Motion planning

ST5 Autonomous robotics

Francis Colas

2022-09-23

Introduction

Motion

- generated by motors
  - electric (AC/DC), pneumatic or hydraulic
  - stepper motor: fixed positions
  - servomotors: motor+integrated sensor for position control
- controlled to perform a given trajectory

Introduction

Motion

- generated by motors
  - electric (AC/DC), pneumatic or hydraulic
  - stepper motor: fixed positions
  - servomotors: motor+integrated sensor for position control
- controlled to perform a given trajectory

Motion planning

- compute trajectory
- to reach a defined goal
- while following constraints (collision avoidance, dynamics...)

Introduction

Motion

- generated by motors
  - electric (AC/DC), pneumatic or hydraulic
  - stepper motor: fixed positions
  - servomotors: motor+integrated sensor for position control
- controlled to perform a given trajectory

Motion planning

- compute trajectory
- to reach a defined goal
- while following constraints (collision avoidance, dynamics...)

Aim of the session

- configuration space
- planning algorithms

01

Configuration space

Trajectory

Path

- sequence of poses
- typically from a start to a goal location

Trajectory

Path

- sequence of poses
- typically from a start to a goal location

Trajectory

- mapping from time to configuration
- allows to compute velocities and commands

Trajectory

Path

- sequence of poses
- typically from a start to a goal location

Trajectory

- mapping from time to configuration
- allows to compute velocities and commands

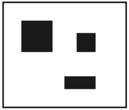
Constraints

- minimum length/distance
- minimum cost (application-dependent)
- security/distance to obstacles
- kinematics or dynamics

Configuration space

Workspace $\mathcal{W}$

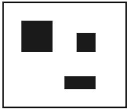
- space in which robot evolves
  - in general: $\mathcal{W} = \mathbb{R}^3$
  - sometimes: $\mathcal{W} = \mathbb{R}^2$



Configuration space

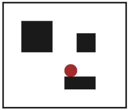
Workspace $\mathcal{W}$

- space in which robot evolves
  - in general: $\mathcal{W} = \mathbb{R}^3$
  - sometimes: $\mathcal{W} = \mathbb{R}^2$



Configuration space $\mathcal{C}$

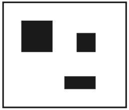
- set of feasible configurations $\boldsymbol{q} \in \mathcal{C}$
- taking into account physical constraints
- in general $\mathcal{C} \neq \mathcal{W}$



Configuration space

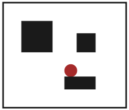
Workspace $\mathcal{W}$

- space in which robot evolves
  - in general: $\mathcal{W} = \mathbb{R}^3$
  - sometimes: $\mathcal{W} = \mathbb{R}^2$



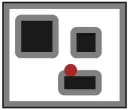
Configuration space $\mathcal{C}$

- set of feasible configurations $\boldsymbol{q} \in \mathcal{C}$
- taking into account physical constraints
- in general $\mathcal{C} \neq \mathcal{W}$



Free space $\mathcal{E}$

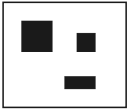
- configurations not in collision
- $\mathcal{E} = \mathcal{C} \setminus \mathcal{O}$
- adapted for planning



Configuration space

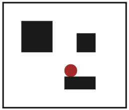
Workspace $\mathcal{W}$

- space in which robot evolves
  - in general: $\mathcal{W} = \mathbb{R}^3$
  - sometimes: $\mathcal{W} = \mathbb{R}^2$



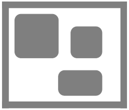
Configuration space $\mathcal{C}$

- set of feasible configurations $\boldsymbol{q} \in \mathcal{C}$
- taking into account physical constraints
- in general $\mathcal{C} \neq \mathcal{W}$



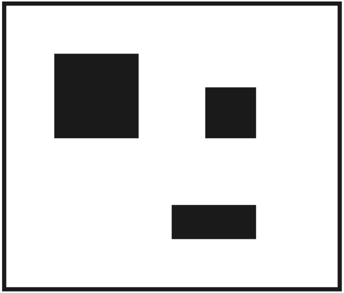
Free space $\mathcal{E}$

- configurations not in collision
- $\mathcal{E} = \mathcal{C} \setminus \mathcal{O}$
- adapted for planning



