Map My World Robot

A SLAM project for the Robotics Nanodegree Program, Udacity

Muhammad Khalaf

Abstract—Simultaneous Localization And Mapping or SLAM is the problem of creating a map of unknown environment by a robot while trying to localize the robot simultaneously. In this project, ”Map My world Robot”, one of the best SLAM techniques is used, which called RTAB-Map. Using the rtabmap ROS package, a simulation is carried on two gazebo worlds which are mapped using a robot model and visualized with rviz. The outputs of this package are a 3d map of point clouds and a 2d occupancy grid map.

Index Terms—SLAM, Robot, Udacity, ROS, rtabmap, Gazebo.

1 Introduction

Simultaneous Localization And Mapping is one of the most fundamental and yet most difficult problems in robotics. It occurs when the robot does not have any information about his location and does not have a map for the environment. Its his responsibility to map this unknown environment while simultaneously localizing itself [1].
There are two forms of the SLAM problem, online SLAM and offline SLAM. The online SLAM problem computes a posterior over the current pose along with the map and the full SLAM problem computes a posterior over the entire path along with the map.
Another important characteristic is the nature of the problem. SLAM has continuous and discrete natures. The robot is continuously taking measurements and senses its environment to locate its pose. So robot poses are continuous. The discrete part comes from the correspondence of objects. If the robot sees an object that is seen before, it will make him know that he has been in this location. And if it is the first time to see this object, then the robot is visiting a new location. So it is a discrete state.
These are the most common SLAM algorithms:-

- Extended Kalman Filter (EKF)
- Sparse Extended Information Filter (SEIF)
- Extended Information Filter (EIF)
- FastSLAM
- GraphSLAM

FastSLAM and GraphSLAM are illustrated in greater details in the next section.

2 Background

SLAM has many challenges. One major challenge is the highly dimensional space. As the robot moves, it sees a lot of objects, and it has to solve the correspondence problem with all these objects, here the number of variables will increase exponentially with time.
SLAM, in both of its two forms, can be solved with different approaches. FastSLAM and GraphSLAM are two common methods that a roboticist should bear in mind when it comes to solving such a problem. With SLAM difficulty, comes the complexity of the used algorithms. Here in this section, the two mentioned approaches are explained briefly.

2.1 FastSLAM

FashSLAM uses a particle filter approach to solve the full SLAM problem with known correspondences. Each particle holds an estimation of the robot trajectory and a map, to solve the mapping part of SLAM with known poses. So the problem now is separated into two independent problems to estimate the features of the map. Then a low dimensional extended Kalman filter is used to solve these two problems. This approach assume that the environment has known landmarks, and this is one of the disadvantages of FastSLAM. However, with the grid mapping algorithm the environment can be modeled using grid maps without predefining any landmark position. An instance of FastSLAM is Grid-based FastSLAM which adapts FastSLAM to grid maps. The SLAM problem can be solved in an arbitrary environment. Grid-based FastSLAM uses Monte Carlo Localization and occupancy grid mapping to solve the SLAM problem.

2.2 GraphSLAM

GraphSLAM is another approach to solve the full SLAM problem. The main advantage of GraphSLAM over FastSLAM is it has no particle limitations, so it is more accurate. In GraphSLAM, a graph is created representing the entire path of the robot along with features from the environment as the nodes of the graph, some sort of motion and measurement constraints tie these nodes together like strings. Then the GraphSLAM algorithm tries to minimize the forces arise between these strings to reach the most stable state. Minimization is done using optimization algorithms such as: Levenberg Marquardt and stochastic gradient decent.

RTAB-Map, which stands for Real-Time Appearance-Based Mapping, is one of the graph based SLAM techniques which uses data coming from vision sensors for localization and mapping. It uses loop closure detection, a process indicating that a location has been seen before. Which in turn uses a bag-of-words approach to extract features from images, compare them, and associate some sort of linkages to these images that have a number of common features above a predefined threshold number.

