Robotics Project : SLAM

Shotaro Oyama

Abstract—This project implements SLAM (Simultaneous localization and mapping) with RTAB-Map (Real-Time Appearance-Based Mapping) approach. We create a 2D occupancy grid and 3D octomap from a provided simulated environment, then, we create our own simulated environment to map as well.

Index Terms—ROS, RTAB-MAP, Robotics Nanodegree, Udacity

1 Introduction

SLAM is abbreviation of Simultaneous Localization And Mapping. It is the problem of constructing a map of an unknown environment while simultaneously localizing a robot relative to this map. The robot is not aware of the map or position of the robot in advance and try to get the information from a set of noisy controls and observations. In this report, we have 2 environment. No.1, kitchen dining, which is provided by the course material. No.2, the original environment, which is created by using gazebo. These 2 environments were successfully mapped by the robot model which had RGB-D camera and Hokuyo Lidar and rtab-map library.

2 Background / Formulation

SLAM is fundamental problem in mobile robotics. There are many applications, on the robot, where there isn't a known map, either because the area is unexplored or because the surroundings change often and the map may not be up to date. The robot must estimate a map of its environment while simultaneously localizing itself to the map. Up to now, the two most useful approaches to SLAM are Grid-based FastSLAM and GraphSLAM.

2.1 Grid-based FastSLAM

The FastSLAM algorithm uses a custom particle filter approach to solve the Full SLAM problem with known correspondences. Using particles, FastSLAM estimates a posterior over the robot path along with the map. Each of these particles holds the robot trajectory which will give the advantage to SLAM to solve the problem of mapping with known poses. In addition to the robot trajectory, each particle holds a map and each feature of the map is represented by a local Gaussian. FastSLAM has a big disadvantage since it must always assume that there are known landmark positions, and thus with FastSLAM we are not able to model an arbitrary environment. Grid-based FastSLAM extends the FastSLAM algorithm and solve the SLAM problem in term of grid maps, thus you can now solve the SLAM problem in an arbitrary environment. Specifically, Grid-based FastSLAM algorithm estimates the robot trajectory using the MCL. Then, the Grid-based FastSLAM algorithm estimates the map by assuming known poses and using the occupancy grid map- ping algorithm. The basic steps of Grid-based FastSLAM are as follows:

1) Previous belief
2) Sampling motion
3) Importance weight
4) Map estimation
5) Resampling
6) New Belief

2.2 GraphSLAM

GraphSLAM uses a graph to represent the problem, it involves to construct a graph whose nodes represent robot poses or landmarks and in which an edge between two nodes encodes a sensor measurement that constrains the connected poses. The goal of GraphSLAM is to find a configuration of the nodes that minimize the error introduced by the constraints. RTAB-Map (Real Time Appearance Based Mapping) is a RGB-D Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. In this report, RTAB-Map is used as SLAM algorithm.

3 SCENE AND ROBOT CONFIGURATION

3.1 Scene

The environment is as shown in Fig.1 and Fig.2. The Fig.1 is provided environment which models kitchen dining at home, and the fig.2 is newly designed environment which models very simple room, square type, coffee table and cabinet.

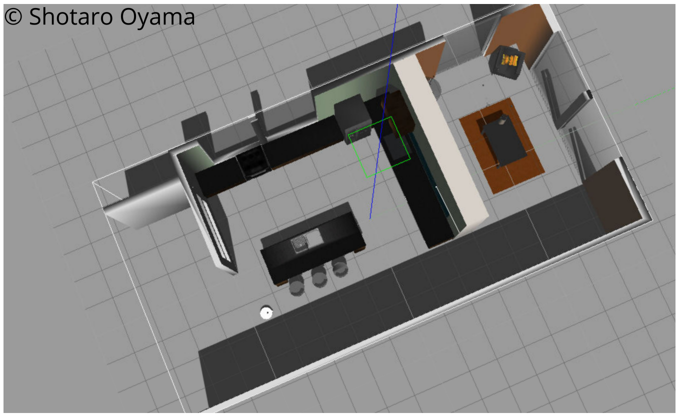
3.2 Robot Configuration

The robot is composed of a 2D laser, a RGB-D camera, wheel encoders, a chassis, two differential wheels, and one caster. The picture is shown in Fig. 3. The backend process which combine the information from sensors and transformed by algorithms is as shown in Fig. 4.

