Map My World Robot

Tan Kai Xiong

Abstract—The objective of this project is to perform SLAM (Simultaneous Localization and Mapping) using GraphSLAM to generate a 2D occupancy grids map and 3D mesh point-clouds of two simulated environment in Gazebo. The custom robot model consists of a chassis, two wheel actuators, a RGB-D camera and a 2D LIDAR sensor(Kinect camera and Hokuyo 2D laser scanner). A kitchen environment and a custom office environment will be used in Gazebo. Using the ROS (Robotic Operating System) packages and RTAB-map (Real-Time Appearance Based Mapping) are being used in this project.

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

1 Introduction

In robotics, an autonomous robot must know the environment in order to move safely and accurately to the destination. SLAM estimate the robot's pose and produce a map of the environment. It able to solve the mapping and localization problem at the same time.

In this project, two different 3D world scenarios are simulated using Gazebo simulator. The two environments shall be mapped in both 2D and 3D by using Gazebo simulation and ROS frameworks.

2 Background

There are two different forms of SLAM. They are online SLAM and full SLAM. The online SLAM solve instantaneous poses independently from previous measurements and controls. The full SLAM need to estimate the entire path instead of an instantaneous pose given all the measurements and controls.

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

The two most commonly used methods in robotics are FastSLAM and GraphSLAM. Both algorithms can solve online and full SLAM problems.

3 FastSLAM

The FastSLAM algorithm solves the full SLAM problem with known correspondences. It estimates a posterior over the trajectory using a particle filter approach. It uses a low dimensional EKF filter to solve independent features of map which are modeled with local guassian.

3.1 FastSLAM 1.0

The FastSLAM 1.0 algorithm is simple and easy to implement, but this algorithm is known to be inefficient since particle filters generate sample inefficiency.

3.2 FastSLAM 2.0

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. Both FastSLAM 1.0 and 2.0 use a low dimensional EKF filter to estimate the posterior over the map features.

3.3 Grid-Based FastSLAM

The Grid-Based FastSLAM is an extension to FastSLAM, which adapts FastSLAM to grid maps. It will update each particle by solving the mapping with known poses problem using the occupancy grid mapping algorithm.

4 GraphSLAM

The GraphSLAM using graph-based SLAM approach to constructs a simplified estimation problem by abstracting the raw sensor measurements. The front-end of GraphSLAM looks at how to construct the graph using the odometry and sensory measurements collected by the robot. The back-end is an optimization process that takes all of the constraints and find the system configuration that produces the smallest error.

4.1 RTAB-Map

The Real-Time Appearance Based Mapping using the GraphSLAM approach. In RTAB mapping, loop closure is detected using the Bag-Of-Words approach. The Bag-Of-Words is commonly used in vision based mapping. The image features are extracted using the SURF (Speed Up Robust Features). RTAB-Map[1] uses a memory management technique to limit the number of locations considered as candidates during loop closure detection. This technique is a key feature of RTAB-Map and allows for loop closure to be done in real time.

5 Robot Model

A custom differential wheel mobile robot is built in URDF file. For this type of wheel drive, it requires two independent motor driven wheels to enable this type of driver to work. It can have either one or two caster wheels to support the robot. It consists of a hokuyo laser scanner, camera and differential wheel sensor.

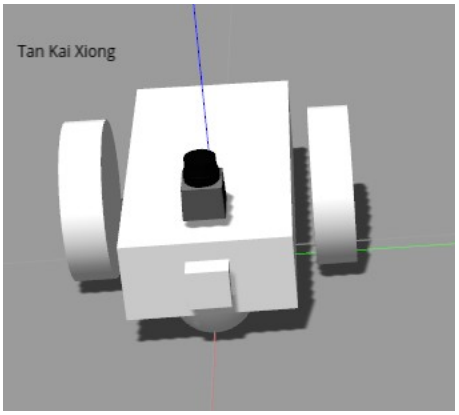

Fig. 1. Robot Model.

6 Gazebo World

Gazebo is a 3D robotics simulator. It also features building editor and model editor.

6.1 Kitchen and Dining

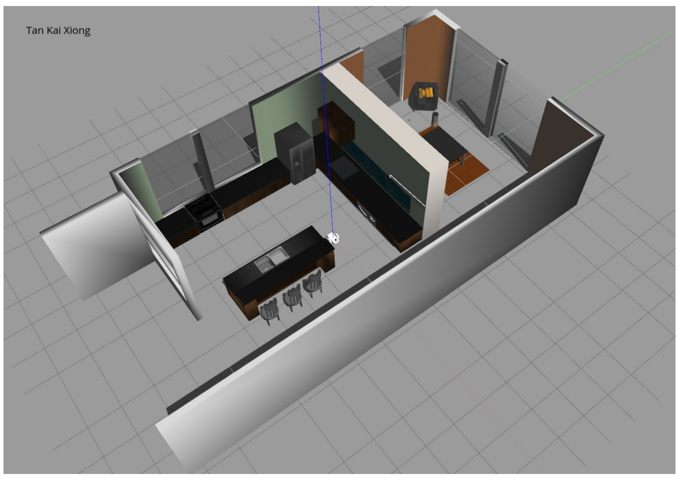

Fig. 2. Kitchen and Dining Gazebo World.

