MOBILE ROBOT & NAVIGATION

ROBOT EXPLORATION ON UNKNOWN ENVIRONMENT

Zhaoqi Li

Chengli Liu

Yang Wang

May 2021

Barcelona

CONTENT

1.INTRODUCTION

Exploration involving mapping and simultaneous localization in an unknown environment is an important task in mobile robotics. A mobile robot exploring and mapping an unknown environment needs to know its own location, while it requires a map information in order to determine the location. To tackle this problem, an exploration strategy should be performed. The central question in exploration is: Given the current information about the world, where should the robot move to gain as much new information as possible? A well-known strategy is a Frontier-based exploration. The central idea behind the exploration is: to gain the newest information about the world and move to the boundary between open space and unknown territory. The boundary is referred to as a frontier. Commonly used frontier strategies are the biggest and closest frontiers strategy. As it can be seen from the names, in these two strategies, the robot is commanded to move towards the frontier with the biggest size or to the closest frontier to explore the unknown environment. Although these strategies are advantageous because of their small computational time, it has one important drawback. The solution to the exploration problem is often far from the optimal solution. It is a greedy algorithm since it solely moves to the frontier that provides the maximum information in one step ahead. This algorithm is often stuck in a local minimum solution. In this project, we applied two improved strategies based on the closest frontier method and add some beneficial adjustments to the given strategies to achieve a better exploration performance. Two new strategies are considered in our project, namely the FRONTIER SCORE strategy and the FRONTIER DENSITY strategy. In the following sections, each exploration strategy is explained. In addition, apart from the base exploration methods mentioned above, we modify the goal position selection policy to shorten the exploration time and therefore enhance the exploration performance.

2. GOAL SELECTION POLICY

- <u>FRONTIER SCORE</u>

The robot evaluates all the frontiers on the map by the following criterion. It considers the distance of the frontier to the robot and the size of the frontier. Using two tune parameters Ω and Φ, we can modify the weighting of the distance and the size of a frontier on the score, thus, depending on the different environment characteristics, the robot can be designed to have a different priority on the features of the frontiers and performs a different exploration behavior. The frontier evaluation policy is as follow:

$$S = \Omega \frac{1}{d_i} + \Phi c_i, i \epsilon \{1, 2, \dots, n\}$$

where Ω , Φ are tune parameters that balance the importance of the distance of the frontier to the robot and the size of the frontier. A higher value of Ω indicates that the weighting of the distance of the frontier to the robot on the score is higher, and the robot has higher priority to move to a closer frontier instead of a big but rather far frontier. In our case, Ω is 5 and Φ is 1. The complete exploration algorithm is illustrated in the following flow chart.

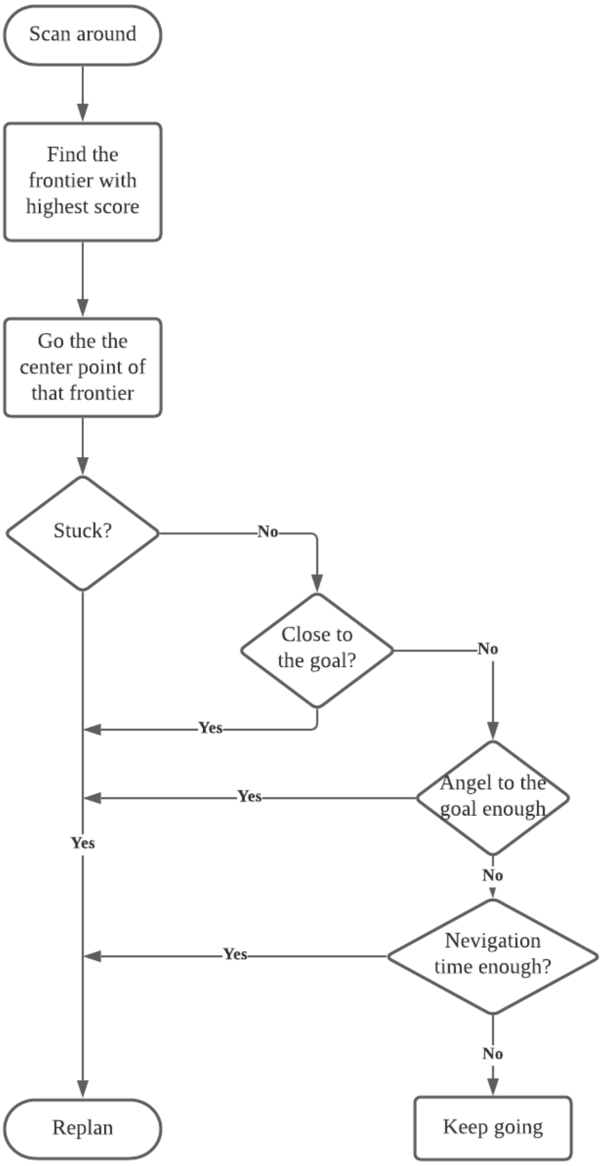

Fig. 1 Flow chart of Frontier score

- <u>FRONTIER DENSITY</u>

In this algorithm, we devote to let our robot take as few repetitive paths as possible, which means, the robot needs to explore completely the unknown area near it to make sure it will never get back to the same area from a territory far away.