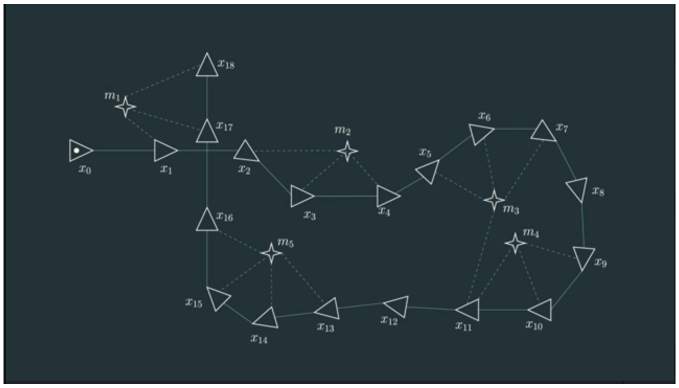

Fig. 1. a graph with poses, features and constraints

Fig. 2. loop closure uses a bag-of-words approach

3 Scene and robot configuration

Personal gazebo world is created using some models from gazebo database.

Fig. 3. gazebo world

To add some features to the environment, four different objects are used in the middle of the world. The borders of the world are different house models. Different models are used to add more features, and so as not to confuse the robot, because when the robot sees the same window design of a house for example in two different locations, the localization output would have very bad results.

Fig. 4. bad localization and mapping when using the same models in many different places

The robot model is the benchmark model used in the localization project [2]. An RGBD camera is added for the rtabmap to create the 3d map from point clouds. The next two figures shows the robot frames from both rviz and tf.

Fig. 5. robot frames from Rviz

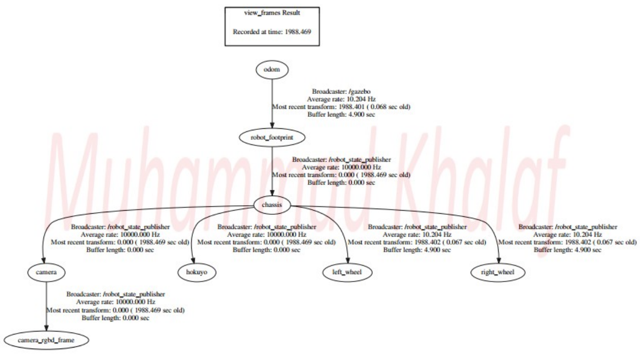

Fig. 6. robot frames from tf

4 Results

4.1 kitchen_dining world

The robot mapped the kitchen and dining world by going over the same path three times. This will allow the robot to extract more feature association, thus increasing number of loop closures. However, it seems that from the first path the robot was able to map the world efficiently.
At the end, 366 global loop clusters were found.

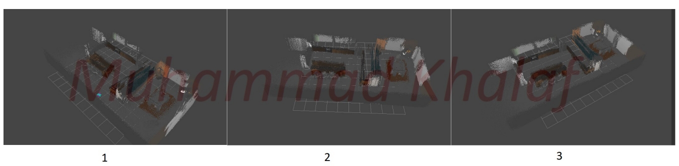

Fig. 7. kitchen and dining world, 3D map

Fig. 8. kitchen and dining world, 3D map points style

The provided gazebo model has no collision attributes it has only visual ones, and it is considered as an obstacle-free environment. So the 2D map was not created correctly because the laser scanner could not see any object in this world. on the contrary to the RGBD camera which relies on the visual elements in the scene.

4.2 Personal world

This world is mapped without moving over the path many times as done before. Consequently, the number of loop

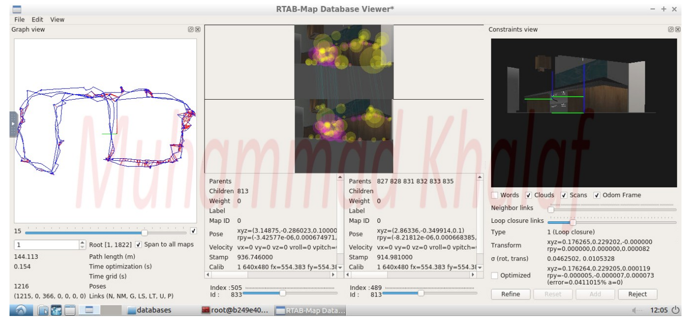

Fig. 9. 2D map shows no objects

closures detected was very small, only four loop closures were detected. But it was good enough for rtabmap to create the 2D and 3D maps.

