Robotic SLAM Project

Chuong Dao
riochuong@gmail.com

Abstract

Simultaneous Localization and Mapping (SLAM) is a useful technique that helps a mobile robot to learn about the unknown surrounded environment while concurrently localizing itself around. Graph SLAM algorithm is applied to solve full SLAM problem by building a graph of poses and features with constraints related to motion or sensory measurements so an optimized map can be obtained as a result. This paper illustrates an experiment using a specific Graph SLAM technique called Real Time Appearance Based Mapping (RTAB-Map) to build three dimensions (3D) and two dimensions (2D) maps of two different environments with rich features.

Introduction

In this experiment, a robot in simulation environment equipped with both depth camera and laser sensor is used to manually driven through the environments to produce the 3D and 2D map of the environments. The robot software stack runs on Robot Operating System (ROS) which includes the open-source RTAB-Map library for Graph SLAM mapping. The RTAB-Map library provides the functionalities to transform robot current pose and sensors data into visible 3D and 2D maps. There are two different environments required to be mapped. The first environment is provided by Udacity and the second one is created by student using the Gazebo simulation tool. The maps produced from the experiment needs to be closely described the correct properties of the environment.

Background

Mapping is one of the core features that determines the degree of autonomous a robot can achieve. However, mapping is also a very challenging problem because of the huge hypothesis space. Also, mapping is closely related to localization which create a controversial challenge of which should be obtained first. If the poses of robot are known throughout mapping, the problem become easier as we can apply occupancy grid mapping algorithm to produce the map of the environment. Although 2D map can easily be obtained by plotting the result of occupancy mapping algorithm, 3D mapping can be used to precisely model the environment and greatly improve robot ability to navigate and avoid collision. In order to aid the generation of 3D map, robot needs to be equipped with some modern sensors technologies like: 3D lidar, 360-rotated 2D Lidar and RGBD camera. On the other hand, if a map of the environment is provided, localization algorithms such as Monte Carlo (MCL) can be used to navigate to the goal within the environment. SLAM problem as its full name is the combination of challenges from both localization and mapping. The robot needs to be able to estimate the map of the environment while it also need to apply those information to localize itself around that same environment. There are two different type of SLAM problems: online SLAM and full SLAM. While in online SLAM problem we only tries to estimate robot pose and map at a specific point in time, full SLAM needs to solve the posterior over the entire path up to the current time. Moreover, two of the most popular SLAM algorithms are grid-based FastSLAM (GFSLAM) and Graph SLAM (GRSLAM). GFSLAM first solves robot trajectory problem using filter particle in which each particle hold an estimate of the robot poses. GFSLAM then applies the occupancy mapping algorithm to solve each particle mapping problem with known pose. The limitation of GFSLAM is that it relies on particle for pose estimation which can cause some problems if particle is not available at the important location on the map. As a result, GRSLAM has this issue covered with a unique approach to improve accuracy while reducing computation complexity. GRSLAM is divided into two main pieces: the frontend and the backend. The frontend of GRSLAM focus on obtains sensory data to build a graph of constraints among robot poses as well as features of the environment. For example, RTAB-Map algorithm frontend takes in odometry data and images data from RGBD camera and lidar in order to detect loop closure which marks a location as already visited. By applying long-term and window memory management techniques, RTAB-map can support detecting loop closure in real time. On the other hand, The backend of GRSLAM will execute the map optimization algorithm in order to correct the poses of the robot along with all its links transformations. There are various of map optimization algorithms that can be applied such as Tree-based Network Optimizer, General Graph Optimization. As a result, RTAB-Map can produce 2D occupancy grid map, 3D grid map or 3D point cloud.

Scene and Robot Configuration

The custom scene is designed using Gazebo and has rich features so we can manually drive the robot around to map the environment using RTAB-Map library. The custom environment is divided into to two areas (Figure 1). Each area will have multiple unique objects in order to help RTAB-Map easily to detect loop closer as well as a specific corner of the environment.