The ideal situation is the robot explores a territory of the environment only once.

To realize this objective, we firstly transfer the global coordinate to the coordinate of robot and divide the whole environment into 4 quadrants.

Define the number of frontiers in each quadrant as frontier density for each quadrant. We would like our robot to go to the quadrant with lowest frontier density, because this means in that quadrant, most territory have been explored.

After detecting the low-density quadrant, the robot will choose the center point of that frontier, define it as the next goal point.

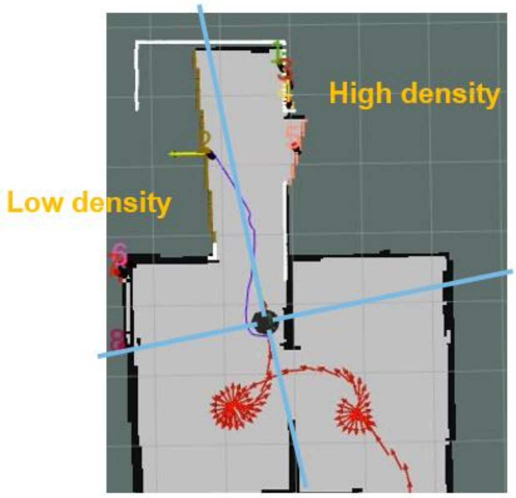

Fig. 2 Low-density and High-density quadrant

There are some specific cases we need to notice:

1. We need to exclude the quadrant that has 0 frontier inside.
2. If there are more than one quadrant have the same density, we need to mix these quadrants, then find the closest frontier in the mixed quadrant.

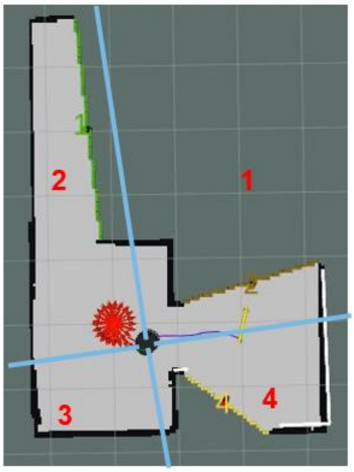

Fig. 3 Quadrants with same density

Like the condition shows in Fig. 3, quadrant 1, 2 and 4 all have 1 frontier, so that we should consider quadrant 1, 2 and 4 as 1 mixed quadrant. The complete exploration strategy is shown in the following:

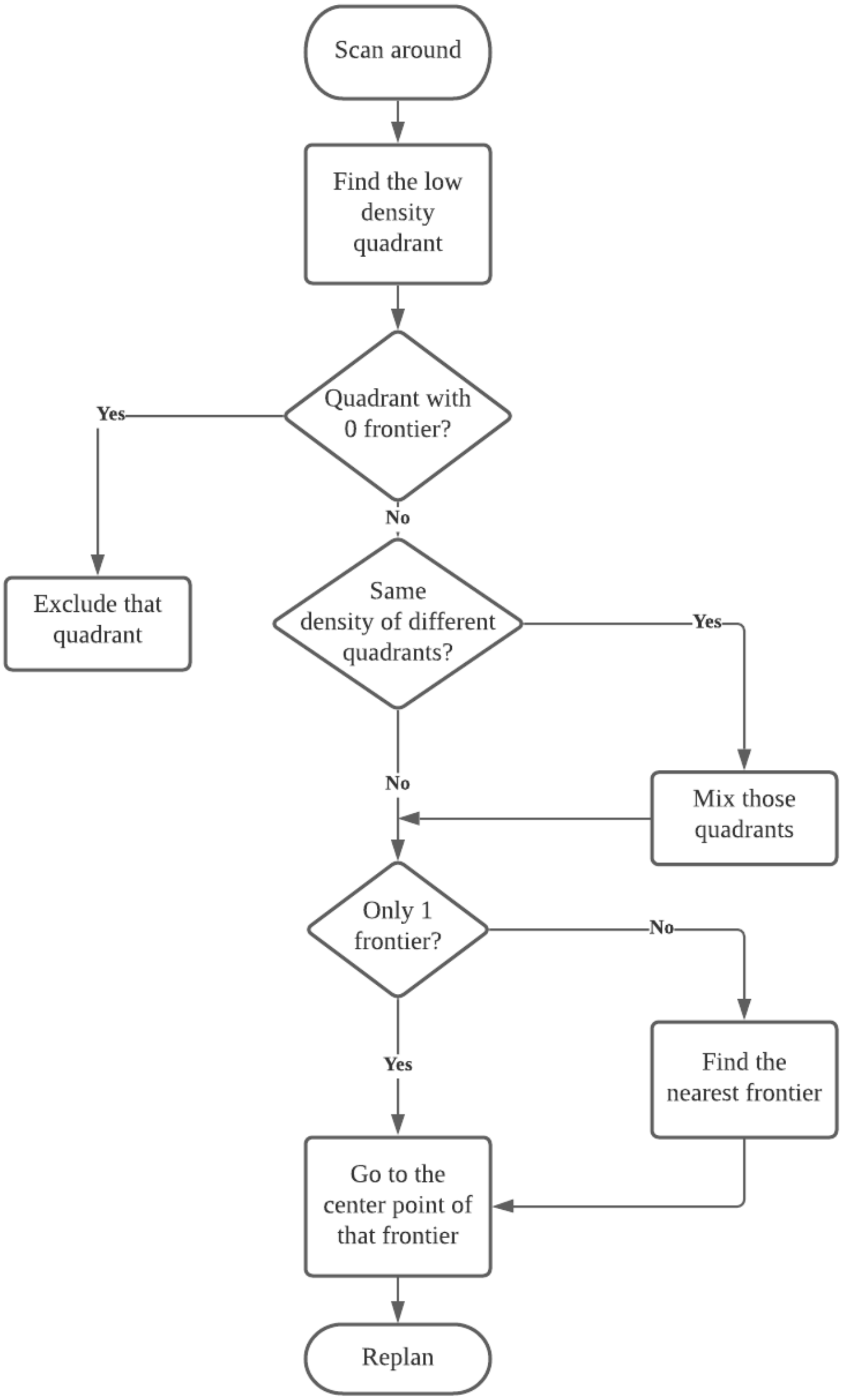

Fig. 4 Flow chart of Frontier density

- <u>GOAL POSITION ADJUSTMENT</u>

In the first two approaches, we choose the goal point on the center of goal frontier. However, this might not be necessary.

The closer the frontier is to the wall, the easier the center point of the frontier is unreachable. In this case, we consider moving the goal point from frontier to a point close to the frontier but not reaching.

That is why we use the point called center of gravity of 3 frontiers.

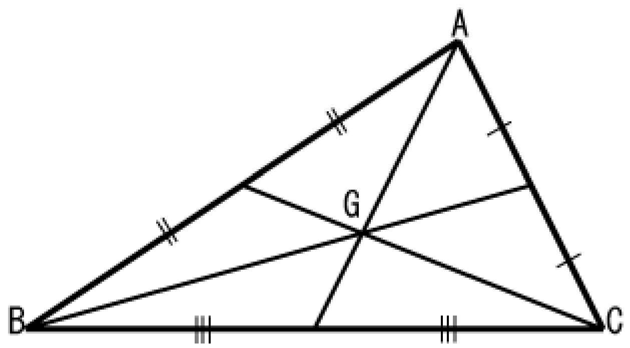

Fig. 5 Center of gravity

This method is based on the previous two algorithms.

We firstly choose the frontier with highest score and other two second and third highest score frontier near it. Then we use the center point of these 3 frontiers to make a triangle, and use the center of gravity point of this triangle as robot’s next goal point, just as the Fig.6 shows.

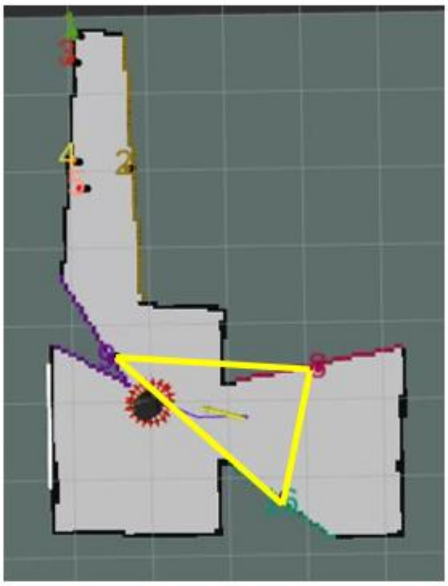

Fig. 6 Center of gravity of 3 highest score frontiers

By this adjustment, we can prevent the robot from getting stuck to a certain extent.