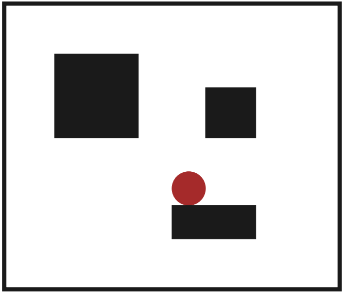
Holonomic circular mobile robot

Workspace



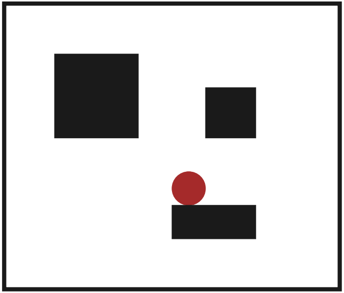
Holonomic circular mobile robot

Workspace and robot

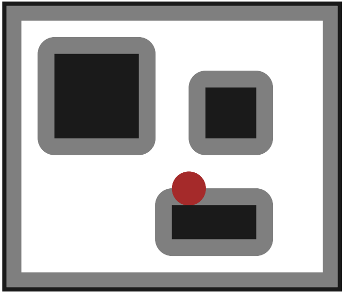


Holonomic circular mobile robot

Workspace and robot

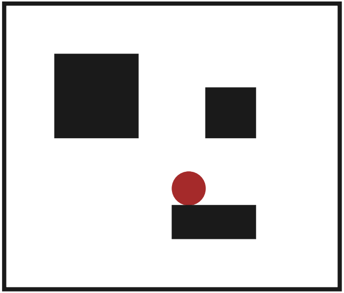


Obstacle inflation (Minkowski sum)