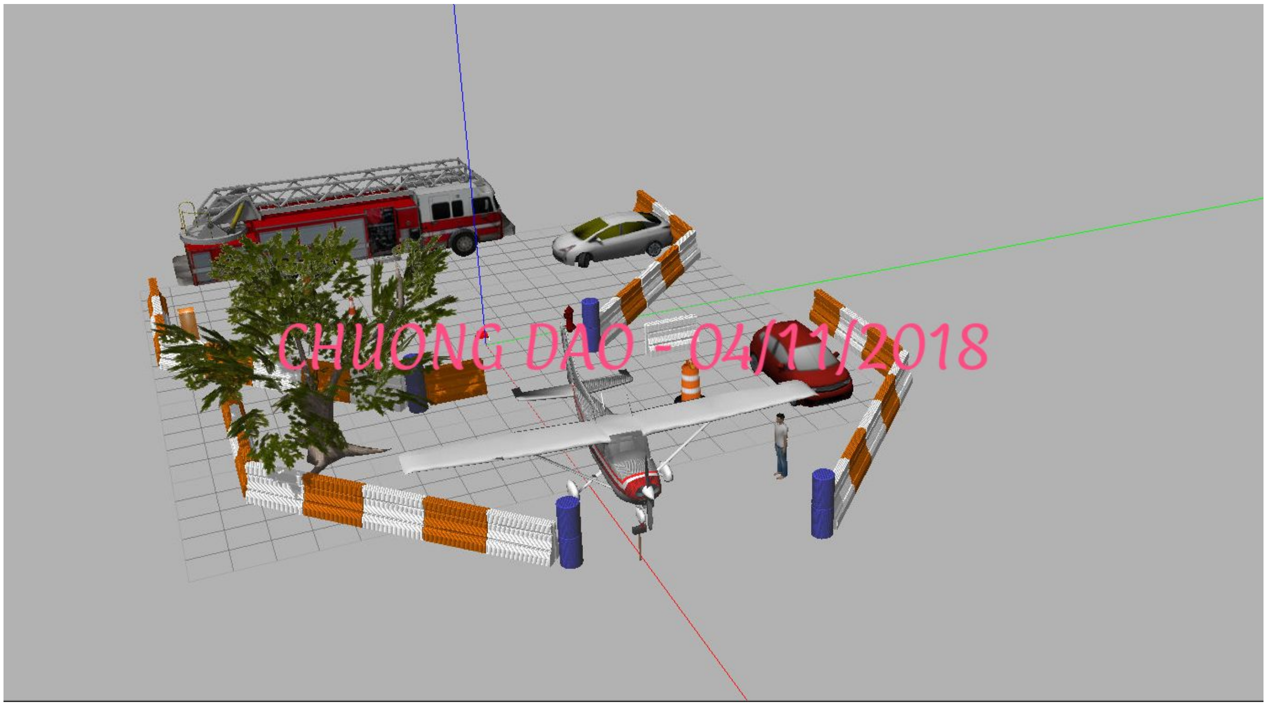


Figure 1: Custom Gazebo World

The robot use for the mapping task is a four-wheel robot equipped with one laser range sensor on top of the chassis and a RGBD camera positioned in the front side of the chassis (Figure 2). The positions of the visionary sensors help the robot to detect specific features in the environment better especially while turning or driving backward to capture broader view of a specific area.

