Map My World Robot

Sim Kae Wanq

Abstract—This project aims to successfully perform Simultaneous Localization And Mapping (SLAM) leverage GraphSLAM to generate 2D occupancy grids and 3D octomaps of two simulated environments in Gazebo. A two-wheel mobile robot, equipped with RGB-D camera (Kinect) and 2D Lidar sensors (Hokuyo) was simulated to traverse the two Gazebo world environments. The mobile robot mapped the environment using Real-Time-Appearance-Based Mapping (RTAB-Map) GraphSlam approach. After the robot traversed both environments, maps were evaluated on accuracy.

Index Terms—Robotics, IEEEtran, Udacity, LaTeX, ROS, mapping, RTAB-Map.

1 Introduction

Autonomous robot have to understand it's environment and it's pose in this dynamic world, when doing task via series of sensors. In robotics, Simultaneous Localization and Mapping (SLAM) is the computational problem of creating and updating map of an unknown environment while simultaneously keeping track of robot's pose within the environment. [1]

In this project, two virtual 3D environment, one provided and one custom, were simulated and mapped by a prebuilt mobile robot using SLAM. The SLAM algorithm used in this project was Real-Time-Appearance-Based Mapping algorithm. Constructed 2D map and 3D map were then evaluated.

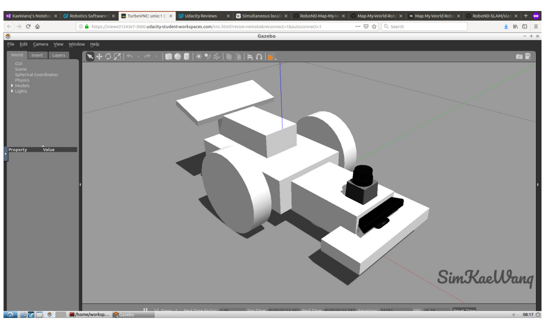

Fig. 1. Robot model.

2 Background

Acquiring map is challenging due to number of reasons, including

- Unknown map
- Huge Hypothesis Space
- Size
- Noise
- Perceptual Ambiguity
- Cycles

SLAM is a challenging problem that decided and also constraint by the hardware used on the robot. There are several types of SLAM including:

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

In this papers, the algorithms that will be cover includes, Occupancy Grid mapping, FastSLAM, and GraphSLAM.

2.1 Occupancy Grid Mapping

Occupancy Grid Mapping address the issue of creating reliable maps from noisy and uncertain measurement data, with the belief that robot's pose is known. The concept of this algorithm is to represent the map as a field of random value variables, organized in uniform spaced grid. Each unit of grid represents the occupancy of the location.

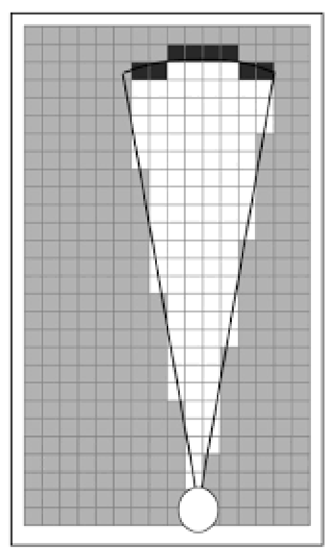

Fig. 2. Occupancy grid mapping.

2.2 Grid-based FastSLAM

The basic idea of FastSLAM is to preserve a set of particles to approximate a posterior over the trajectory and it uses low dimensional EKF to solve independent features od the map which are modeled with a local Gaussian. FastSLAM estimates the robot's full path, solves the full SLAM problem. However, each particle in FastSLAM approximates the robot's immediate pose, thus this technique also solves the online SLAM problem.

The disadvantage if FastSLAM os having to always assume that there are known landmarks, thus hindering its ability to model an random environment. However, grid-based FastSLAM able to fill this gap by keeping particle filter to solve the localization problem in SLAM, and at the same time, extending the algorithm to occupancy grid mapping. Thus, it can acquire the environment using grid maps, without predefining any landmark position.

2.3 GraphSLAM

GraphSLAM address the full SLAM problem, which means the algorithm recovers the entire path and map. The advantage of this algorithm is reduced onboard processing capability and the improved accuracy compared to FastSLAM, especially in large environments. GraphSLAM can work with all of the data at once to find the optimal solution.

GraphSLAM is simpler, it extracts from a data set of soft constraints, represented by a graph as shown in fig3. Then, it recovers the map and the robot path by solving these constraints into a generally reliable estimate. Usually these these non-linear constraints will be linearized. Process keep iterating until the optimal solution converges.

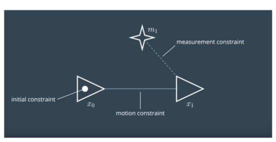

Fig. 3. GraphSLAM

RTAB-Map is a graph-based SLAM method that that uses data gathered from vision sensors to localize the robot and map the environment in real time. This algorithm uses a concept - loop closure to establish is a robot has seen a location before. RTAB-Map is optimized for large scale and long-term SLAM by using numerous strategies to allow loop closure to be done in real time.

