Map My World (SLAM Project)

Esraa Magdy

Abstract—The simultaneous localization and mapping problem is one of the most important problems in robotics development. The importance of SLAM problems arises as in most cases the robots are deployed in an unknown environment for exploration like Mars rover where they need to create a map for the environment while at the same time localize themselves relative to the map. In this project, the SLAM problem was implemented on a mobile robot to successfully map two different worlds using Real-Time Appearance-Based Mapping (RTAB-Map)algorithm.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, SLAM.

1 Introduction

THE difficulty of the SLAM problem lies in the fact that no previous information about the environment is provided so the robot has to solve both the localization and the mapping problem concurrently. There are two forms of the SLAM problem. 1- Online SLAM where the robot solves for instantaneous pose only along with the map independent from the previous poses and measurements. 2- Full SLAM where the robot estimates the full path up to time t along with the map instead of evaluating instantaneous poses only. Another difficulty in SLAM problem is due to its nature as the parameter space could be divided into Continuous and discrete parts. The continuous parameter space is composed of the robot poses and the location of the objects which is highly dimensional whereas the discrete space contains the correspondence values. A probabilistic approach to solve SLAM problem with known correspondences using custom particle filter (Rao-Blackwellized) known as FastSLAM is able to estimate a posterior over the robot path along with the map. Another approach called GraphSLAM uses optimization techniques and transforms the SLAM posterior into a graph that represents the log-likelihood of the data. In this project Real-Time Appearance-Based Mapping (RTAB-Map) which is a Graph based algorithm will be used to solve the SLAM problem for a mobile robot in two different worlds.

2 Background

In the localization problem the map was given to the robot so the robot used the sensor data, the control actions and the given map to estimate its pose using either Kalman filter or Particle filter. However, In the mapping problem, the robot's pose was known and the map of the environment was the only unknown information using occupancy grid mapping algorithm which divides the entire space into a finite number of grids and the Binary Byes filter the robot was able to retrieve the map of the environment successfully. In SLAM problem neither the robot's pose nor the map is known so the robot has to solve the both problems simultaneously. Two approaches to solve SLAM problem will now be presented:

2.1 Fast SLAM

Fast SLAM is a probabilistic algorithm that solves the full SLAM problem with known correspondences. it uses particles to estimate a posterior over the robot path, each particle holds a possible trajectory and a map where each feature in the map is represented by a Gaussian distribution. The main disadvantage of Fast-SLAM is that it assumes known landmarks positions which makes it unable to model arbitrary environment. Another instance of Fast-SLAM algorithm is Grid-based FastSLAM that adapts the FastSLAM to grid maps which extends the application of FastSLAM to work in environment where the landmarks positions are unknown. Like FastSLAM, In Grid-based FastSLAM each particle holds a guess of the robot trajectory and maintains its own map. The map will be updated by solving mapping with known poses problem using occupancy grid map algorithm.

2.2 Graph SLAM

Graph SLAM is an optimization-based approach to create the most likelihood set of poses and map. It uses a graph approach where each robot pose or feature is represented by a node in the graph, the nodes are connected by edges which represents a type of constraint on the graph. The constraints are either a motion constraint or a measurement constraint. The Goal is to minimize the overall error present in all the constraints. The Graph SLAM is composed of two parts: The front end where the graph is constructed using odometry and sensor measurements and the data association problem is solved where the algorithm identifies if a feature is previously seen or not. The second part of GraphSLAM also known as the Back-End uses the completed graph and the constraints to solve for the most probable configuration of the robot poses and map features.

2.3 RTAB-MAP

Real-Time Appearance-Based Mapping or RTAB-Map is a GraphSLAM based algorithm that uses vision data to map the environment and localize the robot within it. There is a loop closure process that determines whether the robot has seen a location before or not. For real time SLAM problem, RTAB-map uses Memory Management to limit the number of locations considered as candidates during loop closure process. Hence, reducing the loop closure processing time. It also uses Speed-ed up Robust Feature (SURF) for feature extraction then each feature is quantized into Vocab (Visual words) when all the image is quantized its called a bag of words. The loop closure process and bag of words approaches improved the efficiency of RTAB-Map over FastSLAM and GraphSLAM. RTAB-Map could map large and complex environments successfully.

3 Simulations

3.1 Robot's hierarchy

For this project a mobile robot was used to implement the SLAM problem on it. Each component of the robot has an assigned frame. The frames' hierarchy is indicated in the following figure.