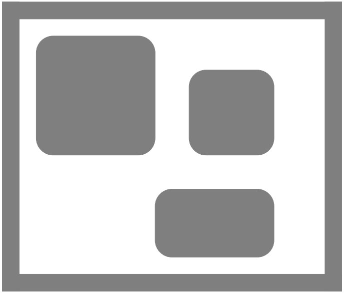


Holonomic circular mobile robot

Workspace and robot

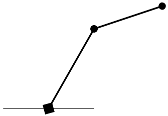


Free space



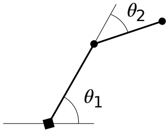
Robotic arm

Robotic arm

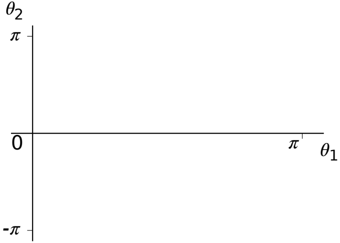


Robotic arm

2 degrees-of-freedom arm

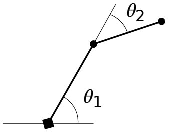


Degrees of freedom

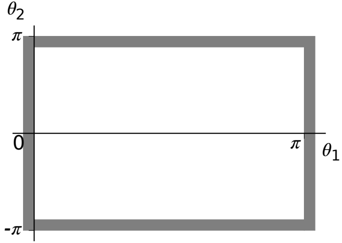


Robotic arm

2 degrees-of-freedom arm

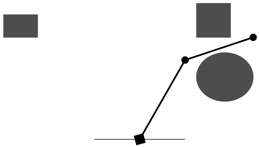


Configuration space

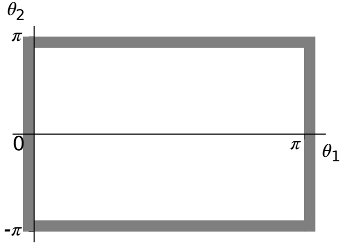


Robotic arm

Workspace

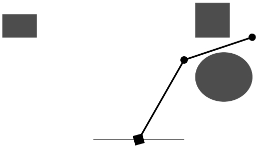


Configuration space

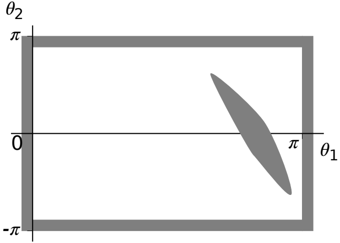


Robotic arm

Workspace

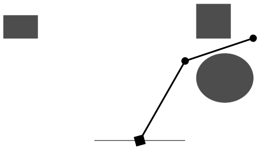


First obstacle

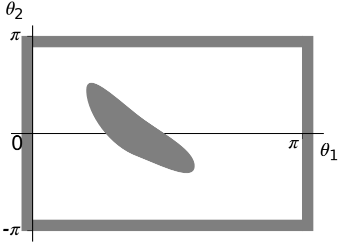


Robotic arm

Workspace

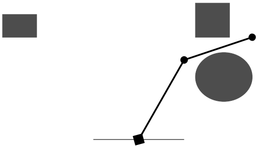


Second obstacle

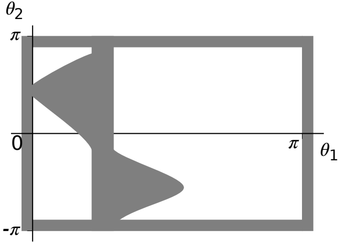


Robotic arm

Workspace

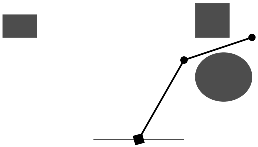


Third obstacle

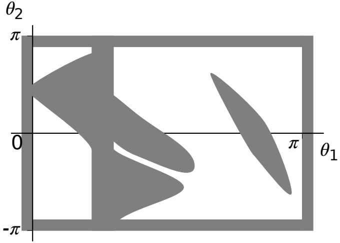


Robotic arm

Workspace

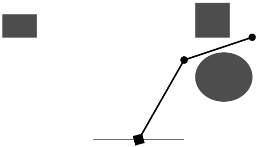


Free space

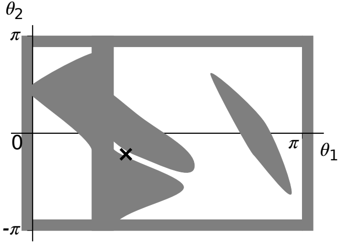


Robotic arm

Workspace



Free space



Conclusion on configuration space

Configuration space

- unification of mobile robots and robotic arms
- free space (no collision)
- adapted for planning

Building

- Minkowski sum for a mobile robot
- collision test for an arm

Limits

- only geometric constraints
- no kinematics nor dynamics
- potentially high-dimension space

02

Planning algorithms

Algorithms

Approaches

- space decomposition
  - grid
  - cell decomposition
- sampling
- potential fields
- geometric resolution
- path refinement

Path planning with a grid

Path planning with a grid

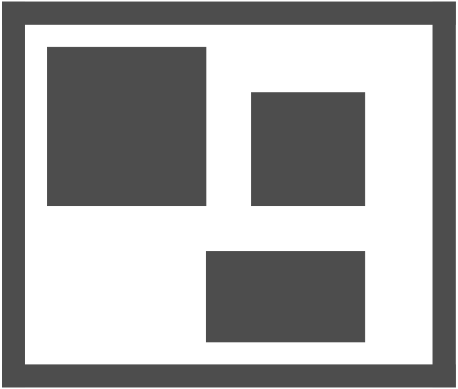
- adapted to occupancy grids
- neighborhood graph
- graph search (Dijkstra, A*)

Result

- path
- discretized orientation
- not necessarily optimal in distance
- high complexity in medium to high dimension