3 Scene and robot configuration

In this section, the provided virtual scene - kitchen and dining and custom virtual scene, as well as the robot model.

Explain how your personal Gazebo world was created and what is the layout of it. Justify your choice of robot parameters, sensor location, and how you decided to configure your package structure.

Student explains how the gazebo world was created by providing an overview of the layout of items in his/her customized Gazebo world. Student also describes the robot's parameters, sensor features, and reasoning on the package structure

3.1 Scene

3.1.1 Kitchen and dining

This is the provided virtual scene that consists of a kitchen and a living room.

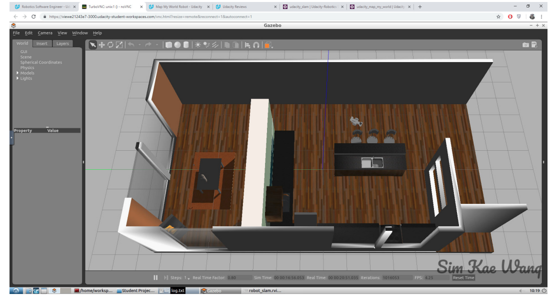

Fig. 4. kitchen and dining room

3.1.2 Custom room

Figure below shows a virtual room that was custom created using Gazebo. This room consists of 2 tables, 2 pine trees, 2 book shelves, a fire truck and a traffic cone. To limit the space, room was built with 3 grey walls and form a open square. These objects were fixed and let the robot have more features to identify based on appearance.

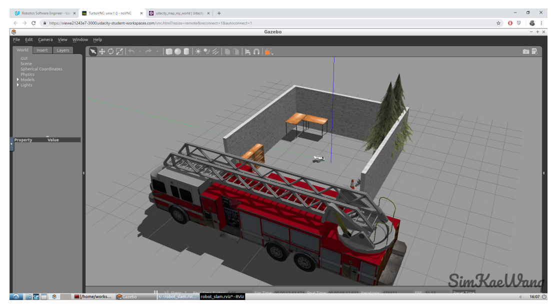

Fig. 5. Custom room

3.2 Robot

As shown in figure 1, the robot used in this project equipped with two actuators, 1 RGB-D camera - Kinect and 1 2D lidar sensor - hokuyo sensor. Instead of using a normal RGB camera, a RGB-D camera allows robot to detect the depth of its environment. Along with the 2D lidar, this allows robot to leverage the RTAB-Map ROS package and perform GraphSLAM. Both sensor were installed on the front part of robot, to avoid laserscanner covered by the actuators. Since mapping was done manually, teleop package was included and launched to control robot.

Figure below shows the TF tree of robot model.

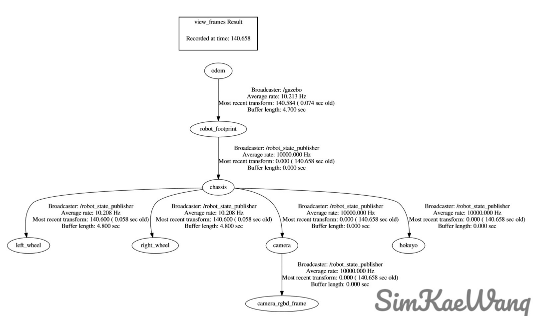

Fig. 6. kitchen and dining room

4 Results

This section show the mapping result - 2D grid map and octomap, and the evaluation result via rtabmap-databaseviwer.

4.1 Kitchen and dining

When RTAB-Map started, the robot traversed the room 3 to 4 loops to collect most of the features. Most of the features in the octomap were clear and identifiable. Evaluation of the constructed map showed in Fig 9. Number of global loop closures is 223.