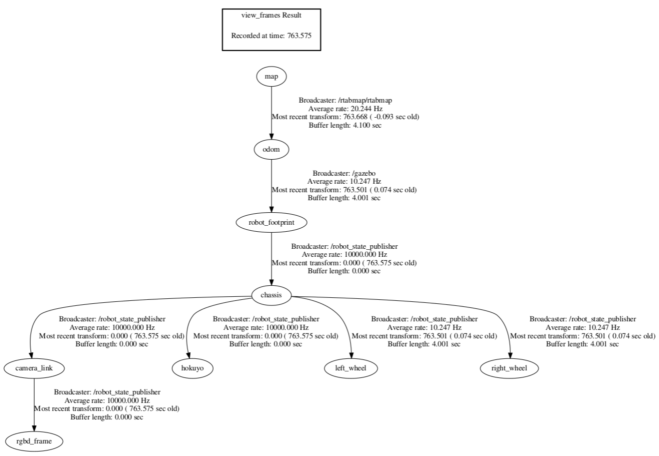

Fig. 1. frames

3.2 Gazebo worlds

The algorithm was tested on two different worlds, the Kitchen dining world and another custom world that contains a grocery store , a playground , a cabinet a fountain and a doll. The custom world was created to look like an outdoor environment to test the algorithms on both indoor/outdoor environments.

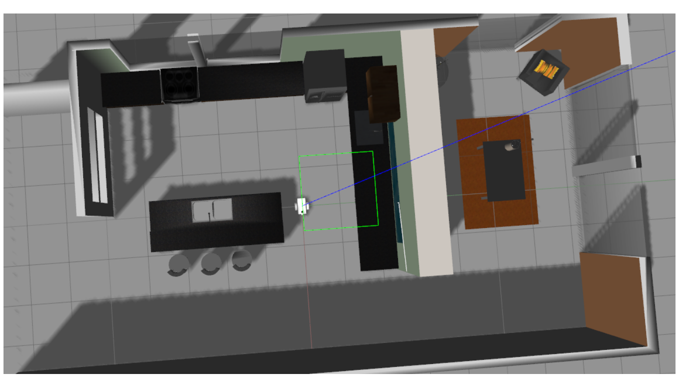

Fig. 2. Gazebo world

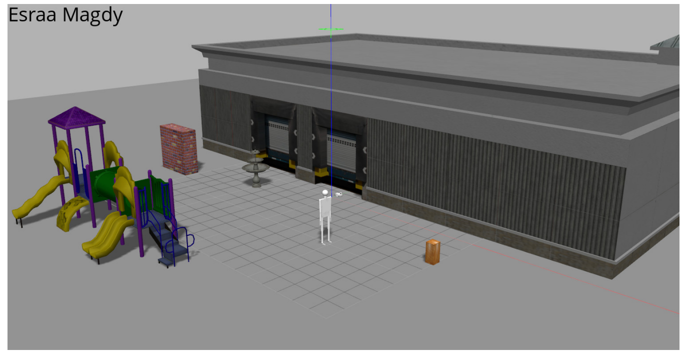

Fig. 3. My Custom World

The robot was equipped with a laser scanner and RGB-D kinect like camera. The ROS Gmapping package configuration depends on a laser scanner sensor while RTAB-MAP ROS package depends on the kinect. That was the main reason to choose the presented sensors configuration.

4 Results

4.1 Kitchen dining world

The following figures represents the 2D and the 3D map generated for the kitchen world.