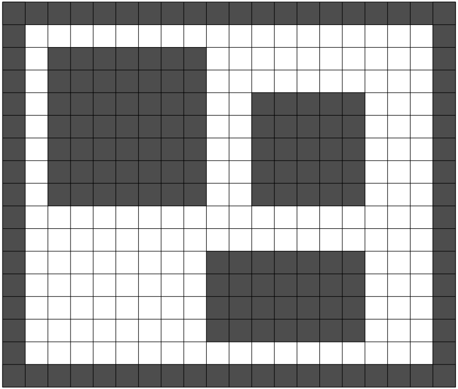
Example

Free space



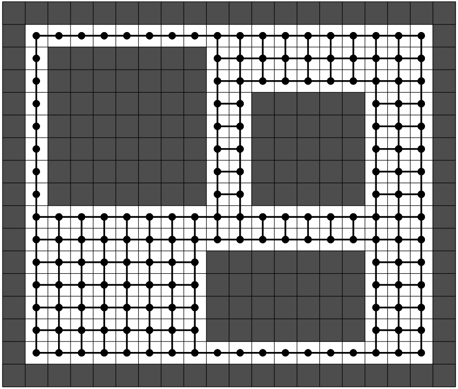
Example

Grid decomposition



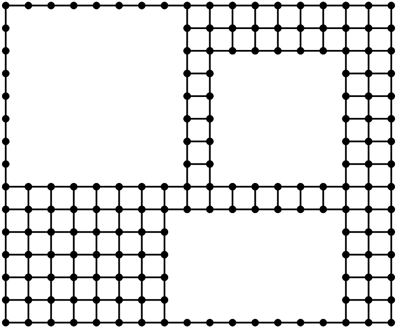
Example

Cell graph



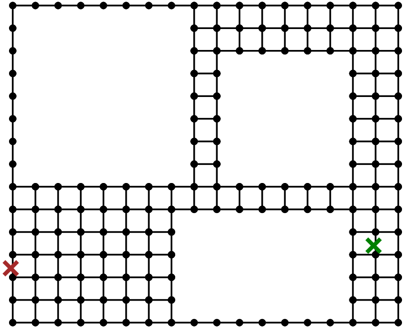
Example

Cell graph



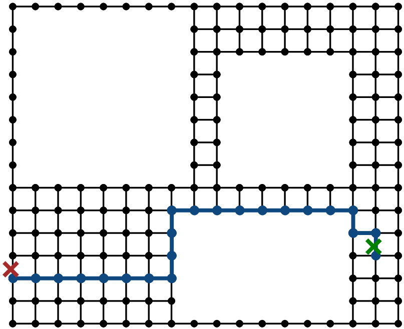
Example

Start and goal locations



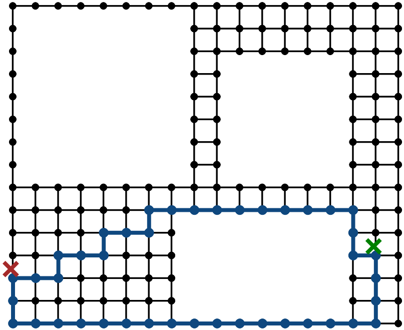
Example

Graph search (A^* or other)



Example

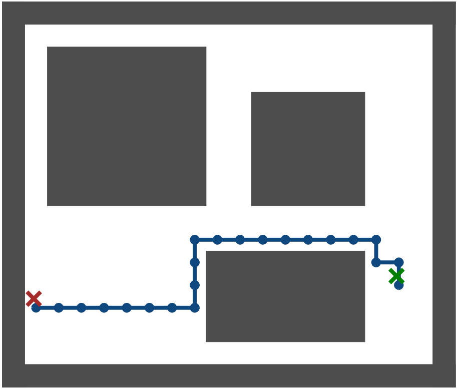
Graph search (A* or other)



Shortest path not unique!

Example

Result



Planning with cell decomposition

Cell decomposition

- various tessellation methods

Voronoi diagram

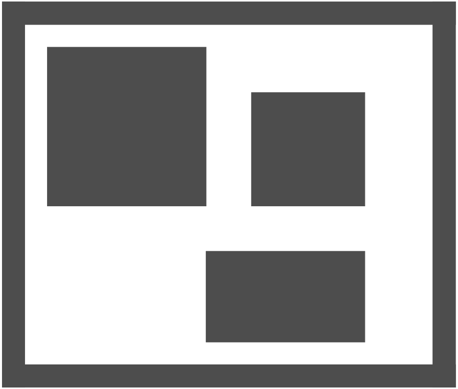
- tiling based on distance to obstacles
- dual of Delaunay triangulation
- path following of the edges of the cells

Result

- path
- as far from the obstacles as possible
- difficult to build in high dimensions
- not distance optimal

