2

RBE-550 Motion Planning
Daniel Montrallo Flickinger, PhD

Flatland

DUE: 2023-02-06 @ 12:00 UTC

Abstract

Take a journey to the 2^{nd} dimension with these awesome gridworld point robots. Set up obstacle courses of doom, and implement various search algorithms to guide a little point robot. Does it have what it takes to survive the onslaught?

1 Introduction

As in introduction to motion planning techniques, utilize a simulation to explore discrete planning of a point robot (Figure 1) in a two dimensional grid. This point robot starts in the Northwest corner of the world, and must travel to the Southeast corner. Stationary obstacles are placed randomly according to a preset density, which the robot must avoid. Create a simulation world, visualization, and planner implementations.

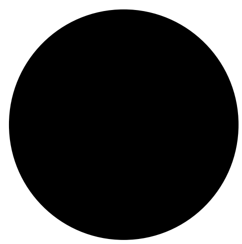

Figure 1: A point robot (actual size)

2 Software

The most straightforward implementation is to write a 2D grid simulation from scratch using Python. There exist 2D grid systems for point robots in various libraries and software systems, but many require significant work to interface with the requirements of this assignment.

However, there are options such as the Grid World model in the MATLAB Reinforcement Learning Toolbox. Another option builds upon the ROS installation created in the introductory assignment. Install the Grid Map library Fankhauser and Hutter [2016] in ROS, following the installation instructions. Costmap 2D could also work, with some modifications. Otherwise, implementations in any programming language may be submitted for this assignment.

3 Environment

The grid world must have the following attributes:

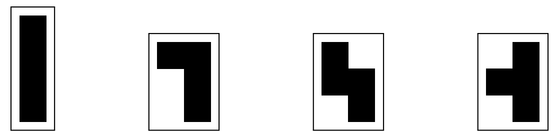

Figure 2: Example free tetromino obstacle shapes.

- 128 x 128 two dimensional grid
- point robot, occupying a single cell at each time step
- robot moves to a single neighboring cell at each step (but include portals or teleportation if you want to be weird)
- multiple randomly placed obstacles
- flat plane with hard borders (not a toroidal topology, unless you cheat)
- wumpus not included (or is it?)

4 Implementation

Instantiate a grid world with a configurable density of obstacles. Implement three forward search planning algorithms: depth first search, breadth first search, and Dijkstra's, as discussed in Section 2.2.2 of LaValle [2006]. Additionally, create a random planner that simply moves to a random neighboring cell at each iteration.

TIPS: In creating obstacles, create shapes (see Figure 2) instead of filling random individual cells. And in the random planner implementation, consider a maximum number of iterations, since it might take too long to find the goal cell.

5 Results

Produce a plot of the four implemented methods superimposed, showing the number of iterations to reach the goal versus obstacle density (varied at an arbitrary step size from 0% to 75%).

Generate three select figures depicting the world, including obstacle placement, and resulting path, for three different arbitrary obstacle densities.

Submit all source code, organized into functions, classes, associated files, etc. Also submit a brief report with a least three figures depicting example calculated paths, and the performance plots.

6 Grading and Submission

This assignment is due 2023-02-06 @ 12:00 UTC. *Late submissions are not accepted.* Upload completed assignment components (as individual files, not a single ZIP or TAR file) to the course site on Canvas.

Weight	Type	Component
25%	tar.gz	full source code
25%	PDF	report
25%	PDF	planner path figures (3)
25%	PDF	planner performance plots

7 References

Péter Fankhauser and Marco Hutter. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. In Anis Koubaa, editor, *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, chapter 5. Springer, 2016. ISBN 978-3-319-26052-5. doi: 10.1007/978-3-319-26054-9{_}5. URL `http://www.springer.com/de/book/9783319260525`.

Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, May 2006. ISBN 9780521862059. URL `http://lavalle.pl/planning/`.

8 List of URLs

Last update: January 3, 2023