Map My World Robot

Quah Yu Soon

Abstract—This documentation discusses about FastSLAM and GraphSLAM which are two most common Simultaneous Localization and Mapping approaches that could accurately construct a map of the environment while simultaneously localizing robot itself relative to the map. An experiment that implemented Real-Time Appearance-Based Mapping on two different Gazebo worlds would be carried out for mapping performance evaluation.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, SLAM, RTAB-Map.

1 INTRODUCTION

ROBOT is provided a map of its environment in order to perform localization. However, under circumstances such as map is not up-to-date, map is unavailable and frequent change in map environment, robot localization would not be working well. To overcome such problems, Simultaneous Localization and Mapping (SLAM) was introduced. As the name implies, this approach constructs map of environment robot operates in while simultaneously localizing the robot relative to the map. Mapping generates map of the environment robot perceived through analysis of movement and measurement data.

In the experiment, a benchmark Gazebo world has been provided by Udacity and it was a kitchen dining setting. Another Gazebo world would be self-designed which provides students an opportunity to demonstrate their skills in building a simulated environment using Gazebo. A robot model provided by Udacity would be teleoperated around simulated environments to collect mapping data.

2 BACKGROUND

SLAM problems are more difficult than localization or mapping since both the map and robot poses are unknown. There are some algorithms catered for solving SLAM problems:

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

Among these algorithms, FastSLAM and GraphSLAM would be discussed in detail in the upcoming sections.

2.1 FastSLAM

FastSLAM estimates a posterior over the trajectory using a particle filter approach. Each particle will hold a guess of the robot trajectory. In estimating the posterior over the map features, both FastSLAM 1.0 and 2.0 algorithms use a low dimensional Extended Kalman filter. The FastSLAM 2.0 algorithm overcomes the inefficiency of FastSLAM 1.0 by imposing a different distribution, which results in a low number of particles. The drawback of both FastSLAM 1.0 and 2.0 is that they are not capable of modelling an arbitrary environment as both algorithms must always assume that there are known landmark positions.

By extending the FastSLAM algorithm to occupancy grid maps, the environment is possible to be modelled using grid maps without predefining any landmark positions. Such algorithm is called Grid-based FastSLAM. Similar to FastSLAM 1.0 and 2.0 algorithms, each particle holds a guess of the robot trajectory. The difference is that map estimation is done using occupancy grid mapping algorithm instead of low dimensional Extended Kalman filter.

2.2 GraphSLAM

GraphSLAM uses graphs to represent robot trajectory and the environment. Robot pose and its orientation at specific timestamp are represented by a single triangle node. A feature in the environment such as a landmark or an identifiable element of the environment is represented by a star. Two nodes are connected by an edge which represents soft spatial constraint between two robot poses. Constraints come in two forms: motion constraints and measurement constraints. Motion constraints tie together two successive poses and they are represented by solid lines. Measurement constraints tie together a feature and a pose and they are represented by dashed lines. Nodes, stars and edges are added to the graph as the robot traverses the environment. After constructing graph, graph optimization is performed to find the node configuration that minimizes the overall error present in all the constraints. The information matrix and information vector are two data structures that store information from linear constraints. As for nonlinear motion and measurement constraints, they must be linearized before they can be added to the information matrix and information vector. This could be accomplished by using Taylor Series as well, but it inevitably introduces some errors. To reduce the error, linearization process are repeated several times due to the fact that solution is improved with every iteration.

One of GraphSLAM approaches is Real-Time Appearance-Based Mapping(RTAB-Map). It uses data collected from vision sensors to localize the robot and map the environment. A process called loop closure is used to determine whether the robot has seen a location before. RTAB-Map implements multiple strategies to allow for loop closure which could take longer time when number of images to be compared increases to be done in real-time. Furthermore, RTAB-Map supports three dimensional mapping.

2.3 Selection of Mapping Algorithm

GraphSLAM has improved accuracy over FastSLAM. From the aspects of 3D mapping and good memory management, RTAB-Map was selected as mapping algorithm to be used in the experiment.

3 Scene and Robot Configuration

Two different Gazebo worlds have been used in the mapping experiment. The layout of each scene will be discussed in upcoming sections. In collecting sensory information, same robot model has been used. This robot has a cuboid base, installed with two cylindrical wheels, a RGB-D Camera as well as a Hokuyo laser range finder.

3.1 Benchmark World

In the benchmark world (See Figure 1), there is a room for kitchen and dining. The small living room is laid out with a sofa, a table and an ornament at one of the corners.

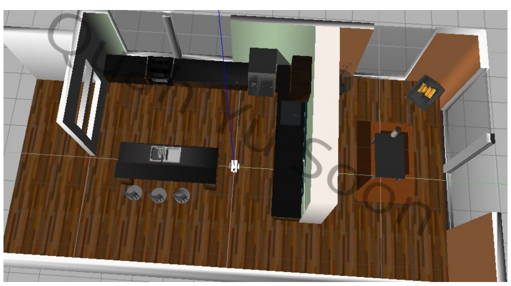

Fig. 1. Benchmark Gazebo World

3.2 Self-Designed World

As can be seen in Figure 2, the scene is surrounded by walls. Inside the scene, there are total two trees, four cabinets, three tables, three persons and so on.

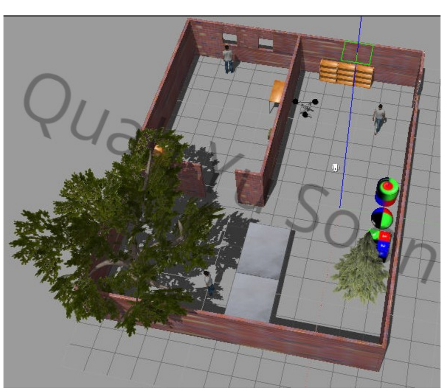

Fig. 2. Self-Designed Gazebo World

4 ROS Package Structure

A custom ROS package has been created to launch Gazebo world, robot model, teleop Python script, RTAB-Map ROS package and Rviz virtualization tool. This ROS package is named as slam_project and it follows a conventional directory structure. All the directories and the files each directory contains are listed in Table 1. Figure 3 shows a tf tree of this package indicating how each coordinate frame is connected.