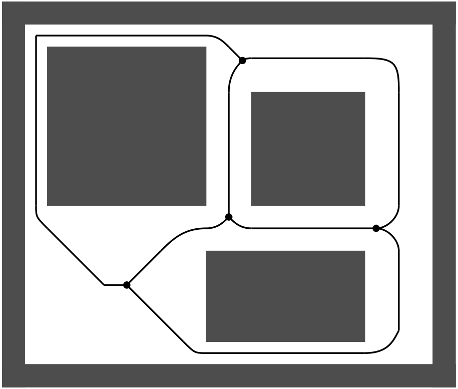
Example

Free space



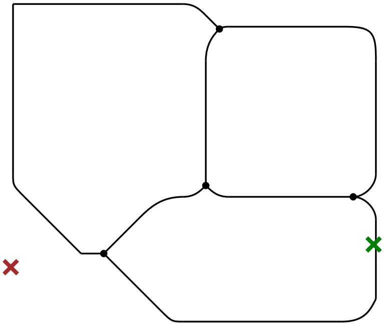
Example

Voronoi diagram



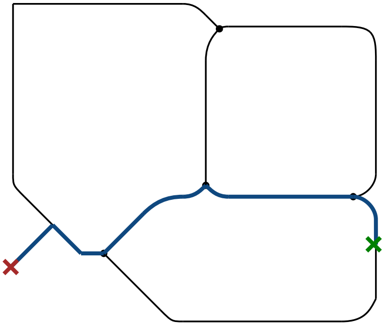
Example

Voronoi diagram



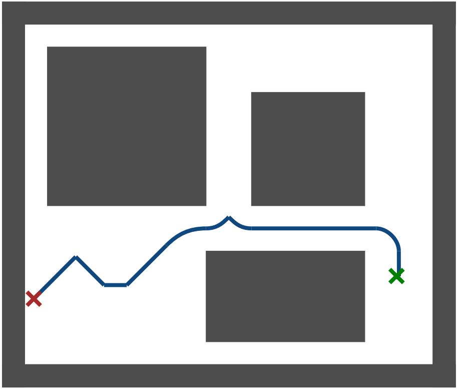
Example

Graph search



Example

Result



Sampling-based planning

Rapidly-expanding Random Trees

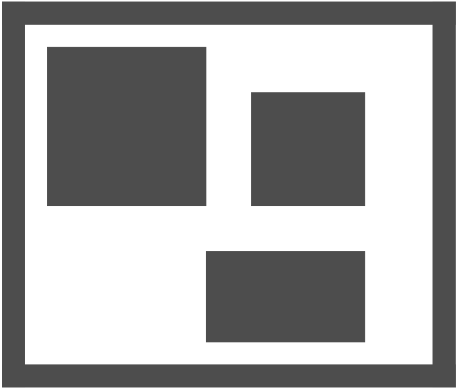
- RRT, RRT*...
- stochastic algorithm by sampling the space
- connection tree building
- until goal is found
- refinement of the path

Result

- path
- distance optimal for infinite samples
- quick and anytime as soon as the path is found

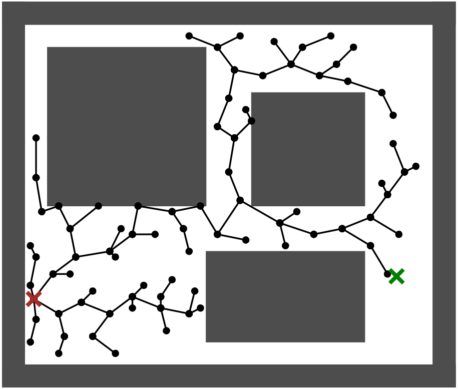
Example

Free space



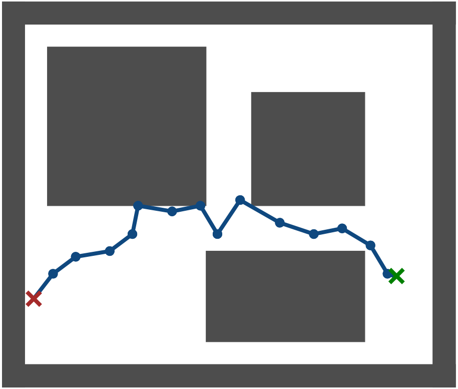
Example

Graph expansion



Example

Result



Algorithm

RRT

```
$V \leftarrow \{x_{\text{init}}\}; E \leftarrow \emptyset$
for $i = 1, \ldots, n$ do
    $x_{\text{rand}} \leftarrow \texttt{SampleFree}()$
    $x_{\text{nearest}} \leftarrow \texttt{Nearest}\left(G = (V, E), x_{\text{rand}}\right)$
    $x_{\text{new}} \leftarrow \texttt{Steer}\left(x_{\text{nearest}}, x_{\text{rand}}\right)$
    if $\texttt{CollFree}\left(x_{\text{nearest}}, x_{\text{new}}\right)$ then
        $V \leftarrow V \cup \{x_{\text{new}}\}$
        $E \leftarrow E \cup \{(x_{\text{nearest}}, x_{\text{new}})\}$
    end if
end for
return $G = (V, E)$
```

