PROJECT 5

Planning for Autonomous Robots

May 5, 2022

Instructors:
R. Monfaredi

Semester:
Spring 2022

Course code:
ENPM661

Students:
Jerry Pittman, Jr. (117707120)

Maitreya Kulkarni (117506075)

Contents

List of Figures

1 Introduction

1.1 Background

For a robot to navigate efficiently in any given environment, there has to be a predefined map loaded in the software stack of the robot. The general method of mapping the environment is to manually run the robot or a vehicle that has Lidar attached to it, to generate point cloud data, and later the map. There have been new autonomous map / environment exploration strategies, which take the robot into unexplored spaces. These methods van be useful to generate the map of unknown environment autonomously without the operator being present. These autonomous exploration methods work by detecting frontiers, which are the boundaries dividing explored and unexplored spaces. In the past, frontier detection used image processing tools such as edge detection which limited the use of these methods to 2-D spaces and their exploration. We have implemented Rapidly-exploring Random Tree (RRT) algorithm as it is biased towards unexplored regions in the map. Upon going through literature on RRT, it was deducted that it can be used as a general approach to extend the exploration of maps with higher dimensional spaces and also can be implemented for multi-agent systems. The proposed strategy is implemented using ROS ad Gazebo framework.

In addition, this implementation uses local and global trees for detection of frontier points that enhances and makes the exploration of unknown spaces more efficient.

1.2 Goal Of the Project

The main goal of this project is to autonomously direct the robot to explore unknown spaces, thus expanding and updating the map server as the robot moves. As this project uses frontier based exploration, these direct the robot to frontier edges, which are lines that separate known and unknown spaces in occupancy grid map. There are various exploration strategies that can be used. Here, we have used RRT path planning algorithm that samples the space using randomly generated points, these are used to extend edges in a tree like structure, that contains nodes and edges. This enables the robot to follow the explored tree-like structure as the new nodes are explored. As mentioned, RRT is biased towards unexplored spaces, when used for exploration of unknown spaces this ensures that all of the map is covered and explored. We have implemented this strategy, in a ROS based environment using a single robot, but can be extended to multiple robots as well as exploration of 3-D spaces. We have paired this autonomous exploration strategy with other commonly used navigation components, importantly Simultaneous Localization And Mapping (SLAM) and path-planner module. Here we have used inbuilt ROS SLAM algorithm, gmapping, which updates the map simultaneously as the map is being explored.

The final goal of the project is to run turtle bot in an unknown and unexplored environment, so that the robot moves autonomously and creates the map. The robot can then go to any goal in the newly generated map.

2 Methodology

This project is divided into three different modules; RRT-based frontier detector module, filter module and task allocator module. RRT based frontier detector module has 2 components

- Local Frontier Detector: it starts from a single initial vertex V = x_{init}, and the edge set E = , at each iteration a random point x_{rand} belongs to X_{free} is sampled. The first vertex of the tree which is nearest to x_{rand} is found, this point is called $x_{nearest}$ belongs to V. Then, the Steer function generates a point xnew. The GridCheck function checks if x_{new} lies in the unknown region, or if any point of the line segment between x_{new} and $x_{nearest}$ lies in the unknown region. If either of the above conditions is true, then x_{new} is considered as a frontier point. Once the tree reaches an unknown region, a frontier point is marked and the tree is reset. This process happens during a robot's motion, therefore the tree grows from a new initial point each time it resets.

- Global Frontier Detector: The outline for the global frontier detector is identical to the local frontier detector's outline, but here the tree does not reset but keeps on expanding during the whole exploration period. which makes the global frontier detector algorithm similar to RRT. The global frontier detector is meant to detect frontier points through the whole map and in regions far from the robot.
- Filter Module: ll the local frontier detectors, and from the global frontier detector. The filter module first clusters the points, and it stores only the center of each cluster, the remaining points are discarded (not stored). The clustering and subsequent discarding process is needed to reduce the number of frontier points. This is done to remove the redundancy of frontier points as increase in number of points will unnecessarily use computational resources, without any gain in map information.
- Robot Task Allocator Module: This module provides the robot controller with the filtered frontier points keeping in mind the navigation cost or the cost of the path required to reach goal and information gain, which is the information acquired while travelling in the unexplored spaces of map. The information gain is quantified by counting the number of unknown cells surrounding a frontier point within a user defined radius. This radius is referred to as the information gain radius.