TABLE 1
slam_project Package Structure

Directory	Files
launch	mapping.launch, robot_description.launch, rviz.launch, teleop.launch, world.launch
meshes	hokuyo.dae
scripts	teleop
urdf	udacity_bot.gazebo, udacity_bot.xacro
worlds	custom_environment.world, kitchen_dining.world

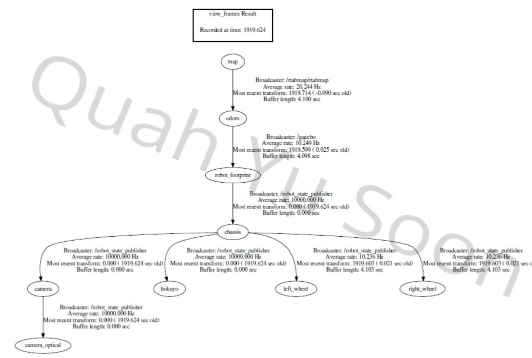

Fig. 3. TF Tree

5 Results

The results below were taken after roaming around the entire environment twice.

5.1 Benchmark World

Figure 4 shows final 3D map of benchmark world virtualized in Rviz after mapping. Figure 5 shows final 2D grid map of benchmark world and the path robot has traversed indicated by a blue line within Graph view panel of rtabmap-databaseViewer.

Fig. 4. Final 3D map of Benchmark World

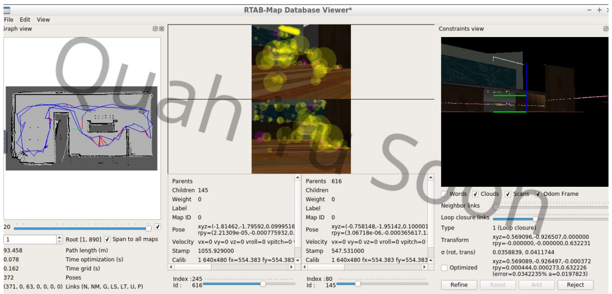

Fig. 5. Final 2D Grid Map of Benchmark World (On The Left)

5.2 Self-Designed World

Figure 6 shows final 3D map of self-designed world virtualized in Rviz after mapping. Figure 7 shows final 2D grid map of self-designed world and the path robot has traversed within Graph view panel of rtabmap-databaseViewer.

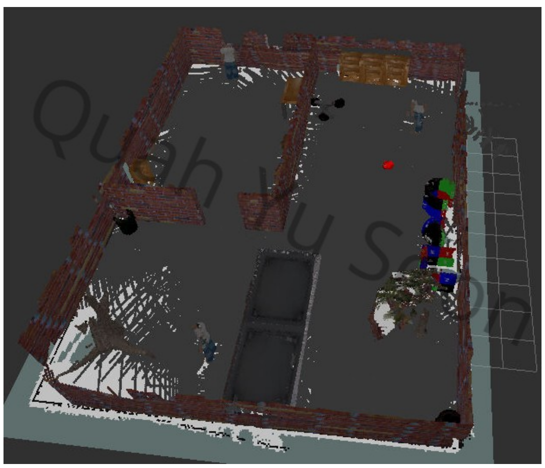

Fig. 6. Final 3D map of Self-Designed World

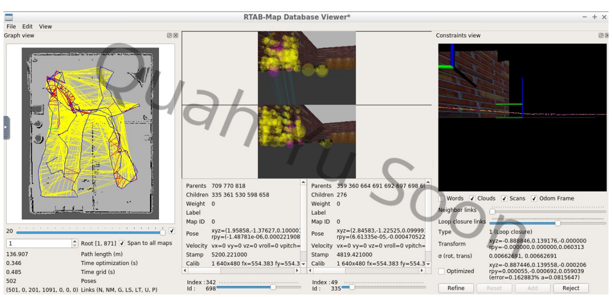

Fig. 7. Final 2D Grid Map of Self-Designed World (On The Left)

6 Discussion

RTAB-Mapping performed well on benchmark Gazebo world once the mapping node was subscribed to correct topics. Both 2D and 3D maps were constructed accurately. As for self-designed Gazebo world, when applying same mapping parameters, spiraling 3D map was produced. However, after changing the value of Reg/Strategy parameter from 0 which represents visual registration to value 2 which represents visual and Iterative Closest Point registrations, 3D map was constructed without spiraling effect.

In term of mapping accuracy, 3D map constructed from benchmark world seemed to be more accurate than that constructed from self-designed world. In the 3D map of self-designed world, there are some leaves on the tree having same color as wall and mismatched surface colors on a cube and a cylinder objects (See Figure 8). One may believe 3D map of benchmark world has higher accuracy because its environment contains more unique and identifiable features.

Fig. 8. Errors in self-designed world's 3D map

7 Conclusion / Future work

With RTAB-Map, maps of two different simulated environment were constructed accurately in real-time. On the other hand, RTAB-Map provides visual odometry feature which is essential to robots that do not have wheel encoders such as aerial robots. For future work, such mapping shall be tested in real-life environment to identify other potential issues that could not found in simulated environment.

SLAM could be best alternative to GPS as GPS does not work well indoors and it is only accurate up to a few meters at outdoors. When autonomous car is equipped with SLAM, collision prediction would become better since car position and surrounding objects could be identified with higher accuracy offered by SLAM.