Functions

- $\texttt{SampleFree}()$: sample point in free space
- $\texttt{Nearest}(G, x)$: point in G nearest from x
- $\texttt{Steer}(x_1, x_2)$: point toward x_2 at a given distance from x_1
- $\texttt{CollFree}(x_1, x_2)$: no obstacle between x_1 and x_2

Algorithm

Improvement of RRT

```
$V \leftarrow \{x_{\text{init}}\}; E \leftarrow \emptyset$
for $i = 1, \dots, n$ do
    $x_{\text{rand}} \leftarrow \texttt{SampleFree}()$
    $x_{\text{nearest}} \leftarrow \texttt{Nearest}\left(G = (V, E), x_{\text{rand}}\right)$
    $x_{\text{new}} \leftarrow \texttt{Steer}\left(x_{\text{nearest}}, x_{\text{rand}}\right)$
    if $\texttt{CollFree}\left(x_{\text{nearest}}, x_{\text{new}}\right)$ then
        $X_{\text{near}} \leftarrow \texttt{Near}\left(G = (V, E), x_{\text{new}}, \delta\right)$
        $x_{\text{min}} \leftarrow \arg\min_{x \in X_{\text{near}}} \mathrm{C}(x) + c(x, x_{\text{new}})$
        $V \leftarrow V \cup \{x_{\text{new}}\}$
        $E \leftarrow E \cup \{(x_{\text{min}}, x_{\text{new}})\}$
    end if
end for
return $G = (V, E)$
```

Functions

- $\texttt{SampleFree}()$: sample point in free space
- $\texttt{Nearest}(G, x)$: point in G nearest from x
- $\texttt{Steer}(x_1, x_2)$: point toward x_2 at a given distance from x_1
- $\texttt{CollFree}(x_1, x_2)$: no obstacle between x_1 and x_2
- $\texttt{Near}(G, x, d)$: points in G at a distance from x less than d
- $\mathrm{C}(x)$: cost between x_{init} and x, walking up the graph
- $c(x_1, x_2)$: cost between x_1 and x_2

Algorithm

RRT*

```
$V \leftarrow \{x_{\text{init}}\}; E \leftarrow \emptyset$
for $i = 1, \ldots, n$ do
    $x_{\text{rand}} \leftarrow \texttt{SampleFree}()$
    $x_{\text{nearest}} \leftarrow \texttt{Nearest}\left(G = (V, E), x_{\text{rand}}\right)$
    $x_{\text{new}} \leftarrow \texttt{Steer}\left(x_{\text{nearest}}, x_{\text{rand}}\right)$
    if $\texttt{CollFree}\left(x_{\text{nearest}}, x_{\text{new}}\right)$ then
        $X_{\text{near}} \leftarrow \texttt{Near}\left(G = (V, E), x_{\text{new}}, \delta\right)$
        $x_{\text{min}} \leftarrow \arg\min_{x \in X_{\text{near}}} \mathrm{C}(x) + c\left(x, x_{\text{new}}\right)$
        $V \leftarrow V \cup \{x_{\text{new}}\}$
        $E \leftarrow E \cup \{(x_{\text{min}}, x_{\text{new}})\}$
        for all $x \in X_{\text{near}}$ do
            if $\mathrm{C}(x_{\text{new}}) + c\left(x_{\text{new}}, x\right) < \mathrm{C}(x)$ then
                $E \leftarrow E \setminus \{(\mathrm{P}(x), x)\}$
                $E \leftarrow E \cup \{(x_{\text{new}}, x)\}$
            end if
        end for
    end if
end for
return $G = (V, E)$
```

Functions

- $\texttt{SampleFree}()$: sample point in free space
- $\texttt{Nearest}(G, x)$: point in G nearest from x
- $\texttt{Steer}(x_1, x_2)$: point toward x_2 at a given distance from x_1
- $\texttt{CollFree}(x_1, x_2)$: no obstacle between x_1 and x_2
- $\texttt{Near}(G, x, d)$: points in G at a distance from x less than d
- $\mathrm{C}(x)$: cost between x_{init} and x, walking up the graph
- $c(x_1, x_2)$: cost between x_1 and x_2
- $\mathrm{P}(x)$: parent of x.