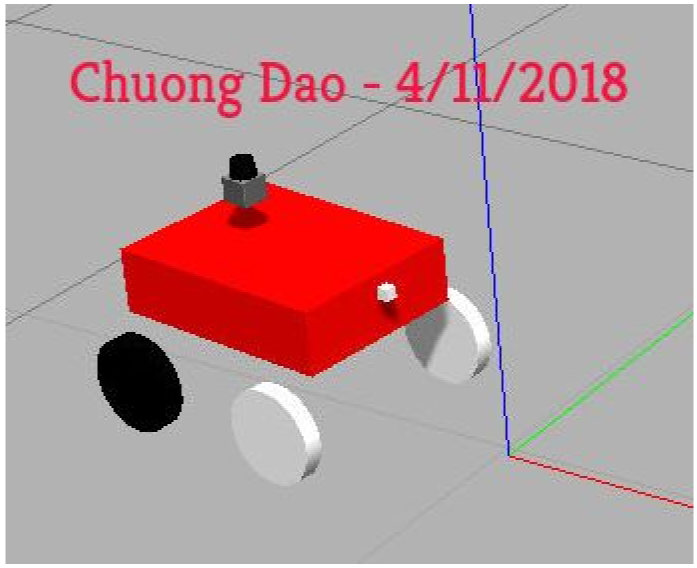


Figure 2: Robot Model with Sensors

Also, the ROS package also need to be configured to take input from odometry, the laser range sensor and RGBD camera and feed it in the inputs of the RTAB-Map library to support detecting loop closure(Figure 3)**.**