4 Results

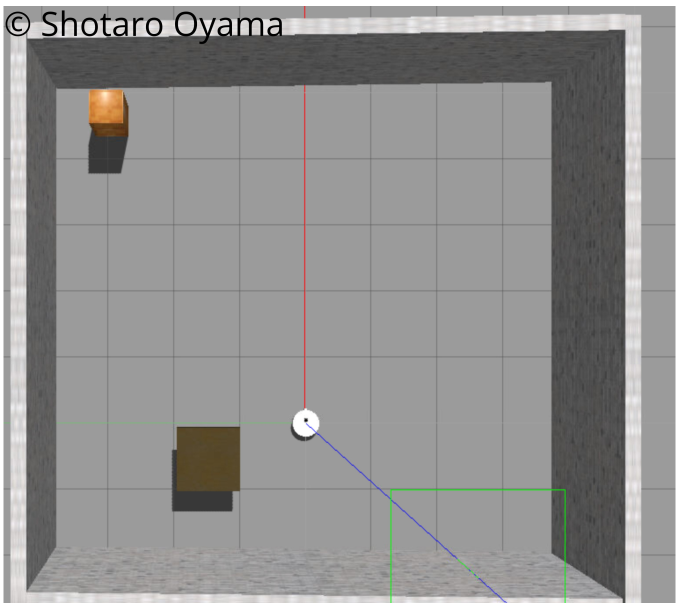
4.1 Designed Environment

On the scene simulation, while launching the mapping node, the robot started mapping the environment immediately and after traversing, it looks that it did well. (Fig. 6). From the rtabmap database, several loop closures were observed, a 2D occupancy grid map and a 3D octomap were generated for the provided environment (Fig. 5 and 7).