6.2 Custom Office

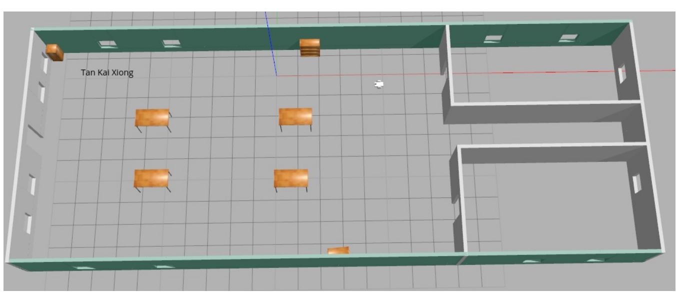

Fig. 3. Custom Office Gazebo World.

7 Result

Overall, both the 2D and 3D map quality looks good.

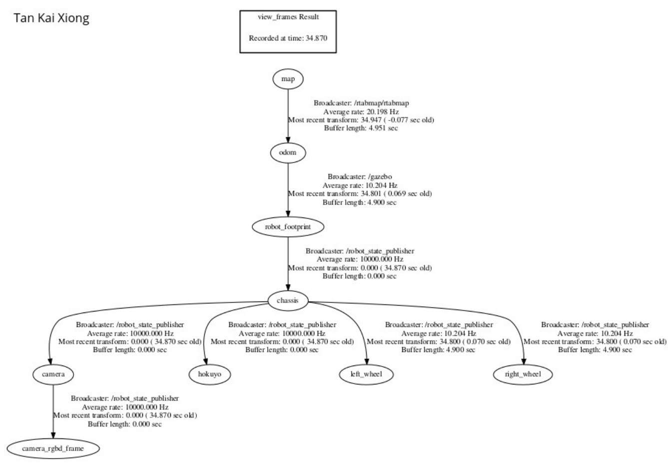

Fig. 4. TF tree.

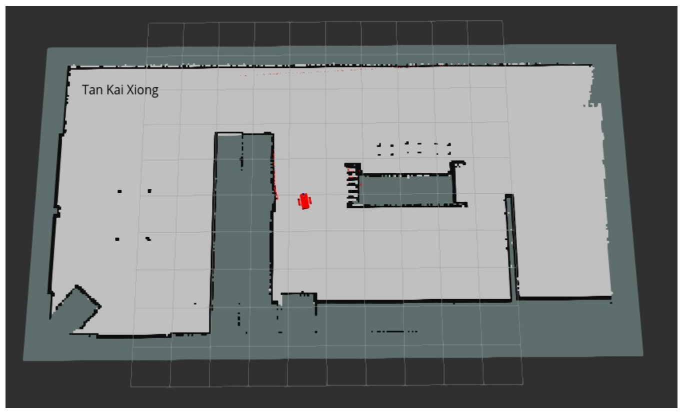

Fig. 5. Kitchen and Dining 2D Grid map.

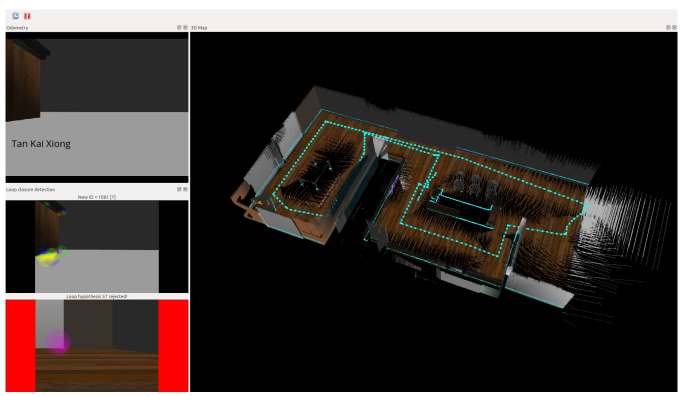

Fig. 6. Kitchen and Dining RTAB-Map.