3 Implementation

- We have used Option 1: Practical Implementation of RRT based autonomous exploration of unknown spaces.
- A map was made in gazebo, and RRT based path planning algorithm was implemented for exploration of unknown space of the map.
- For autonomous exploration of the map, we initially have to select 4 points which approximately define the corners of the map or the area to be mapped. The final point is to be defined in the known space, after which the robot uses RRT path planning algorithm to explore the nodes and then plan the path accordingly. As RRT is biased towards unexplored nodes, the whole of map is explored.

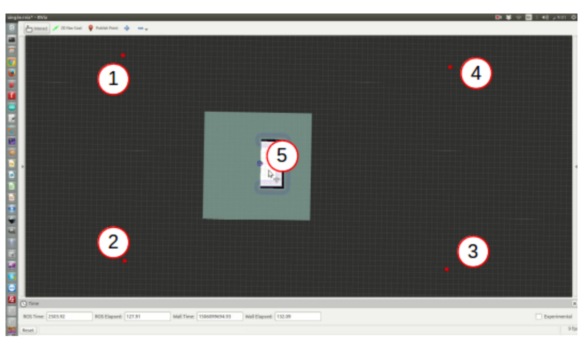

Figure 1: Sequence for "Publish Points"

- As the robot moves in the unexplored spaces of the map following the path generated by the RRT algorithm, it updates the information of the map, by using SLAM algorithm gmapping and on-board lidar, when robot explores the space, the algorithm uses lidar data to update the map information, initially the grid of the map has -1 value, if the robot encounters any obstacle it updates the grid value to 1 and if there is no obstacle, the value is updated to 0.

- The robot controller is designed in such a way that every node of the tree-like exploration nodes is visited by the robot, this helps in accurate updation of map with taking in account the obstacles.
- The explored map with updated information is updated and saved as an image, which is later used for autonomous navigation. This generated map is loaded as map server in ROS navigation stack and goal points are given as 2-D nav points.