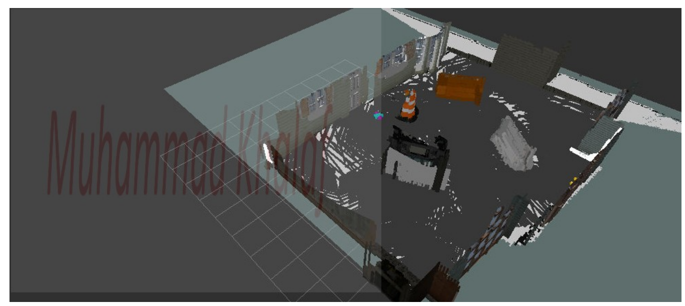

Fig. 10. Personal world, 3D map points style

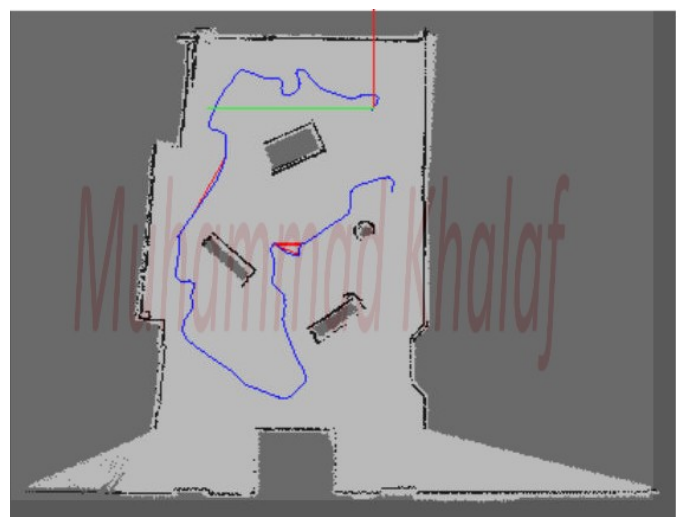

Fig. 11. Personal world, 2D map

5 Discussion

One important factor that affects the mapping output greatly is the position of the sensors. The RGBD camera and the laser scanner need to be placed in a relative higher place to be able to see the world with a better perspective. The object in the next figure could not be seen by the laser scanner, so it treated as an empty space in the 2D map and an occupied space in the 3D map.

Fig. 12. dumpster model from gazebo database

The overall mapping of the kitchen_dining world was much better because it has a rich feature environment. The robot was able to detect many loop closures without failing in localizing itself. While in the second world, the robot failed many times.

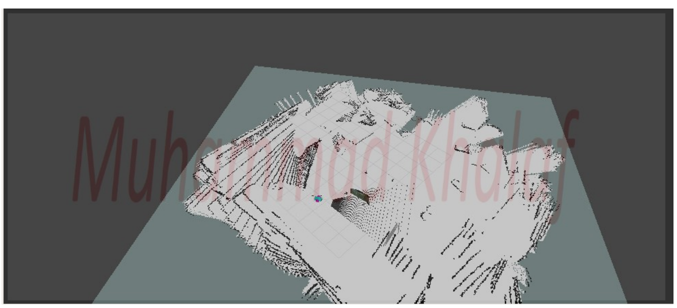

Fig. 13. SLAM failed to map the environment

6 Future work

This work can be extended with more new feature as SLAM can be useful in many situations. In mining for example, a robot can map an unknown subterranean tunnel systems to help engineers and workers do their job in a more efficient way [3]. SLAM can also be used in dangerous places with a rescue robot, or in planets never visited before [4].
Such applications require a robust robot design that is able to navigate a rough terrain environments.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. 2000.
[2] gazebosim.org, "Gazebo : Tutorial : Make a mobile robot." http://gazebosim.org/tutorials?tut=build$_{r}obot$.
[3] DARPA, "Darpa subterranean challenge aims to revolutionize underground capabilities." https://www.darpa.mil/news-events/2017-12-21.
[4] NASA.gov, "Stmd: Centennial challenges." https://www.nasa.gov/directorates/spacetech/centennial$_{c}hallenges/sample_{r}eturn_{r}obot/index.html$.