Planning with potential fields

Potential fields

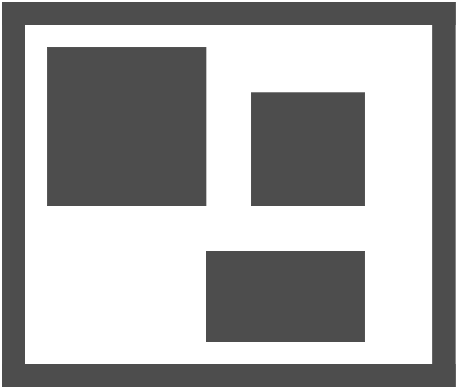
- repulsive field around obstacles
- attractive field around goal
- combination of both
- gradient descent

Result

- path
- away from obstacles
- quick to compute
- local minima

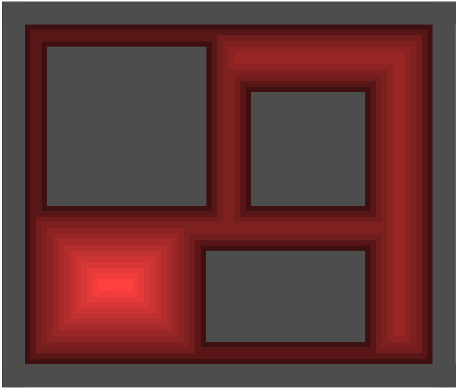
Example

Free space



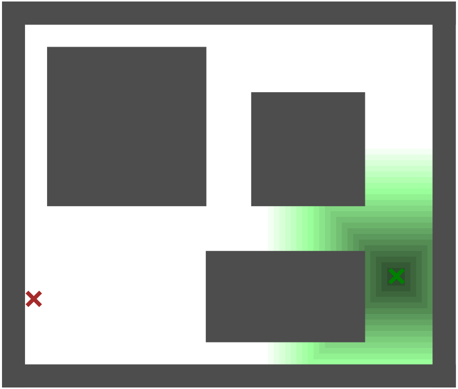
Example

Repulsive field



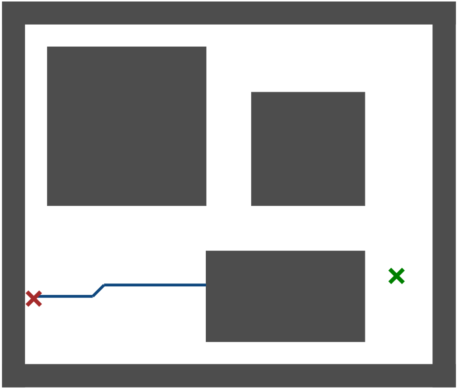
Example

Attractive field



Example

Gradient descent



Local minimum!

Planning with visibility graph

Visibility graph

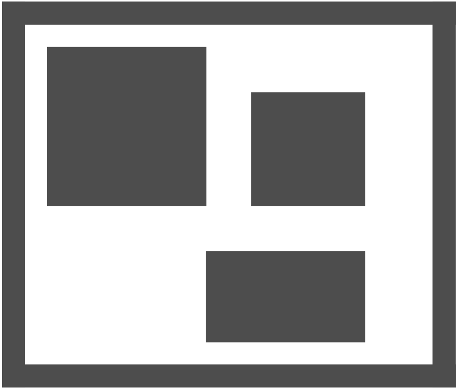
- nodes: vertices of obstacles
- edge: iff visibility between nodes
- start and goal as nodes
- graph search

Result

- path
- distance optimal
- can follow obstacle edges
- need polygonal/polyhedral obstacles