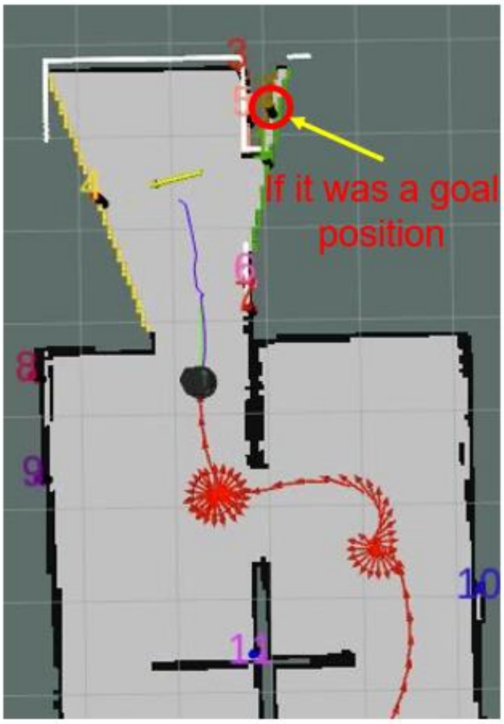

Fig. 7 Condition that the robot might get stuck.

3.REPLAN

There are 4 conditions that we consider the robot need to implement replan:

1. Distance of robot to the goal position is smaller than 0.5 m.
2. Angel difference between robot’s orientation and the goal orientation is 20 deg.
3. Navigation duration is more than 1 second.
4. Abnormal status (like stuck)

4.RESULTS

- <u>MAP 1</u>

MAP 1	Travel distance (meters)	Travel time (minutes)	Explored area (m^2)	Stuck time	Average travel time	Average travel area
Frontier Density + Gravity point	31.00	02:48	44.14	2 Times	02:29	43.98
	25.17	02:20	44.18			
	27.63	02:18	43.95			
	29.15	02:37	44.31			
	33.29	02:51	44.23			
	23.81	02:06	43.66			

	24.63	02:19	43.62			
	25.22	02:33	43.78			
Frontier Score + Gravity point	24.13	02:07	43.36	1 Time	02:10	43.61
	21.43	02:00	43.47			
	21.19	02:14	43.49			
	26.57	02:33	43.62			
	20.27	01:57	43.83			
	25.14	02:22	43.75			
	22.48	01:58	43.71			
	26.72	02:14	43.67			

- <u>MAP 2</u>

MAP 2	Travel distance (meters)	Travel time (minutes)	Explored area (m^2)	Stuck time	Average travel time	Average travel area
Frontier Density + Gravity point	24.69	02:29	22.19	3 Times	02:22	22.09
	22.10	02:09	21.84			
	21.87	02:12	22.00			
	24.17	02:18	21.96			
	31.82	02:12	22.12			
	28.49	02:52	22.43			
Frontier Score + Gravity point	24.78	02:35	21.95	2 Time	02:07	22.00
	22.26	02:06	22.15			
	21.01	01:56	21.74			
	20.84	01:55	21.87			
	23.54	02:23	22.14			
	19.80	01:49	22.17			

- <u>MAP 3</u>

MAP 2	Travel distance (meters)	Travel time (minutes)	Explored area (m^2)	Stuck time	Average travel time	Average travel area
Frontier Density + Gravity point	134.33	11:34	141.36	2 Times	11:34	141.36
Frontier Score + Gravity point	117.31	10:00	141.68	1 Times	10:00	141.68

5.CONCLUSION

Finally, we found that the Frontier Score method performs better in explored time and stuck time. In explored area, both Frontier Score and Frontier Density have similar performance.

So that we can say, Frontier Score method is the most efficient method in our project.

Meanwhile, we notice that a goal position adjustment can additionally improve the performance of robot exploration.

6.REFERENCES

[1] A. Mobarhani, S. Nazari, A. H. Tamjidi and H. D. Taghirad, "Histogram based frontier exploration," 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 1128-1133, doi: 10.1109/IROS.2011.6095018

[2] B. Yamauchi, A. Schultz and W. Adams, "Mobile robot exploration and map-building with continuous localization," Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), 1998, pp. 3715-3720 vol.4, doi: 10.1109/ROBOT.1998.681416.

[3] Mobile robot exploration by using environmental boundary information, W Theeraphol, O Taiki, H Tatsunori, O Jun - International Scholarly Research Notices, 2013

[4] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky and H. F. DurrantWhyte, "Information based adaptive robotic exploration," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 540-545 vol.1, doi: 10.1109/IRDS.2002.1041446

[5] 1. Kollar T, Roy N. Trajectory Optimization using Reinforcement Learning for Map Exploration. The International Journal of Robotics Research. 2008;27(2):175-196. doi:10.1177/0278364907087426