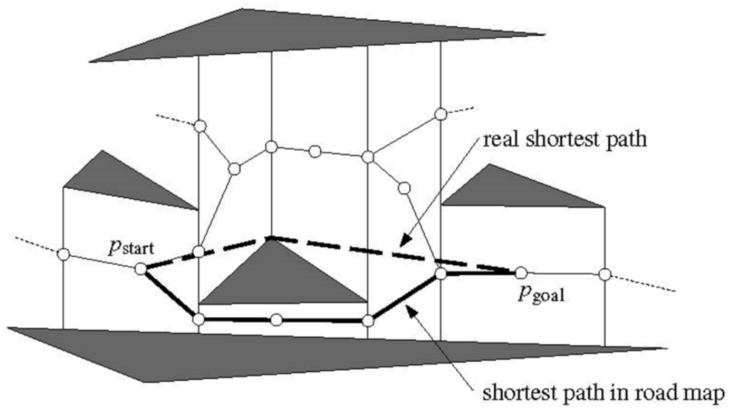


Configuration space with translations and rotations

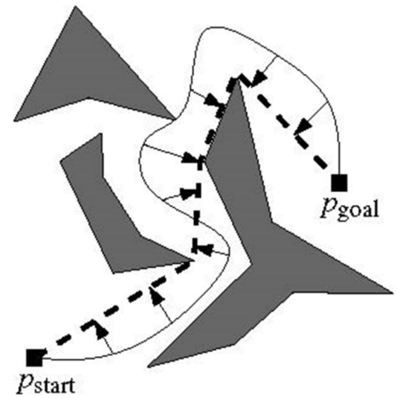




Shortest path for robot



Pull rubber band tight:



Visibility graph