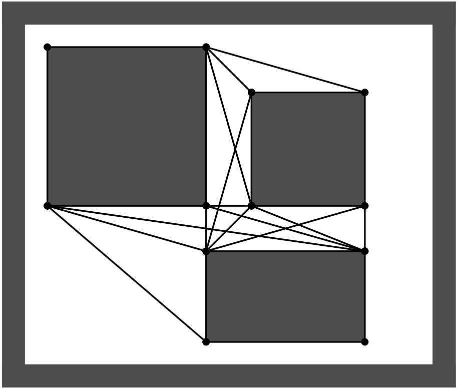
Example

Free space



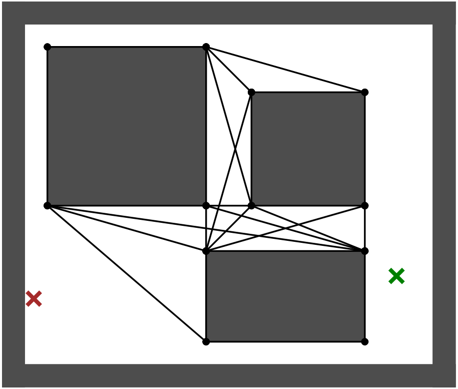
Example

Visibility graph of obstacles



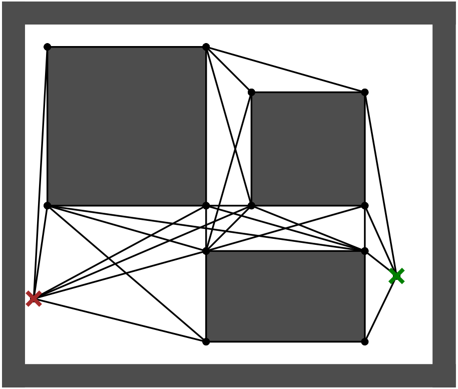
Example

Include start and goal



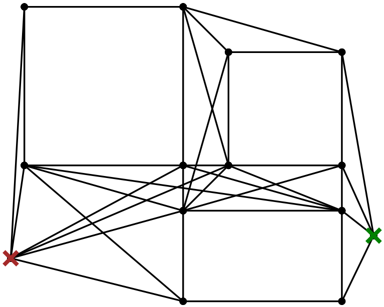
Example

Include start and goal



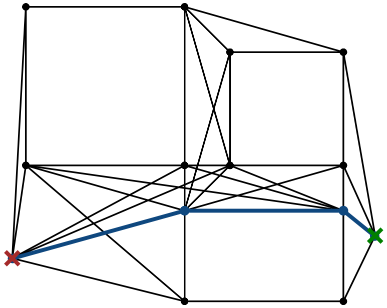
Example

Complete graph



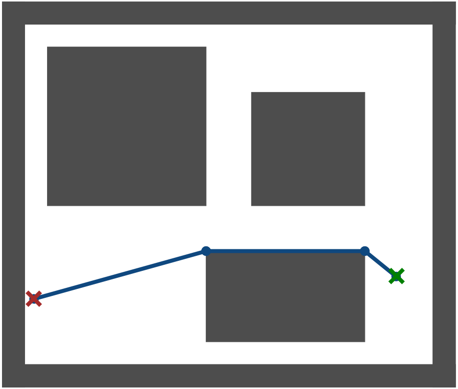
Example

Graph search (A* or other)



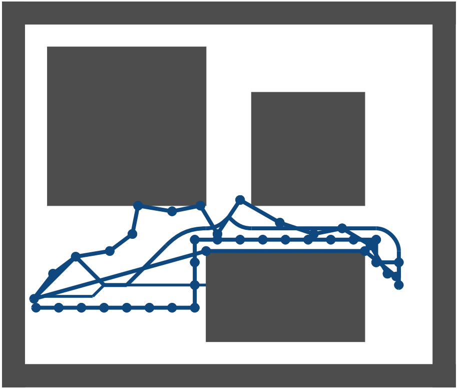
Example

Result



Comparison

Comparison of the paths



03

Conclusion

Conclusion

Configuration space

- space free from collision
- adapted to planning
- similar for mobile and articulated robots

Planning algorithms

- different families with different internal representations
- various optimization criteria

Limits

- known map
- static obstacles

Bibliography

RRT*, PRM*, etc.

- Karaman and Frazzoli, *Sampling-based algorithms for optimal motion planning*, IJRR, 2011.

Books

- Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- Lavalle, *Planning Algorithms*, Cambridge University Press, 2006.
- Siciliano *et al.*, *Springer Handbook of Robotics*, 2nd ed., Springer, 2016.

Thanks for your attention

Questions?