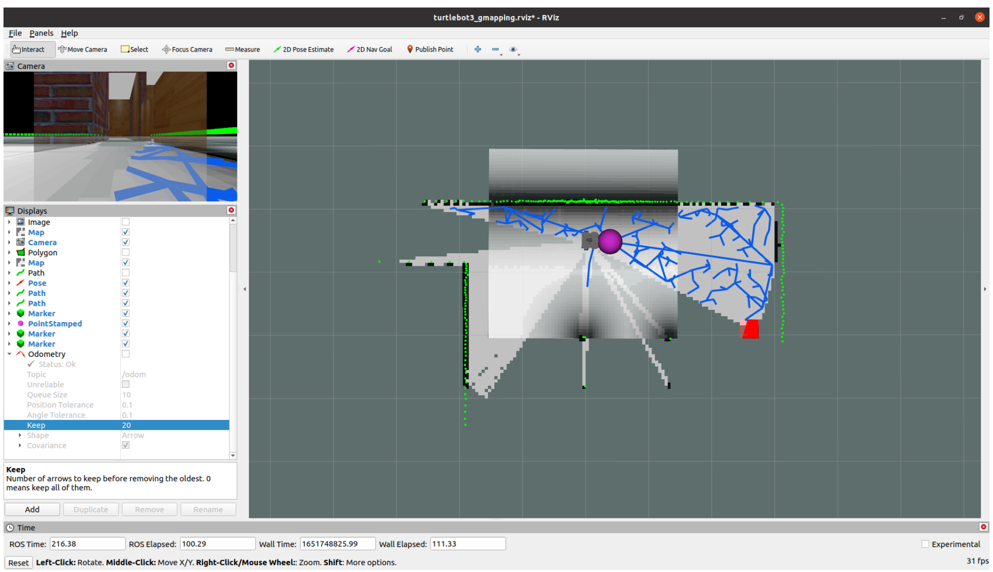

Figure 2: Troubleshooting robot autonomously moving

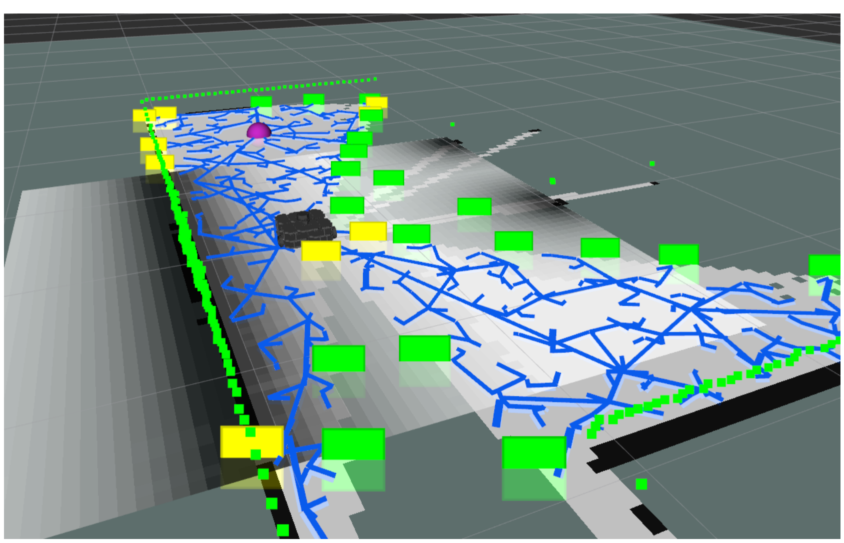

Figure 3: RRT Shown with robot

4 Conclusion and Result

- While working on this project, we analyzed various other exploration techniques mentioned in the literature cited and other sources, but later concluded that RRT was the best approach for exploration of unknown spaces.
- RRT is fast and biased towards unknown spaces hence the exploration of map is accurate and fast as the robot tends to move towards unknown spaces first.
- This method can be useful in hazardous environments, and exploration of unknown spaces and creating a map autonomously. These places can be irradiated zones, mines and other planetary surfaces.

Whole bibliography

[1] M. Fazil. "Ros autonomous slam using rapidly exploring random tree (rrt)." (), [Online]. Available: https://towardsdatascience.com/ros-autonomous-slam-using-randomly-exploring-random-tree-rrt-37186f6e3568?gi=23ca649017c0. (accessed: 04.04.2022).

[2] J. Ibarra-Zannatha, J. Sossa-Azuela, and H. Gonzalez-Hernandez, "A new roadmap approach to automatic path planning for mobile robot navigation," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 1994, 2803–2808 vol. 3. DOI: 10.1109/ICSMC.1994.400298.

[3] M. Kulkarni and J. Pittman. "Enpm661 project 5." (), [Online]. Available: https://github.com/jpittma1/ENPM661_Project5_RRT.git. (accessed: 04.04.2022).

[4] I. Noreen, A. Khan, Z. Habib, *et al.*, "Optimal path planning using rrt* based approaches: A survey and future directions," *Int. J. Adv. Comput. Sci. Appl*, vol. 7, no. 11, pp. 97–107, 2016.

[5] N. Pérez-Higueras, A. Jardón, Á. Rodríguez, and C. Balaguer, "3d exploration and navigation with optimal-rrt planners for ground robots in indoor incidents," *Sensors*, vol. 20, no. 1, 2020, ISSN: 1424-8220. DOI: 10.3390/s20010220. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/220.

[6] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1396–1402. DOI: 10.1109/IROS.2017.8202319.