Robot Mapping and SLAM: Map My World Robot

Naruhiko Nakanishi

Abstract—A 2D occupancy grid and 3D octomap are created from a provided simulated environment. Furthermore, the own simulated environment to map is created as well. RTAB-Map, which is the best solution for SLAM to develop robots that can map environments in 3D, is used. The mobile robot model with a RGBD camera and a laser sensor is created.

Index Terms—Robot, Mapping, SLAM, GraphSLAM, RTAB-Map.

1 INTRODUCTION

IN Simultaneous Localization and Mapping (SLAM), a robot constructs a map of the environment, while simultaneously localizing itself relative to the map. The problem is challenging because neither the map nor the robots poses are provided.

RTAB-MAP (real time appearance based mappimg) is a GraphSLAM implementation. For this project, the rtabmap-ros package, which is a ROS wrapper (API) for interacting with rtabmap, is used. Using the created ROS package, both a 2D and 3D map of the supplied environment are created. From there then the own environment is built and a 2D and 3D map of that environment are generated. This project is completed on Jetson TX2.

2 BACKGROUND

SLAM algorithms generally fall into five categories.

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

2.1 FastSLAM

FastSLAM algorithm uses the particle filter approach along with the low dimensional extended kalman filter to solve the SLAM problem. This algorithm is adapted to grid maps, which results in a grid based FastSLAM algorithm.

2.2 GraphSLAM

GraphSLAM uses constraints to represent relationships between robot poses and the environment, and then tries to resolve all of these constraints to create the most likely map given the data.

RTAB-MAP, which is a GraphSLAM implementation, is used in this project.

After much testing and research, RTAB-Map is found to be the best solution for SLAM to develop robots that can map environments in 3D. These considerations come from RTAB-Map's speed and memory management, its custom developed tools for information analysis and, most importantly, the quality of the documentation. [1]

3 SCENE AND ROBOT CONFIGURATION

The Gazebo and Rviz environments are setup and the robot is launched inside the environment. The appropriate launch files are created.

3.1 Provided Simulated Environment

A provided Gazebo world, that is called Kitchen Dining world, is built. The world is correctly created and loaded.

Fig.1 shows Kitchen Dining world.

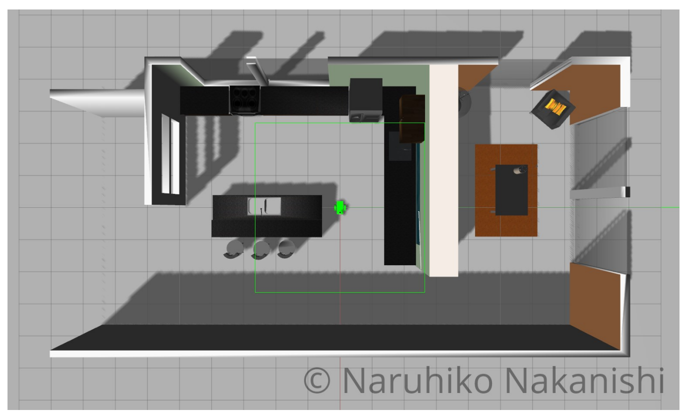

Fig. 1. Kitchen Dining world

3.2 Own Simulated Environment

A personal Gazebo world, that is called Naru Cafe world, is built. The world is correctly created and loaded.

Fig.2 shows Naru Cafe world.

3.3 Robot Configuration

A mobile robot model, that is called NaruBot, is created. The robot model is a mobile robot comprising of a base, two actuators, a RGBD camera, and a laser sensor. The robot is controlled by keyboard.

Fig.3 shows NaruBot. The robot could navigate and map the Gazebo worlds using RTAB-Map.

Fig.4 shows transform tree for NaruBot. The tf-tree makes sure that all of the links are in proper order.