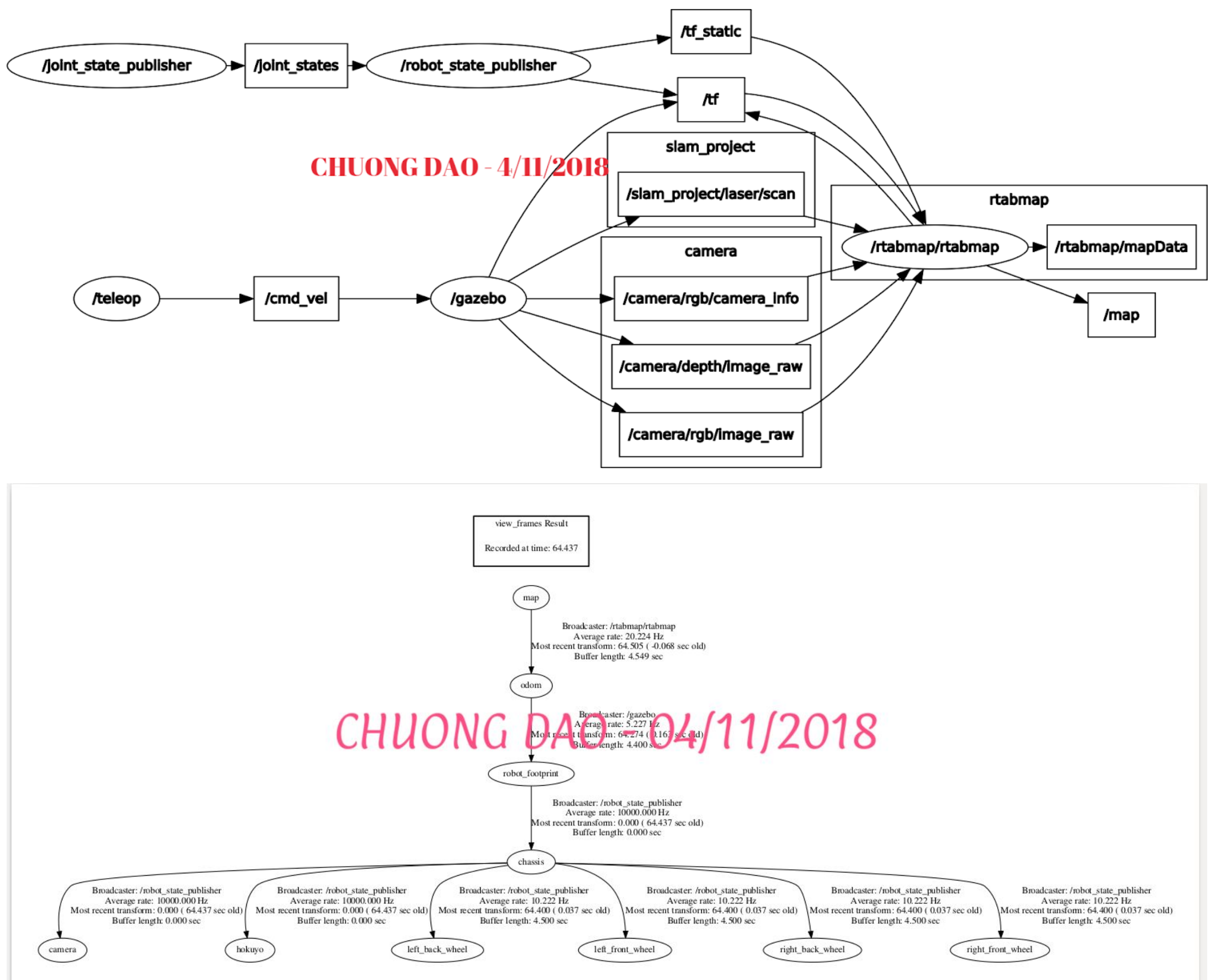


Figure 3: TF transform and ROS Graph of the navigation stack

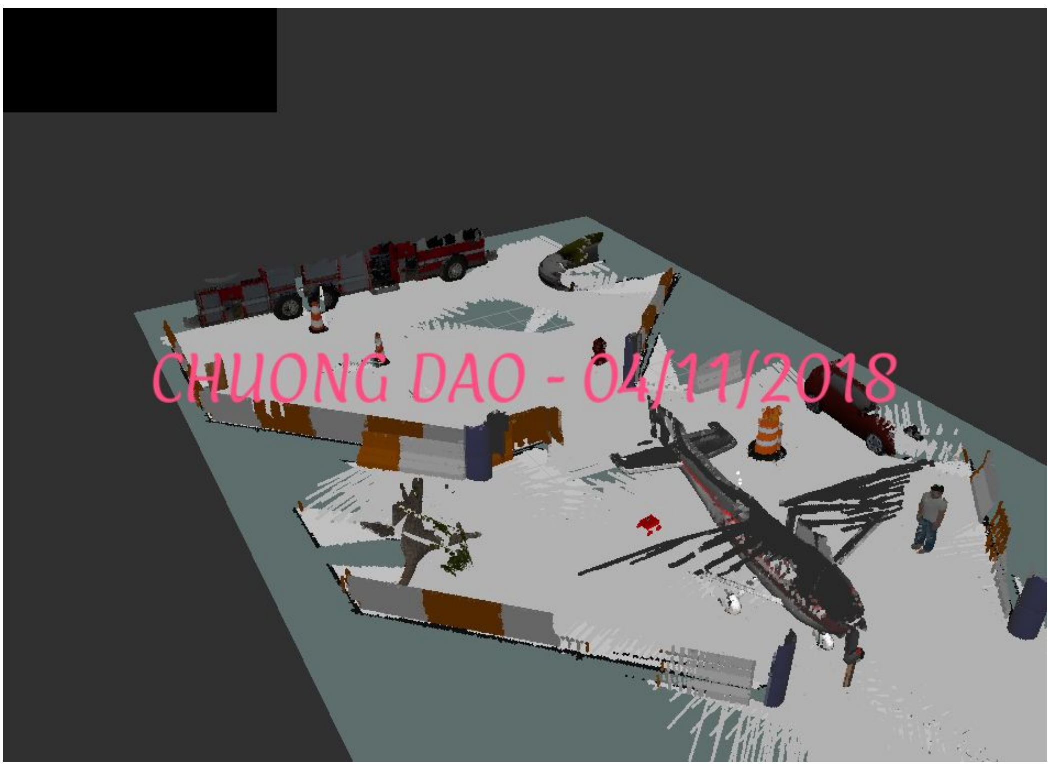
Results and Discussion

The results for the provided world and custom world are shown as in Figure 4 and Figure 5. The provided kitchen world show good results with all the 3D and 2D maps looks very clear and have enough features to describe the original environment. The occupancy grid also looks good as it clearly shows the correct position of the wall and other objects in compared with the original environment. The results on custom world are slightly not up to bar with the kitchen world for couple reasons. Some large objects like airplane and fire truck need the robot to do quite some movements and turns to capture the large portion of them. Also, one area of the custom world is packed with many objects like plane, human, car that leads to RTAB-Map get confused and messing up the map and the whole mapping process have to be restarted again. There are some limitations on the robot model that make it slightly difficult to manual drive around the map, so it is very time consuming to actually complete the mapping tasks. As a result, some of the areas in the custom world is not showed up correctly in the occupancy grid while the outlook of the grid is decent.


