

CS354

Nathan Sprague

October 25, 2016

Representing Space

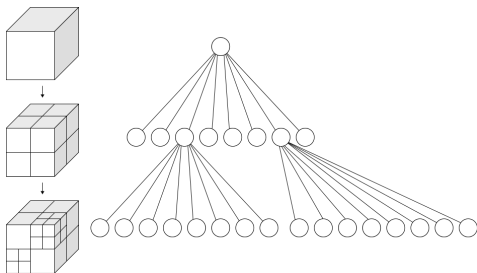
- Two issues to consider today:
 - How to efficiently and conveniently encode a map of the robot's environment?
 - How to parameterize the configuration of the robot?
- Both questions need to be addressed in order to plan.

Grid Based Maps

- Recall:
 - Easy to work with, not space efficient
 - Naive 2d grid representation of a 10m \times 10m room at 1cm accuracy:
 - $1000 \times 1000 = 1,000,000$ cells
- Quadtree is a more space efficient alternative...

Grid Based Maps

- Recall:
 - Easy to work with, not space efficient
 - Naive 2d grid representation of a 10m \times 10m room at 1cm accuracy:
 - $1000 \times 1000 = 1,000,000$ cells
- Quadtree is a more space efficient alternative...
- Octree is the 3d generalization



<http://en.wikipedia.org/wiki/File:Octree2.svg>, <http://creativecommons.org/licenses/by-sa/3.0/>

Topological Maps

Configuration Spaces

- “A configuration $\mathbf{q} \in \mathcal{C}$ of the robot \mathcal{A} is a specification of the state of \mathcal{A} with respect to a fixed frame F_w ” (Dudek and Jenkin)
- Turtlebot configuration: $\mathbf{q} = [x, y, \Theta]$.

Configuration Spaces

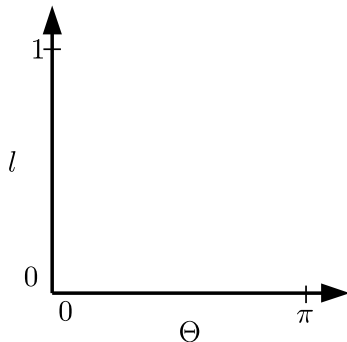
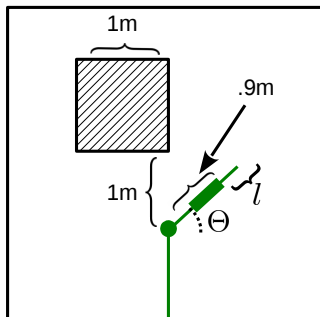
- “A configuration $\mathbf{q} \in \mathcal{C}$ of the robot \mathcal{A} is a specification of the state of \mathcal{A} with respect to a fixed frame F_w ” (Dudek and Jenkin)
- Turtlebot configuration: $\mathbf{q} = [x, y, \Theta]$.
- A **C-Obstacle** \mathcal{CB}_i is defined as:
 - $\mathcal{CB}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset\}$
 - \mathcal{B}_i is the space occupied by obstacle i .
 - $\mathcal{A}(\mathbf{q})$ is the space occupied by the robot in configuration \mathbf{q} .

Configuration Spaces

- “A configuration $\mathbf{q} \in \mathcal{C}$ of the robot \mathcal{A} is a specification of the state of \mathcal{A} with respect to a fixed frame F_w ” (Dudek and Jenkin)
- Turtlebot configuration: $\mathbf{q} = [x, y, \Theta]$.
- A **C-Obstacle** \mathcal{CB}_i is defined as:
 - $\mathcal{CB}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset\}$
 - \mathcal{B}_i is the space occupied by obstacle i .
 - $\mathcal{A}(\mathbf{q})$ is the space occupied by the robot in configuration \mathbf{q} .
- $\mathcal{C}_{free} = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap (\cup_i \mathcal{B}_i) = \emptyset\}$
- $\mathcal{C}_{obs} = \overline{\mathcal{C}_{free}}$

Exercise

Draw C_{free} for this robot:



- Robot arm with a single rotational joint and a single prismatic joint
 - l - prismatic joint extension in meters
 - Θ - angle of rotational joint ($\Theta \approx \pi/4$ in the image)

Planning

- “A free path in C-space is a continuous curve exclusively in \mathcal{C}_{free} that connects two configurations \mathbf{q}_{start} and \mathbf{q}_{goal} .”
- Paths may be
 - free: obstacles are not touched.
 - semi-free: obstacles may be touched.
- The goal of planning is to find a path.

Additional Topics

- Holonomic vs. Non-holonomic constraints
 - Holonomic constraints restrict the allowed configurations.
 - Non-Holonomic constraints put limits on the paths between configurations.
- Point robot assumption and object dilation
 - Simplifies the problem of finding \mathcal{C}_{free}