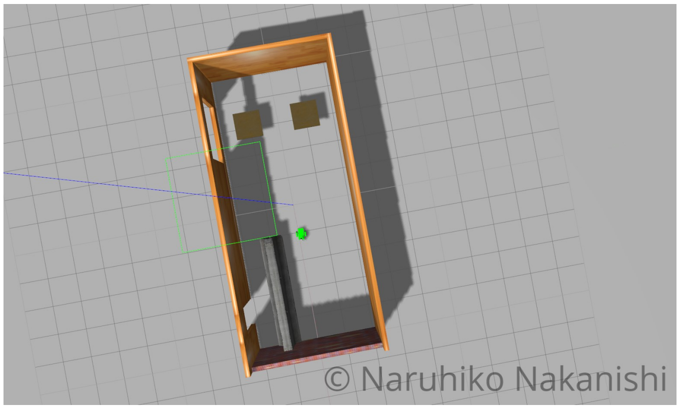

Fig. 2. Naru Cafe world

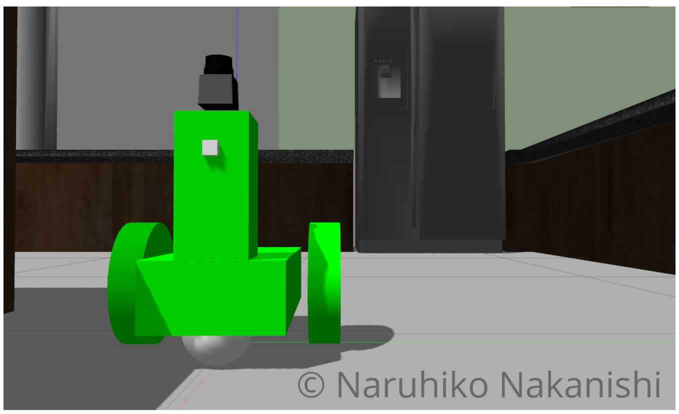

Fig. 3. NaruBot

4 Results

4.1 Provided Simulated Environment

Fig.5 shows the results of 2D occupancy grid for Kitchen Dining Gazebo world.

Fig.6 and Fig.7 show the results of 3D map for Kitchen Dining Gazebo world. Overall 3D map portrays room characteristics and the 3D map of the environment is clearly recognizable.

In Fig.7, rtabmapviz is used to deploy on a real robot during live mapping to ensure that the necessary features are gotten to complete loop closures.

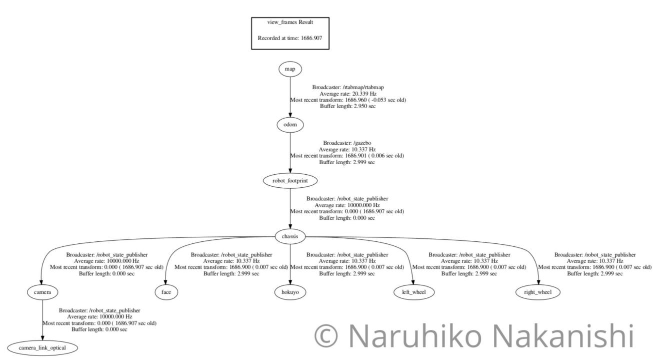

Fig. 4. transform tree

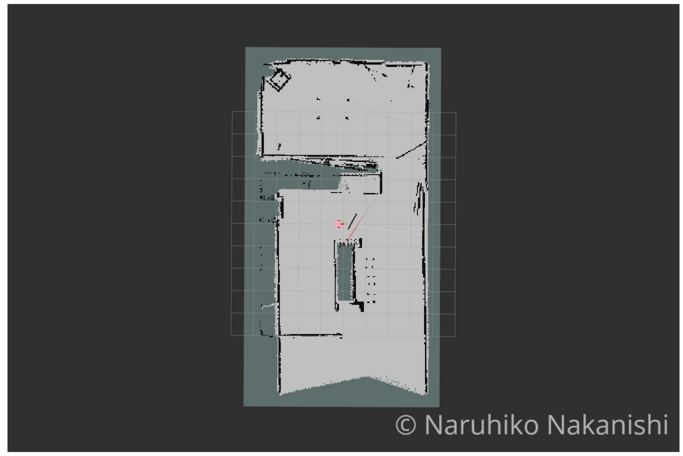

Fig. 5. 2D occupancy grid for Kitchen Dining Gazebo world

Fig. 6. 3D map for Kitchen Dining Gazebo world

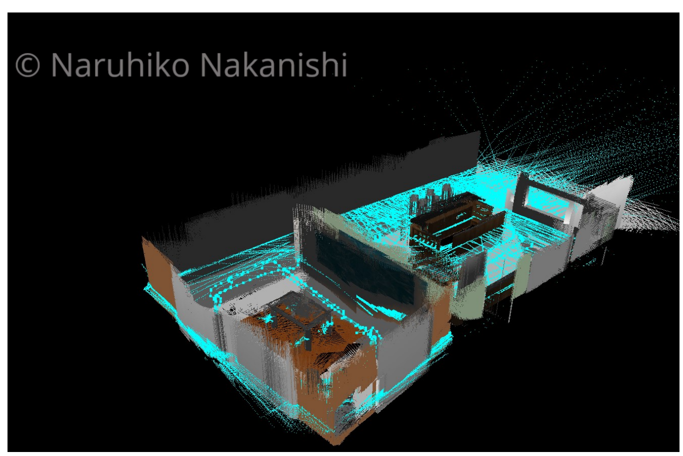

Fig. 7. 3D map with loop closures for Kitchen Dining Gazebo world

4.2 Own Simulated Environment

Fig.8 shows the results of 2D occupancy grid for Naru Cafe Gazebo world. It is 2D grid map in all of its updated iterations and the path of the robot.

