Map My World Robot

Carl Sosa-Rivera

Abstract—This project aims to successfully perform SLAM leverage GraphSLAM to generate 2D occupancy grids and 3D octomaps of two simulated environments in Gazebo. A two-wheel robot equipped with RGB-D and 2D Lidar sensors (configured as a Kinect camera and a Hokuyo lidar, respectively) was simulated to traverse two Gazebo world environments, one provided, and another that was custom built. The robot maps the environment using the Real-Time-Appearance-Based Mapping (RTAB-Map) GraphSLAM approach. After the robot traversed both environments, maps were evaluated on accuracy.

Index Terms—Robotics, SLAM, RTAB-Map, Udacity, IEEEtran.

1 Introduction

In robotics, for a robot to be fully autonomous it must adapt and understand its environment as it changes and be able to recover when it obtains noisy data from sensors. Simultaneous Localization and Mapping (SLAM) provides a robot with the capability to dynamically map its environment as it relates to it. When implemented correctly, SLAM solves the localization problem to map the environment, and maps the environment to solve the localization problem. This is all done as an iterative cycle.

In this project, two 3D world environments are to be simulated, tested, and mapped using SLAM. The environment provided, and set to be the benchmark, is called kitchen dining. Another environment shall be designed and developed in Gazebo and saved in world format; this one is called my world. The two environments shall be mapped in both 2D and 3D by using Gazebo and Rviz simulation frameworks.

2 Background

Given that a map is needed for localization, but at the same time the robots location and orientation is needed for mapping, this has often been considered the chicken or the egg problem. SLAM is a challenging problem, as it has to quickly solve two related problems simultaneously and back-to-back. There are many approaches to perform SLAM such as:

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

The two most commonly used approaches in robotics are FastSLAM and GraphSLAM. Both of these, solve the SLAM problem well, although in different ways.

2.1 FastSLAM

The FastSLAM algorithm can solve the full SLAM problem with known correspondences. Using a custom particle filter approach along with a low dimensional Extended Kalman Filter, FastSLAM is able to estimate the trajectory of the robot which will allow it to map the environment concurrently.

2.2 GraphSLAM

GraphSLAM is a SLAM algorithm that solves the Full SLAM Problem. This algorithm recovers the entire path and map, which allows it to consider dependencies between current and previous poses. GraphSLAM has a few advantages over FastSLAM. GraphSLAM improves upon the need of onboard processing capability, while still improving accuracy over FastSLAM. Since GraphSLAM is able to retain information from past locations, it proves an advantage over FastSLAM which uses less information and has a finite number of particles.

The Real-Time Appearance-Based Mapping algorithm is a GraphSLAM approach and will be used in this project to perform SLAM. This algorithm uses data collected from sensors to localize the robot and map the environment. In RTAB-Map a process called loop closure is used to allow the robot to determine if the location has been observed before. While the robot continues to traverse through its environment the map continues to grow. For other Appearance-Based methods, the robot continues to compare new images to past images to identify whether it has been at that location before. This, over time, increases the number of images for comparison, causing the loop closure process to take longer, proportionally increasing complexity along the way. However, RTAB-Map is optimized for large-scale and long-term SLAM, allowing loop closure to be processed fast enough to be known in real-time without dramatically affecting performance.

3 Scene and Robot Configuration

In order for the robot to perform SLAM, it must have a World to perform it in. For this reason, two environments will be used to test and simulate a robot performing SLAM and to generate 2D and 3D maps from them.

3.1 Kitchen and Dining Scene

The Kitchen and Dining Scene world was provided and will be used as a benchmark for experimentation purposes (See Fig. 1).

Fig. 1. Kitchen and Dining Scene

3.2 My World Scene

The My World Scene was designed in Gazebo to be used as the experimental environment to deploy a robot for performing SLAM. This scene was developed with the RTAB-Map Appearance-Based feature in mind. Therefore, this scene is dramatically feature-rich to help the robot easily identify where it is based on appearance, and map the environment more accurately. This scene includes a fountain, a tree, a postal mailbox, tables, boxes, totes, people, and even floor colors that allow the robot to detect more features (See Fig. 2).

Fig. 2. My World Scene

3.3 Robot Configuration

The robot used for this project was initially built to solve the localization problem in a previous project, which has a cylindrical shape, with two cylindrical wheels, and to spherical casters. This robot has been repurposed to perform SLAM by changing its regular RGB camera for an RGB-D sensor camera, which allows it to detect the depth of its environment. This, along with its existing 2D laser range-finder (Hokuyo) sensor, allows the robot ROS package slam-bot to leverage the rtabmap-ros package to perform SLAM (See Fig. 3). Since manual mapping is to be performed the slam-bot package communicates with a teleop package that allows the user to move and steer the robot.

