STUDY ON DIFFERENT AUTONOMOUS EXPLORATION ALGORITHMS

ARIMA VU RAM THAYUMANAVAR
NUID: **001643148**

INTRODUCTION

For a past few decades, the Robotics field has advanced itself by solving a variety of research problems. In this project, I am mainly focusing to a specific problem of autonomous exploration of a mobile robot in an unknown environment. Specifically, I will be going through some of the state-of-the-art algorithms implemented for autonomous exploration of mobile robots.

And, for this specific task of analyzing the efficiency and optimality of the algorithm(completeness), I will be using a TurtleBot simulator available in ROS package. And for achieving localization and mapping I will be implementing gMapping package to present the TurtleBot traversal in a Rviz UI part of the ROS package. Further, for the purpose of analyzing the efficiency and optimality of the algorithms, I have designed various World maps to test the TurtleBot in each of the unknown environment. Also, I performed multiple trials for each of the Map with different origin points for the TurtleBot in the designed World map for the same algorithm. Based on the understanding and results of various trials of each algorithm, I have built a list of parameters that could possibly affect the efficiency and completeness of the exploration algorithm. Using the understood parameters, I have designed a more efficient autonomous exploration algorithm and again performed studies to verify the optimality of the algorithm.

Hence, from a computational perspective the inputs for the program would be various permutations of World map to test the TurtleBot in various environment. And output is the efficiency of the designed algorithm based on the time taken for the TurtleBot to completely explore the unknown environment under different trials on various maps.

AUTONOMOUS EXPLORATION

Given a map a mobile Robot can quickly localize itself using popular SLAM algorithms and navigate in the known environment. But when the mobile robot does not know the map of its current world then it finds difficulty in understanding and exploring the map. Hence, we come up with a set of algorithm to guide the mobile robot to navigate around the unknown environment by localizing and building a map. Once, a map of the given world has been explored the mobile robot can always use this map in future to reference and perform additional tasks. Further, autonomous exploration has a huge scope in space missions, where the mobile robot is expected to understand a unknown environment without human guidance.

Hence, its important to have an efficient autonomous exploration algorithm to navigate around the space and build a map that potentially can be used for several other applications. Importantly, the algorithm should be optimal to always completely explore a given unknown environment.

Thus an exploration algorithm is mainly about picking the next point in the map for the Robot to traverse and explore the map. For the purpose of this study, of the various different algorithms, I will be focusing my study specifically to two broad basic state-of-the-art algorithms.

1. **Probabilistic Random Walk Algorithm**
2. **Frontier based algorithm**

FRONTIER:

Frontiers are basically the regions on the boundary between the open space (already explored) and unexplored space.

Probabilistic Random walk algorithm:

As it is very self explanatory from the title, the probabilistic random walk algorithm is a method of choosing random points in the open space (explored space) as the next point in the World for the mobile robot to traverse. And once, the mobile robot reaches the next selected point, the mobile robot goes about exploring the space around it to build map and localize and then again goes about picking up a random point in the open space.

Since, there is no constraint involved in picking a next point in the map, this algorithm can be assumed to be optimal (complete) even though it might take a while for completion of exploration in certain cases. And also, the efficiency of this algorithm is unpredictable as it randomly moves around the space, and hence the time taken for exploration varies massively on each trial.

Frontier based Exploration algorithm:

Frontier based exploration technique is a variety of techniques that involves picking the next point from the frontier section of the map. Hence, various different implementation of this algorithm guides the Robot to pick a specific frontier based on different constraints.

Of the different implementations, the state-of-the-art technique used for frontier based exploration is the Centroid based frontier exploration. In the Centroid based frontier exploration, at a given point after exploration the mobile robot analyses various frontiers as a cluster of frontiers and picks one of the frontiers that has more area covered by the cluster and evaluate its centroid. This calculated centroid is chosen as the next point for the mobile robot to traverse in the world map.

From a intuitive perspective, the frontier based exploration also seems to be optimal as it goes on executing until all such frontiers detected are covered by the mobile robot. Unlike the probabilistic random walk algorithm, the frontier based algorithm seems to provide a more efficient algorithm that also has definitive variance in the time for exploring given unexplored space at multiple trials.