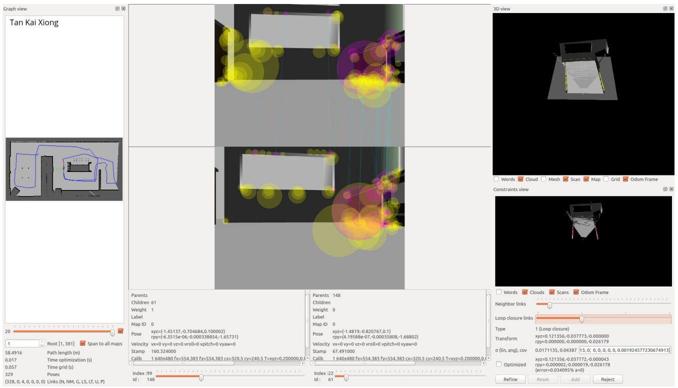

Fig. 7. Kitchen and Dining RTAB-Viewer.

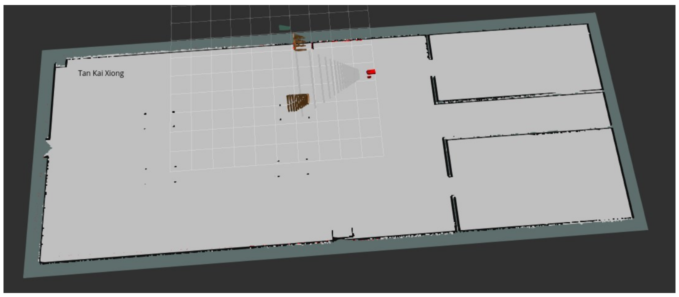

Fig. 8. Custom Office 2D Grid map.

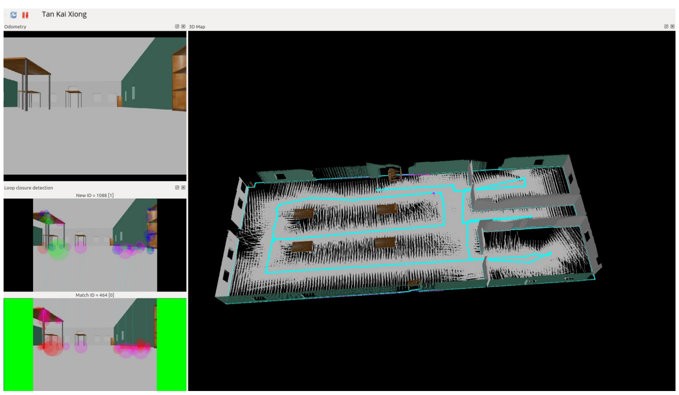

Fig. 9. Custom Office RTAB-Map.

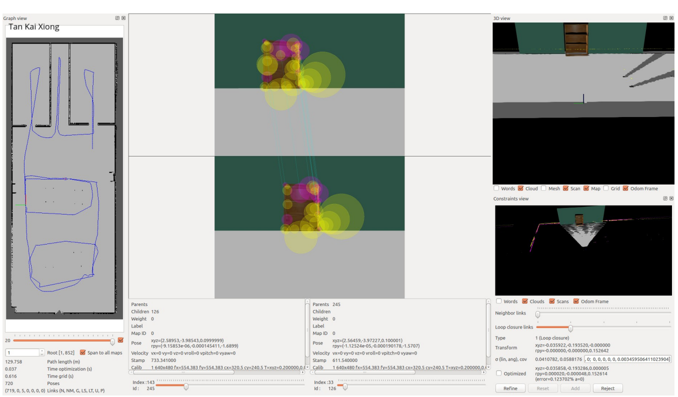

Fig. 10. Custom Office RTAB-Viewer.

8 Discussion

Based on the results, the RTAB-Mapping capable of generating accurate 2D gridmap. There is still some limitation on the RGB-D sensor. The 2D LIDAR able to detect longer distance than RGB-D sensor. The 2D LIDAR can easily map an empty room. In order to generate a full 3D map, the robot have to move nearer to the objects or wall. Based on the two RTAB mapping experiment, the kitchen and dining world have more objects which make it easy to extract more features from the images. It can detect more loop closure. As for the custom office world, it has less objects and the area is big. There is less loop closure detected. In the RTAB mapping experiment, sometime loop closure occured may cause the map to be distorted. It is hard to see the benefit of loop closure in simulation world, as the sensors does not generate any noise.

9 Future work

The RTAB-Map can map a shopping mall, expo hall, school or factory. The industrial cleaning robot can used the map to navigate and clean the area. A differential wheel drive robot equiped with the Jetson TX2, Hokuyo 2d lidar and a kinect camera. The Jetson TX2 do not have sufficient compute power to perform real-time mapping of large area. It will need to log the sensors data first, the processing of the data can be performed on a desktop computer. Map annotation can be performed on the 2D grid-map to mark out the restricted area.

References

[1] M. Labbe, “Ros rtabmap.”