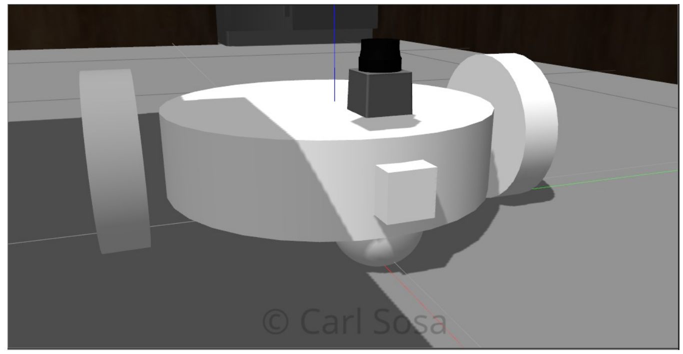

Fig. 3. Slam-Bot Visual

The backend configuration of the robot can be better appreciated in the Transform tree (See Fig. 4).

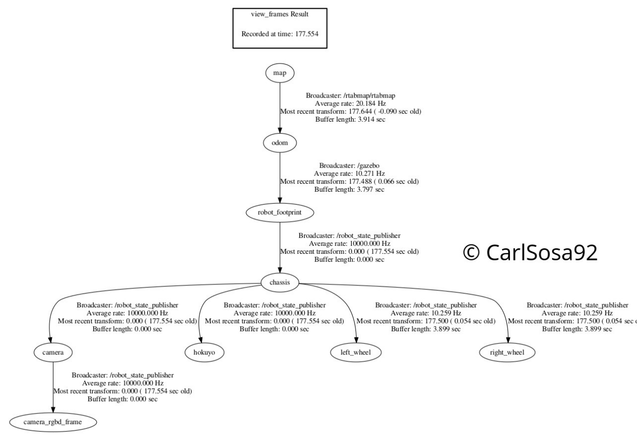

Fig. 4. Udacity Bot Run.

4 RESULTS

4.1 Kitchen and Dining Scene

Upon launching the Kitchen and Dining scene simulation, and launching the mapping node, the slam-bot started mapping the environment in its immediate vicinity (See Fig. 5).

After traversing and navigating around the entire world, the slam-bot was able to generate an accurate map of the world that was provided (See Fig. 6).

As can be seen from the rtabmap database, several loop closures were observed, and bot a 2D occupancy grid map and a 3D octomap were generated for the provided environment (See Fig. 7).

4.2 My World Scene

Upon launching the My World scene simulation along with the mapping node the slam-bot started mapping the immediate environment (see Fig. 8).

After navigating throughout the entire environment, the slam-bot was able to generate an accurate map of the world that was developed (See Fig. 9).