OBSERVATION:

On running the above mentioned algorithms on various different here are some few observations:

- As expected the Frontier based approach performed better than the Probabilistic random walk algorithm.
- However, the Random exploration algorithm quickly explored about the starting 40% of the map compared to the Frontier based algorithm by tracking the exploration rate.
- Also, it was observed that the frontier based algorithm provided a more optimal solution compared to the probabilistic random walk autonomous exploration.

My Approach:

From the above observation its obvious that the frontier based technique provides a better solution both in efficiency and optimality perspective of the computational problem. However, it was also noticed that the probabilistic random walk algorithm in most of the cases explored about 40% of the map before the frontier based exploration technique. Hence, using a trial and error approach by considering different parameters, I came up with the following algorithm:

Idea is to implement both probabilistic random walk and Centroid based frontier exploration. Again, in the probabilistic random walk technique I added up a constraint to the algorithm to search and specifically pick only random points from the frontier instead of picking random points from the whole of the open unexplored space.

Parameters:

1. Fitting of the open space into a rectangle and holding the length of the rectangle inside threshold variable.
2. Calculate the number of already visited cells in the occupancy grid map of the present world and store it in a visitedPrev variable.
3. Calculate the distance of the current position of the mobile robot and the sampled/selected destination point.
4. Condition for bad exploration, when there is a significant drop in the number of newly visited nodes below a visited_threshold set.

Hence, the algorithm keeps switching between the frontier based probabilistic random walk and the centroid based frontier exploration based on the parameters mentioned above.

Step 1: Start propagating the unknown environment using the Frontier based random walk technique.

Step 2: Keep evaluating the distance traveled and the compare it to the threshold variable.

Step 3: When the distance overshoots the threshold variable, the algorithm switched to the Frontier based exploration technique to sample the next destination point.

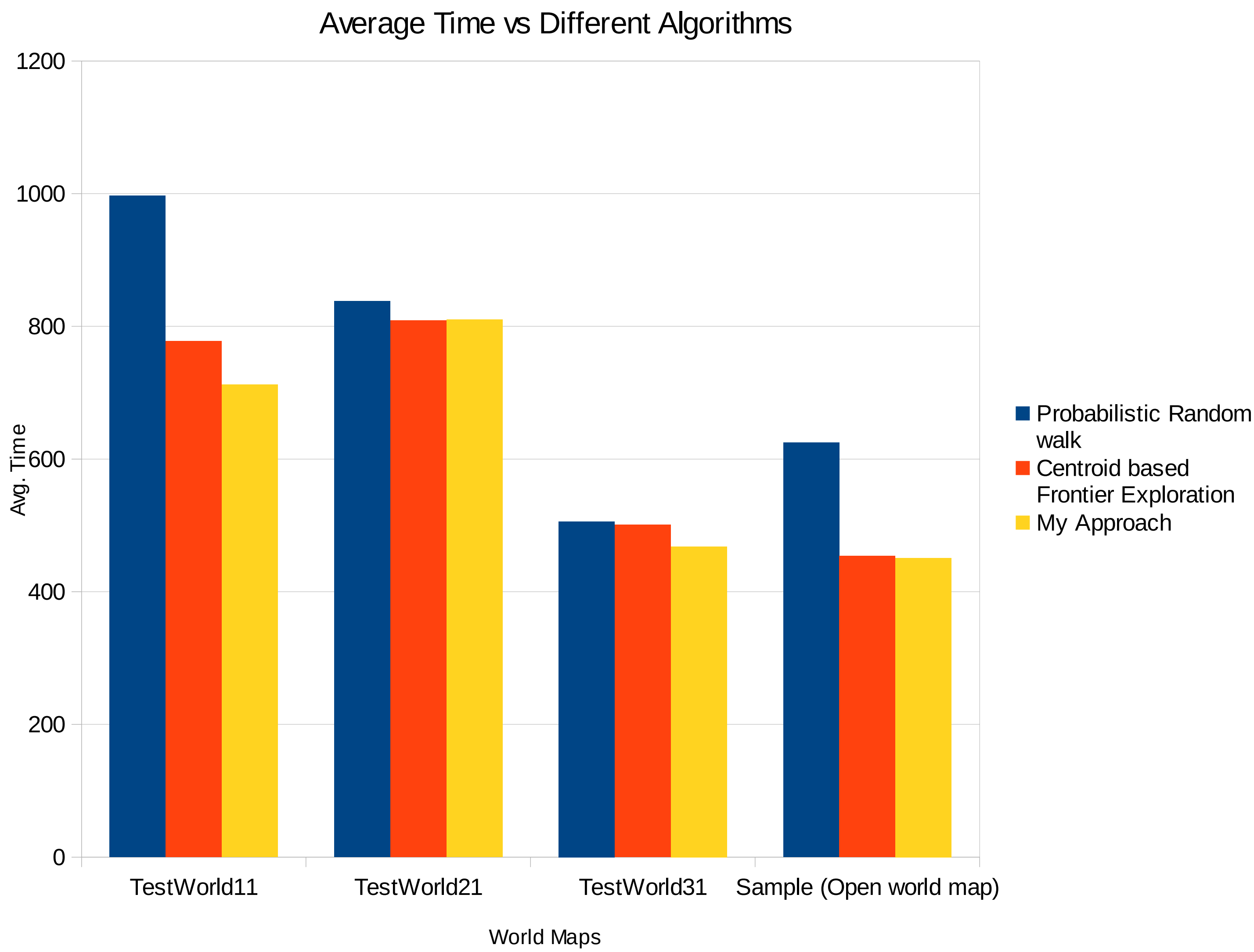
Step 4: At the same time keep evaluating the number of newly visited nodes and compare it to the visited_threshold variable.