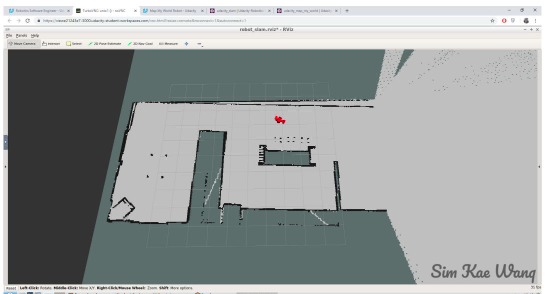

Fig. 7. Grid map of kitchen and dining room

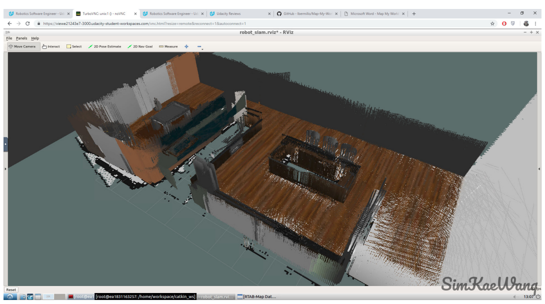

Fig. 8. Octomap of kitchen and dining room

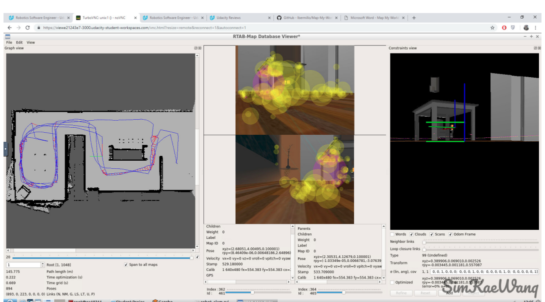

Fig. 9. Evaluation of kitchen and dining room mapping

4.2 Custom room

Figures below shows the 3D model of custom room (Fig. 10), the constructed 2D grid map (Fig. 11) and constructed octomap (Fig. 12).

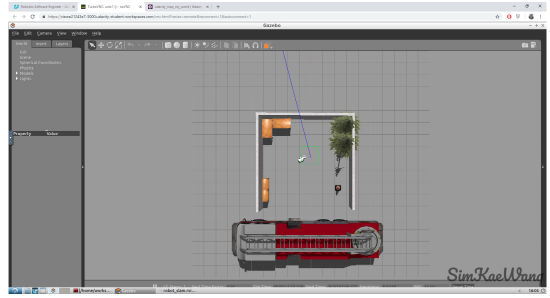

Fig. 10. Custom room

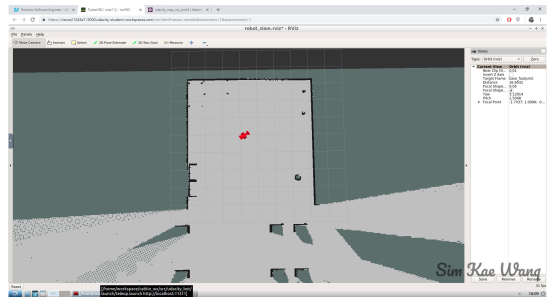

Fig. 11. Grid map of custom room

5 Discussion

When robot traversed the environment one loop, the number of global loop closures was not optimal, as not enough features were collected in the map. Fig. 13 show the evaluation when robot traversed only one time.

Mapping the kitchen and dining room was easier than mapping the custom room, even though the custom room looks more neat and simpler. Kitchen and dining room environment is conducive to the success of the RTAB -Map because it contains feature-rich objects. In the other hand, when mapping the custom room, it's easier to get error and noise like Fig. 14. Robot tend to lost itself when mapping the environment and constructing noise full map. This problem was solved by changing the Detector Strategy from SURF to ORB, when ORB is a good alternative in computation cost, matching performance and patent (SURF is patented but ORB is not). [2]

6 Future work

For simulation, the robot can be improved by changing the design and structure to better adaptive to multi-terrain.

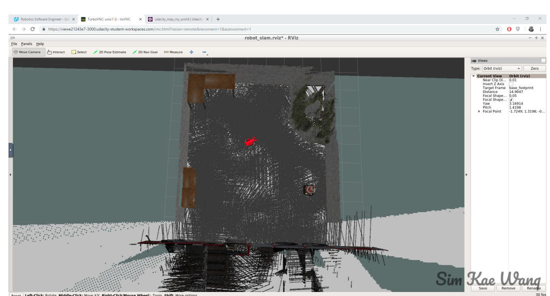

Fig. 12. Octomap of custom room

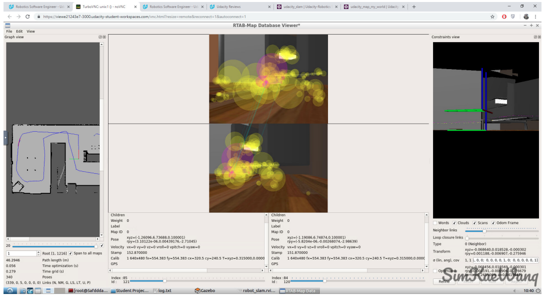

Fig. 13. When traversed only 1 loop, the number of G is 5

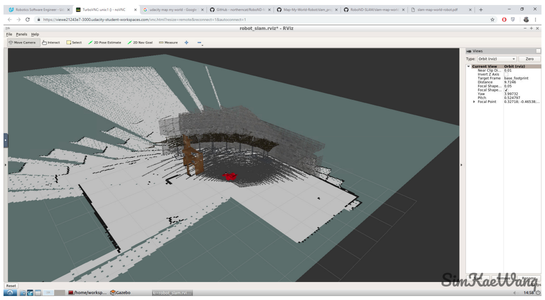

Fig. 14. Robot lost itself when doing mapping

Other possible future work includes actual deployment on mobile robot. Platform like Jetson TX2 provide better computing performance to complete the mapping task. By deploying in actual world, it is easier to get better understanding and evaluation on the performance or RTAB-Map.

REFERENCES

[1] "Simultaneous localization and mapping."
[2] "Orb (oriented fast and rotated brief)."