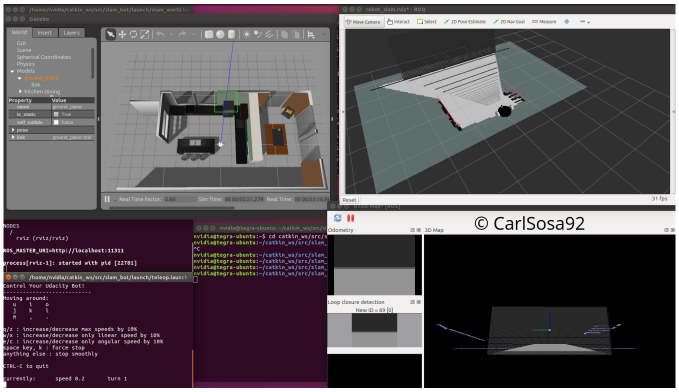

Fig. 5. Kitchen and Dining: SLAM Initialized

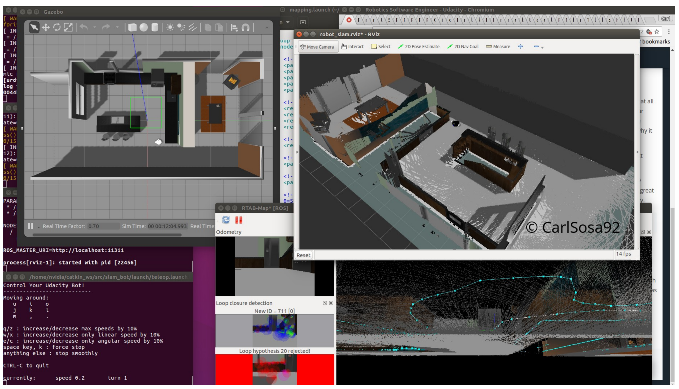

Fig. 6. Kitchen and Dining: Mapped

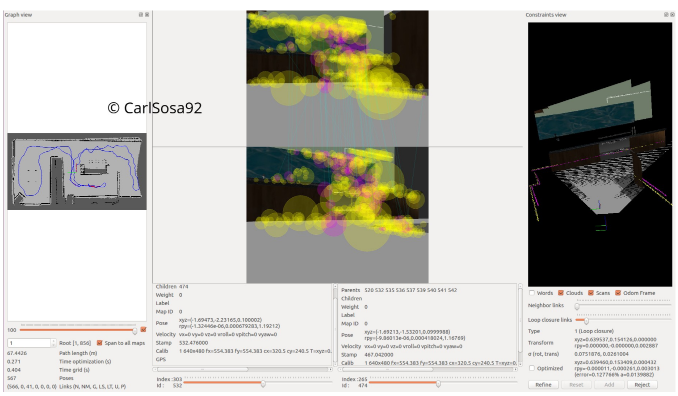

Fig. 7. Kitchen and Dining: Rtabmap Database

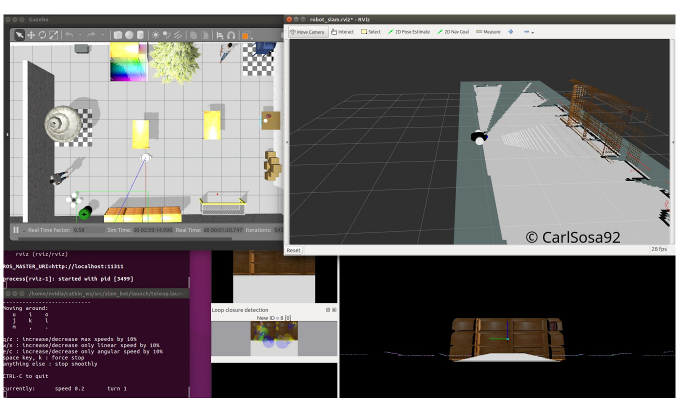

Fig. 8. My World: SLAM Initialized

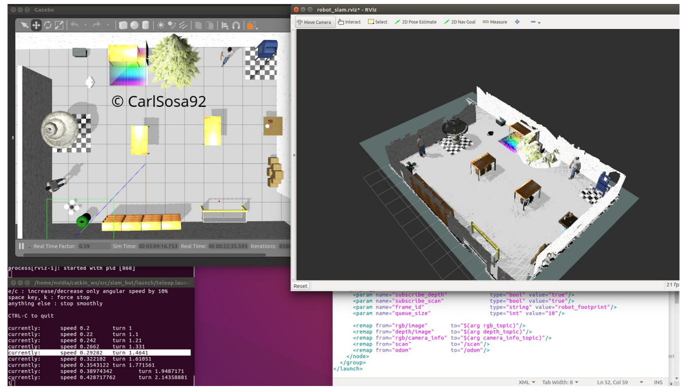

Fig. 9. My World: Mapped

As can be seen from the rtabmap database, several loop closures were observed, and bot a 2D occupancy grid map and a 3D octomap were generated for the custom developed environment (See Fig. 10).

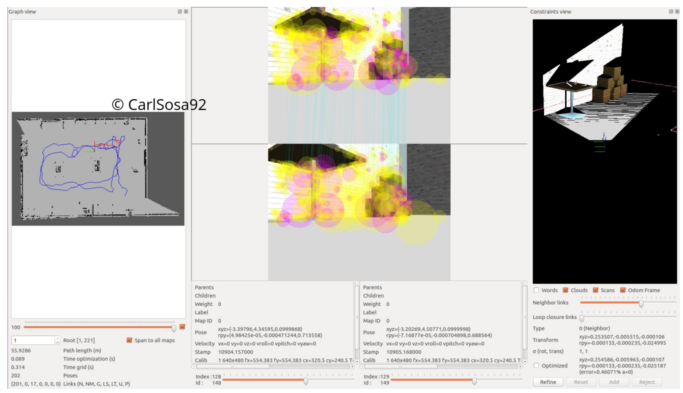

Fig. 10. My World: Rtabmap Database

5 Discussion

Since it was a manual navigation, traversing through the two environment was rather tedious. Furthermore, making a close approaches to obstacles threw off the mapping process at times, and was difficult for the robot to recover afterwards. Although the Kitchen and Dining scene seemed to have less noticeable features that the My World scene, it was less complicated and faster to map that the latter. In order to map the My World scene, navigating around the objects that were placed to serve as both obstacles and features had to be done cautiously. Otherwise, if the robot got too close to objects, it would delocalize the robot, throwing off the mapping process. Whereas not getting close enough, would not allow the robot to map the entire My World scene accurately. One thing that was observed was that the laser range-finder had a lower range than necessary for the designed environment. While the Kitchen and Dining scene was more constrained, with solid objects and obstacles, the range-finder had solid surfaces to bounce off to detect distance accurately. However, the My World scene has more open space which seemed to have made the mapping slightly more complicated for the robot. Nonetheless, the robot was able to successfully map both environments accurately, solving the SLAM problem in the process.

6 FUTURE WORK

Possible future work includes developing a 4-wheeled robot equipped with the Jetson TX2 and a kinect RGBD camera sensor to map an entire real-life apartment. Possibly this would lead to building a custom Roomba robot to vacuum or somehow clean an apartment. With some improvements, this approach can be implemented in a drone for real estate aerial mapping; among other solutions that can be deployed for aerial or ground robotics.