Step 5: When the number of newly visited nodes goes below the threshold value, the algorithm switched back to the Probabilistic random walk technique.

Step 6: Step 1 through Step 6 keeps repeating until all the frontier cells in the occupancy grid map has been explored.

RESULTS:

I have performed the autonomous exploration based on the above mentioned three algorithms, using a mapping a script that effectively shows the map generation by the TurtleBot in the user interface. I further, use this UI to understand and calculate the time taken for the TurtleBot to explore the unknown environment and also evaluated the rate of exploration in terms of number of cells being explored at each second.



These are the average time results calculate for three different algorithms on four different maps run over the 5 times each with different TurtleBot start positions on the same map. On a average the suggested algorithm seems to show some improvement. But in case of TestWorld21 the Centroid based frontier exploration technique seemed to outperform my technique by 10-20 seconds in average time. Even though, on testing for four different maps and several other maps, the algorithm does not seem to always provide or come up with the best possible solution even on a average time case. This was mainly because of the threshold values that were manually set for the algorithm.

When I modified the threshold, there was an improvement in the case of TestWorld21 where My approach equals the Centroid based exploration algorithm on average time for autonomous exploration.

Thus, it was understood that the algorithm requires an implementation of learning system that scores and penalizes the system for every good and bad choice of the threshold value. Hence, I believe implementing an Reinforcement learning technique in future could add more clarity to the algorithm to choose between one of the two exploration algorithms.

FUTURE WORK:

Even though, the algorithms seem to be comprehensive and efficient, the inconsistencies produced from the laser scan messages causes a lot of disturbances in generation of clear map. Sometimes, some frontiers get missed out in certain trials thus ending the loop well before the exploring the complete map. In certain cases, it was also observed that due to inaccurate localization the gMapping suite builds a map with multiple overlap layers which in turn affects certain paths in the map to be blocked. Also it was found out that when theres a thin line space in between the walls and obstacles for the Turtlebot to move, due to inaccuracies in some laser scans, no path could be displayed thus causing incomplete map and latencies in certain cases.

Hence, I believe a better information theory model is required to overcome such inaccuracies generated by laser scan messages.

Also, with respect to the proposed algorithm, the results were not always consistent and requires some reinforcement learning techniques that learns and decides how to set the threshold values which is the key to switching between different algorithms for improving the efficiency and optimality of the autonomous exploration algorithm.