Fig.9 and Fig.10 show the results of 3D map for Naru Cafe Gazebo world. Overall 3D map portrays room characteristics and the 3D map of the environment is clearly recognizable.

In Fig.10, rtabmapviz is used to deploy on a real robot during live mapping to ensure that the necessary features are gotten to complete loop closures.

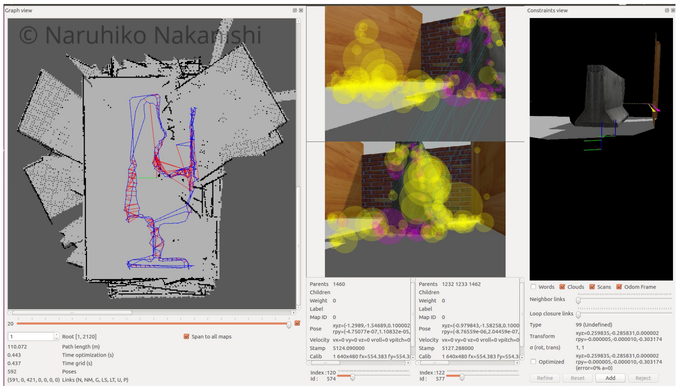

Fig. 8. 2D occupancy grid for Naru Cafe Gazebo world

Fig. 9. 3D map for Naru Cafe Gazebo world

5 Discussion

The goal is to create a great map with the least amount of passes as possible. The loop closures would be maximized by going over similar paths three times. This allows for the maximization of feature detection.

The own robot can map in both worlds accurately, and the SLAM problem is solved in the process.

The robot is controlled by keyboard, not autonomously. So the performance of mapping in both worlds goes well in case that the robot is controlled without collisions.

Compared with Kitchen Dining world, naru Cafe world is simple. Then RTAB Mapping in Naru Cafe world seems more complicated for the robot.

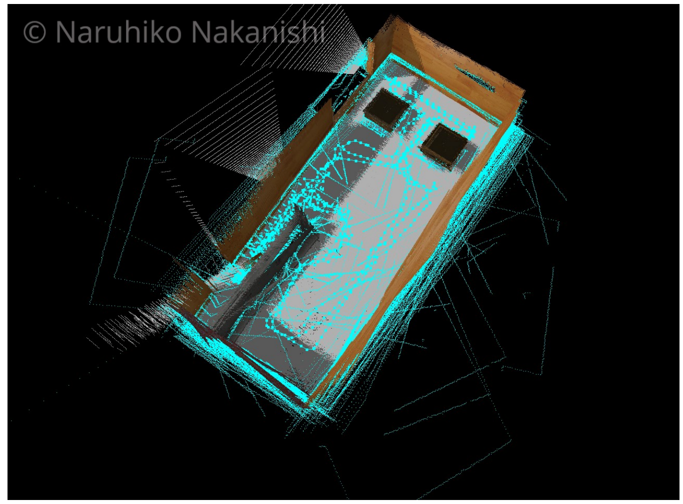

Fig. 10. 3D map with loop closures for Naru Cafe Gazebo world

6 Conclusion / Future work

A 2D occupancy grid and 3D octomap are created from a provided simulated environment. Furthermore, the own simulated environment to map is created as well. RTAB-Map, which is the best solution for SLAM to develop robots that can map environments in 3D, is used. The mobile robot model with a RGBD camera and a laser sensor is created.

In future work, RTAB-Map would be used in a home vacuum cleaning robot or in a drone for aerial robotics. One might argue that it is better to test in simulation first than to make costly mistakes in the real world. Furthermore, RTAB-Map and localization would be combined, and the robots would perform some tasks in an environment such as a warehouse.

References

[1] Udacity, *Robotics Software Engineer Nanodegree Program Term2 Map My World Robot: Project Introduction*. Udacity, 2018.