CSE 276A - Homework 4 - Report

Path Planning

Wang, Shou-Yu[A69030868][a], Huang, Yao-Ting[A69030676][a]

[a] *University of California, San Diego.*

November, 2024

Abstract

In this assignment, we decided our own world representations and also implemented two path planner approaches(maximize safety and minimize distance) with different algorithms(Brushfire and Breadth-First Search) in the same environment setup to evaluate the paths.

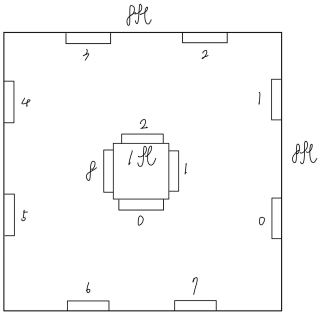


Figure 1: Environment setup.

April Tags (ID)	Position (m, m)
0(Outside)	(1.8, 0)
1(Outside)	(1.6, 0.8)
2(Outside)	(0.8, 1.4)
3	(-0.2, 1.4)
4	(-1, 0.8)
5	(-1, 0)
6	(0, -0.6)
7	(0.8, -0.6)
0(Inside)	(0.3, 0.3)
1(Inside)	(0.45, 0.45)
2(Inside)	(0.3, 0.6)
8	(0.15, 0.45)

Table 1: Ground truth of landmarks' positions.

1. Result

Check out the video of our car's navigation at link. link.

2. Environment setup

2.1. Landmarks

First we placed 8 April tags with different ids as the landmark on the wall of a 8ft × 8ft area, then we placed 4 on the four sides of middle 1ft × 1ft obstacle, shown in Fig.1, and we used those landmarks to estimate and update the robot's pose. The measured positions of the landmarks shown in Table 1.

3. Path planning

In this assignment, we implement two approaches (minimum distance and maximum safety) to generate path. For the world representation, we used configuration grid to represent the free space(world) and reduce the robot to a point, shown in Fig.4. We chose this representation for two benefits. The first is that we can easily decided the resolution of the path with varying step size. The second is the convenience of visualization and implementation.

3.1. Minimum distance

To plan a minimum distance path, we used visibility graph to generate the path. More specifically, we run Breadth-First Search(BFS) algorithm in the configuration space to find the shortest path in the collision free region. The planned paths shown in Fig.3(a)

3.2. Maximum safety

To plan a maximum safety path, we planed our path on the Voronoi graph. To elaborate, we used Brushfire algorithm to get the distance grid shown in Fig.2 that stored all the distances to the nearest obstacle. After that, we also run a BFS algorithm to find the shortest path in the Voronoi graph. The planned paths shown in Fig.3(b)

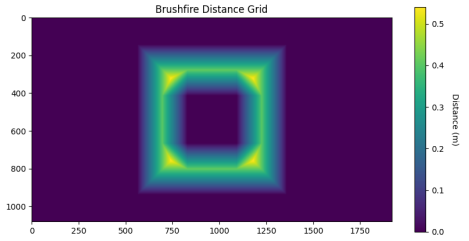
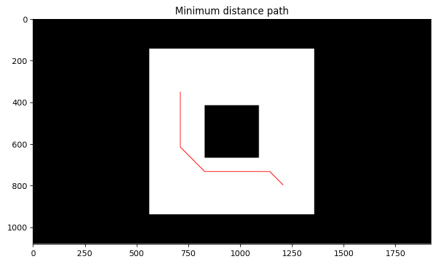
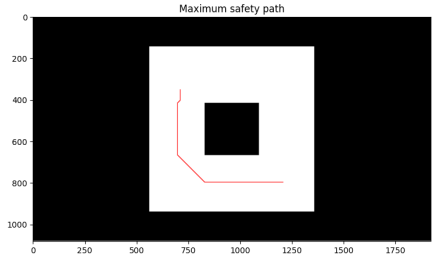


Figure 2: The distance grid get by the Brushfire algorithm.



(a) The minimum distance path.



(b) The maximum safety path

Figure 3: This figure shows the paths planned by two different approaches.

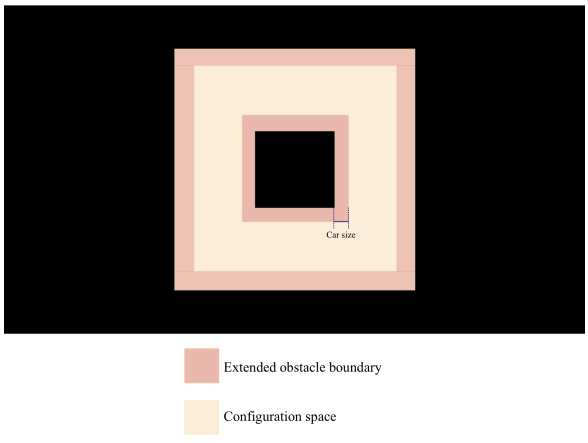


Figure 4: World representation and configuration space.

4. Discussion

4.1. Result of the map

We can see that under minimum distance policy, the path get close to the obstacle at the corner.

4.2. Algorithm decision

For the maximum safety approach, we tried two different path planner algorithm, which are Voronoi graph based method and potential field based method. However, due to the shape of the environment, there will be local minimum at the middle of each corner, and thus might cause car stuck in the corner. Consequently, we chose Voronoi graph based method to make sure the robustness of finding a path.