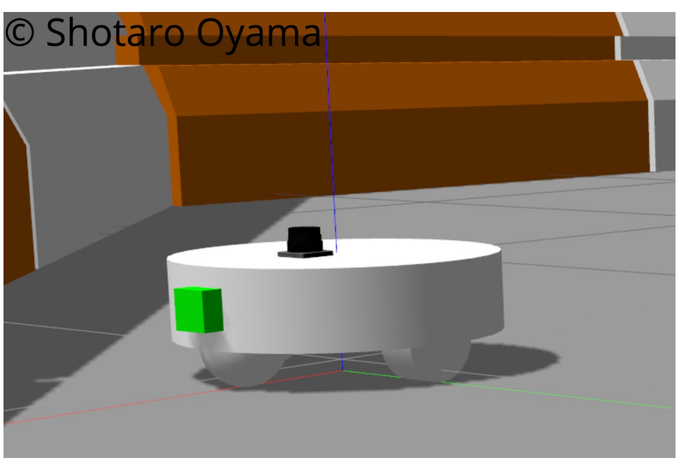
Kitchen_Dining_World

Fig. 1. Provided Environment



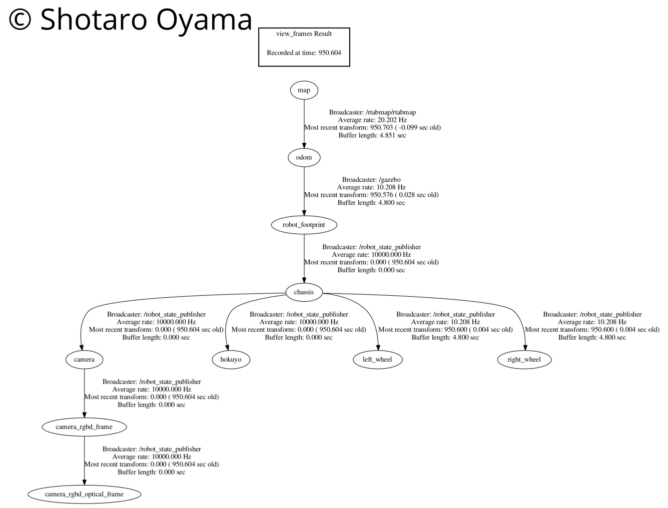
My_World

Fig. 2. Designed Environment



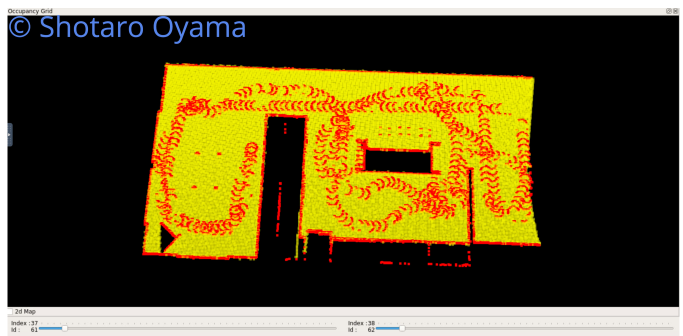
My_Robot

Fig. 3. Designed Robot



frames

Fig. 4. frames



occupancy grid 2d map for Kitchen_Dining_World

Fig. 5. occupancy grid 2d map

4.2 Provided Environment

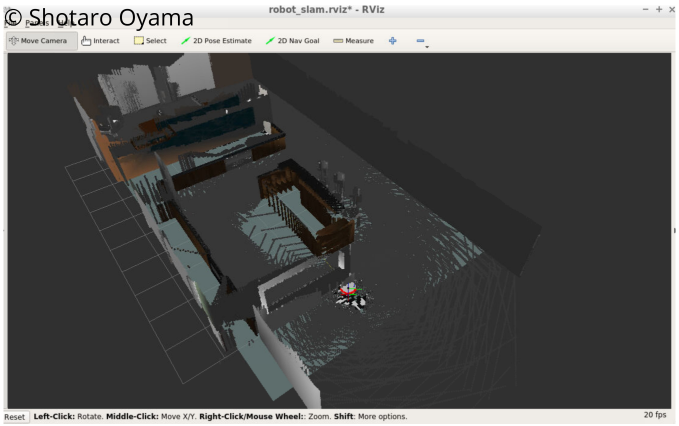
Likewise, after traversing robot could map the environment well. (Fig. 8 and 9)

5 Discussion

The robot had more difficulty to make precise map in designed environment than provided environment. It has less clue in the designed environment because it is very simple, and detected loop closure many times. Finally, it was improved by tweaking the parameter of RTABMAP, as increasing Minimum visual inliers to accept loop closure. In the provided environment, the parameter did not affect the performance of SLAM.

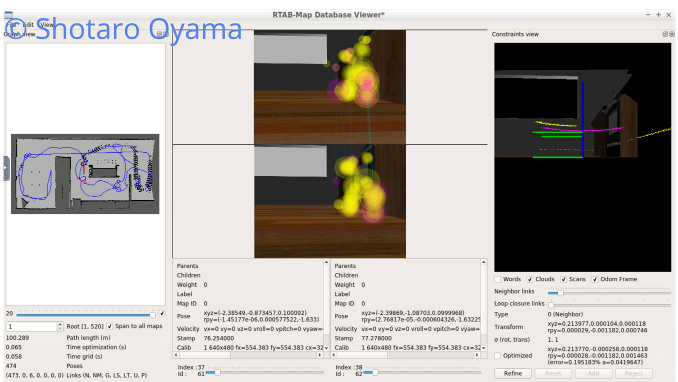
6 Future Work

From the result of simulation, it looks that the parameters need to be adjusted to the environment in order to build precise map, but it should be tedious if there are many environments. To make it easier, I would like to try to find the better combination of sensors, and parameters which can be used in wider variety of situation with better performance.



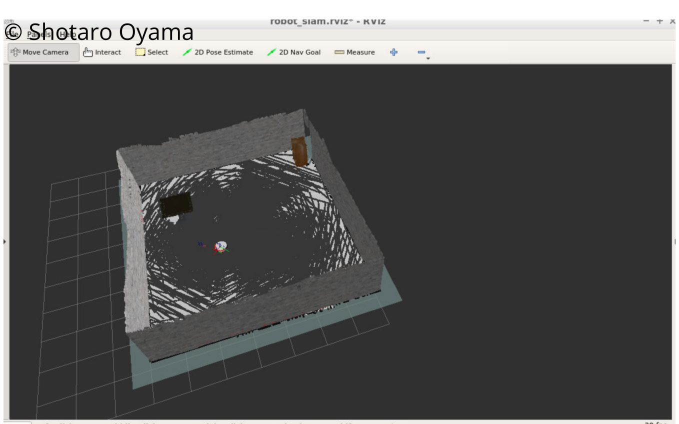
SLAM for Kitchen_Dining_World

Fig. 6. SLAM IMAGE Kitchen_Dining



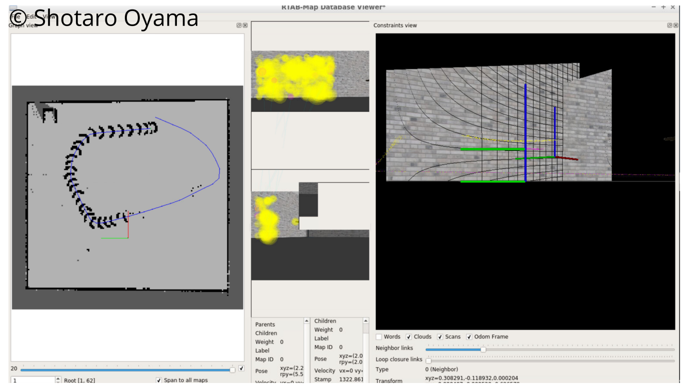
RTABMAP for Kitchen_Dining_World

Fig. 7. RTABMAP for kitchen dining



SLAM for My_World

Fig. 8. SLAM IMAGE my_world



RTABMAP for My_World

Fig. 9. RTABMAP for my world

This technology can be used in self-driving cars, drones, and so on as well as the vacuuming robot in this report. Especially, the situation where several type of robots and human working together will require elaborate map and to be update in real-time, to avoid dangerous accident.