CHUONG DAO - 04/11/2018


CHUONG DAO - 04/11/2018

Figure 4: Provided Kitchen World Results



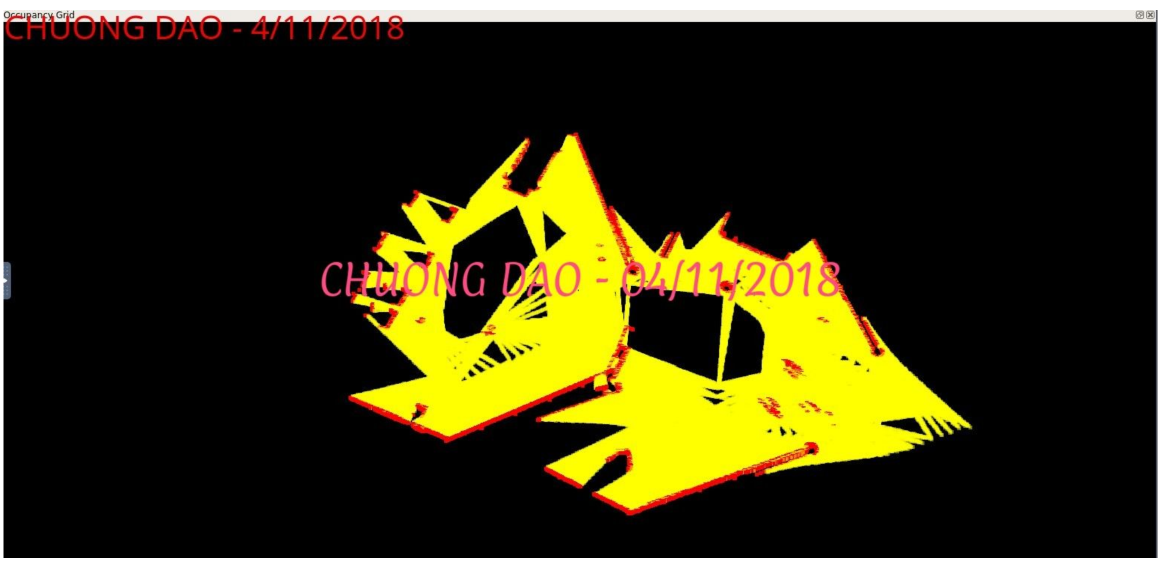


Figure 5: Custom World Results

Future Work

The next experiment is to perform localization using MCL algorithm on the map produced by RTAB-Map. Based on the result of localization task, we can have better evaluation for the quality of the map. In broader view, RTAB-Map and localization can be combined to run on different type of warehouse and delivery robots that perform some repeated tasks in a known environment. It would be fascinating to see how the real robot localize to a goal using the map from RTAB-Map. In the short term, building a small robot with laser range and RGBD sensors to do mapping and navigate around the house with goal can be set from RVIZ running on a laptop is a more practical experiment that can be accomplished.

References

[1] Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Cambridge, Mass: The MIT Press, 2006. Print.