Path Planning for Point Robots

Jane Li
Assistant Professor
Mechanical Engineering Department, Robotic Engineering Program
Worcester Polytechnic Institute
Quiz (10 pts)

• (3 pts) Why human-robot handing-over is not a trivial problem?

• (3 pts) List three motion planning problems you may encounter in physical human-robot handing-over

• (4 pts) Use an example to describe how to combine motion planning with robot learning
Not a trivial problem

• Goal
  • Fluent and natural human-robot object handing-over

• Human-robot object handing-over is not a trivial problem
  • Infer human intent
  • Identify object affordance
  • Planning feasible and natural motion
  • Handle exception
Example

- Robot observing human manipulation
Example

- Communicate Intent
Example

• Identifying and tracking object
Example

• Planning reaching and grasping
Example

- Trajectory control
Project 6 – Online motion planning in dynamic virtual environment

Video Game

Motion planning/learning AI

Haptic Device
Path Planning for Point Robot
Overview

• Platform

• Course projects
Problem setup

• Robot description
  • Point robot, of time-varying position

• Environment & robot Geometry
  • 2D environment, with polygonal obstacles

• Objective
  • Find a collision-free path from start to goal
How does a bug navigate a 2D maze?

- Think as if you are a bug

For now you have a ticket for speeding so you can not fly
Bug 1 Algorithm

\[ q_{\text{start}} \rightarrow q_1^H \rightarrow WO_1 \rightarrow q_2^H \rightarrow WO_2 \rightarrow q_2^L \rightarrow q_{\text{goal}} \]
Bug 2 Algorithm

start

good

L1
H1

L2
H2

goal
Bug 1 VS Bug 2

• Which is better?

• How to measure the algorithm efficiency?
Comparison of Bug 1 and 2 algorithms
Navigation with more global information
Framework of a 2D navigation problem

Continuous representation

Discrete representation

Graph searching
Continuous representation
Framework

Continuous representation

Discrete representation

- Sampling for collision-free point
- Processing geometric features

Graph searching
• If a collision-free path exists
  • There must be a piecewise linear path that bends only at the obstacles vertices
Visibility Graph

- **Nodes**
  - $q_{\text{init}}$, $q_{\text{goal}}$, obstacle vertices

- **Edges**
  - Obstacle edges
  - No intersection with obstacles
Naïve Algorithm for Computing Visibility Graph

Input: $q_{init}, q_{goal}, \text{ polygonal obstacles}$
Output: visibility graph $G$

1: for every pair of nodes $u,v$
2: if segment($u,v$) is an obstacle edge then
3: insert edge($u,v$) into $G$;
4: else
5: for every obstacle edge $e$
6: if segment($u,v$) intersects $e$
7: go to (1);
8: insert edge($u,v$) into $G$. 
1: for every pair of nodes u,v \[O(n^2)\]
2: if segment(u,v) is an obstacle edge then
3: insert edge(u,v) into G; \[O(n)\]
4: else
5: for every obstacle edge e \[O(n)\]
6: if segment(u,v) intersects e
7: go to (1);
8: insert edge(u,v) into G.

• Running time? \[O(n^3)\]

• More efficient algorithm?
  • Sweep-line algorithm – \(O(n^2 \log n)\) (see Principles 5.1.2)
Reduced Visibility Graph

- Construct visibility graph from
  - Supporting and separating lines
Continuous representation → Discrete representation → Graph searching

Search algorithm?
Breadth-first search
Breadth-first search
Breadth-first search
Breadth-first search
Breadth-first search
Breadth-first search
**Breadth-first search**

**Input:** \( q_{\text{init}}, q_{\text{goal}}, \) visibility graph \( G \)

**Output:** a path between \( q_{\text{init}} \) and \( q_{\text{goal}} \)

1: \( Q = \textbf{new} \) queue;
2: \( Q.\text{enqueue}(q_{\text{init}}); \)
3: \textbf{mark} \( q_{\text{init}} \) as visited;
4: \textbf{while} \( Q \) is not empty
5: \quad \textbf{curr} = \( Q.\text{dequeue}() \);
6: \quad \textbf{if} \ \textbf{curr} == \( q_{\text{goal}} \) \textbf{then}
7: \quad \quad \textbf{return} \ \textbf{curr};
8: \quad \textbf{for} each \( w \) adjacent to \( \textbf{curr} \)
9: \quad \quad \textbf{if} \ \textbf{w} \ \textbf{is} \ \textbf{not} \ \textbf{visited}
10: \quad \quad \quad \textbf{w.parent} = \textbf{curr};
11: \quad \quad \textbf{Q.enqueue}(\textbf{w})
12: \quad \quad \textbf{mark \ w \ as \ visited}
Other graph search algorithms

• Depth-first
  • Explore newly-discovered nodes first
  • Guaranteed to generate shortest path in the graph? No

• Dijkstra’s Search
  • Find shortest paths to the goal node in the graph from the start

• A*
  • Heuristically-guided search
  • Guaranteed to find shortest path
Recap

Continuous representation

Discrete representation

Graph searching
Recap

• Running time
  • Compute the visibility graph – Naïve method – $O(n^3)$
    • An optimal $O(n^2)$ time algorithm exists.

• Space?
  • Store graph as adjacency list or adjacency matrix
    $O(n^2)$
Application to Shakey the Robot (1969)
Navigation using visibility map
Other ways for building roadmaps?

- **Voronoi graph**
  - Introduced by *computational geometry* researchers.
  - Generate paths that *maximizes clearance.*
Voronoi map
Cell decomposition

- Exact methods
  - 2D - Trapezoids, triangles, etc.
  - Adjacency map

- Approximate methods
  - Decompose space into cells usually have **simple, regular** shapes
  - Facilitate **hierarchical** space decomposition
Cell decomposition – Exact method
Cell decomposition – Approximate method

Quad-tree

empty, mixed, full
Hierarchical Decomposition

- **Strategy**
  - (1) **Decompose** the free space into cells
  - (2) **Search** for a sequence of mixed or empty cells that connect the initial and goal positions
  - (3) **Further decompose** the mixed
  - (4) Repeat (2) and (3) until a sequence of empty cells is found
Octree

- EMPTY cell
- MIXED cell
- FULL cell
Assignment 2 – Due on Wed (Jan 24)

• Reading
  • Principles 5.1.2: Sweep-line algorithm
  • Individual literature review
  • Select 4 best work and 1 for presentation
Optional Assignment

- **Optional**
  - Accept volunteer to give a 5 min- talk on next Wed (Jan 24)
  - Topic – a cool application of motion planning
  - Send your presentation slides (notes must be included) by Monday (Jan 22) at noon
  - TA will email the selected presenter by 10 am of Tuesday (Jan 23)
  - Extra credit – use to replace one of your low-score quiz in the future
Next Friday (Jan 26) – TRINA workshop

- **Objectives**
  - Introduction to TRINA system (Hardware + software)
  - Help you setup workstation

- **Project group will be announcement next Wednesday (Jan 24)**

- **Requirement**
  - Each group need to bring at least one workstation with
  - Ubuntu 16 and ROS kinetic installed
  - Prefer to be dual-boot. VM can be SUPER slow for what we need
End