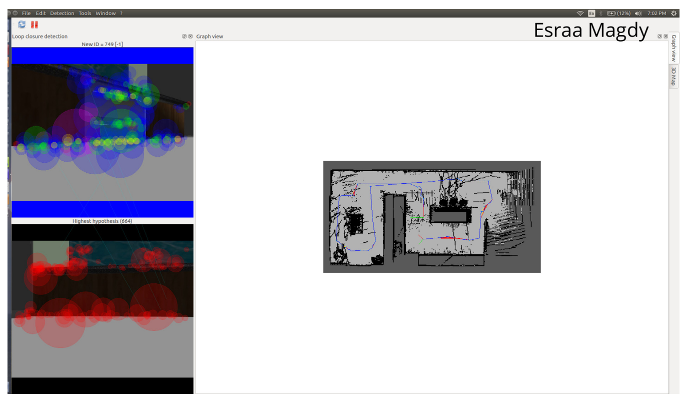

Fig. 4. 2D map

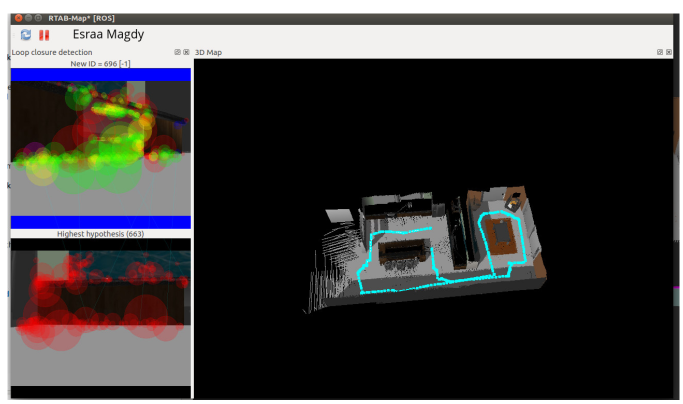

Fig. 5. My Custom World

4.2 My World

The following figures represents the 2D and the 3D map generated for the custom world.

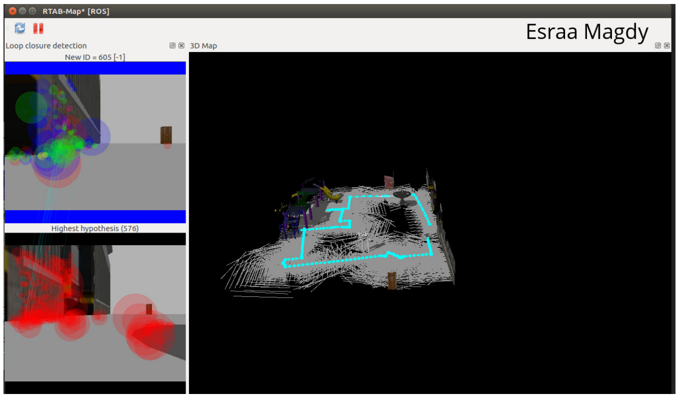

Fig. 6. RTAB-Custom World

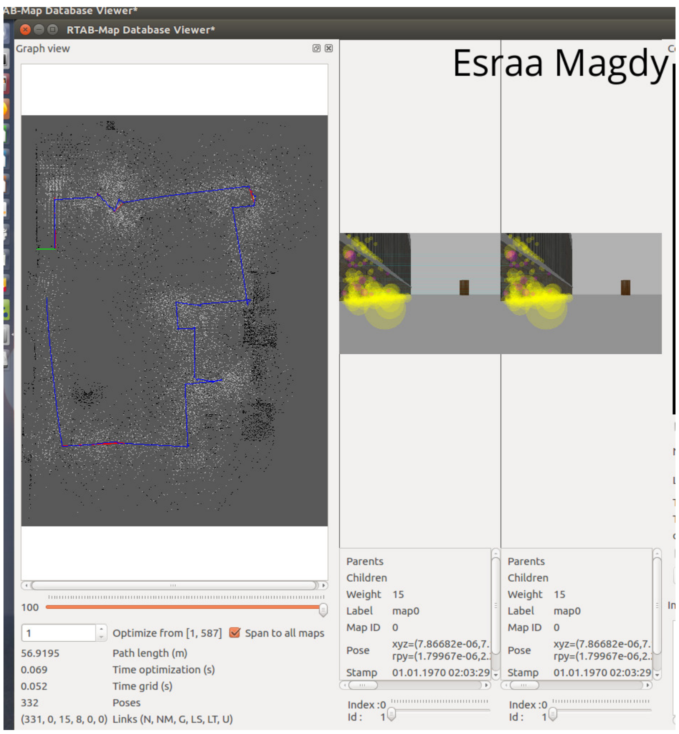

Fig. 7. My Custom World

5 Discussion

The ROS Package was created using the same hierarchy suggested by ROS in [1]. The mapping results for the provided world was a bit better compared to the custom world where the map might get corrupted during navigation. As the kitchen world has a more organized structure that might be a reason to make it more suitable for mapping tasks.

6 Conclusion / Future work

RTAB-Map is computationally efficient algorithm. it can be used on a TX2 to deploy the slam package on a real robot. it also presents the loop closure feature which enables the robot to know if it has previously seen the scene or not. While RTAB-Map is able to solve the SLAM problem in static environment, it might fail if the environment became dynamic. An extension for this work might include an improvements over the current algorithm to make it able to solve the SLAM problem in a dynamic environment.

7 References

[1] http://wiki.ros.org/Packages.