Udacity Robotics Nanodegree Term 2

Map My World Robot

Robert Aleck | github.com/mnbf9rca |21st July 2019

# Abstract

In this paper, the author describes the steps taken to implement the ROS RTAB-Map [1] module in a provided and then in a customised simulation workspace provided by RViz and Gazebo, producing a 2D occupancy grid and 3D octomap of each. Good results were achieved in the provided world; the customised world proved more challenging, requiring the addition of significant modifications to meet the success criteria.

# Introduction

The author adapted a 4-actuator, planar robot previously developed by the author [2] by replacing the monocular front facing camera with a top-mounted Kinect camera, using the ROS openni Kinect package (<http://wiki.ros.org/openni_kinect>). A small block was placed under the laser sensor to raise it up and increase clearance for depth perception above the robot and camera.

A simulated environment, kitchen\_dining.world, was provided. The objective was to spawn the robot in to this world (by convention at the 0,0,0 coordinates, but this location is arbitrary), and navigate the world sufficiently to obtain a 2D occupancy grid, and a visually recognizable 3D map with at least 3 identified (global) loop closures.

The author was then required to generate a second environment and perform the same task.

# Background

## SLAM

As noted in the author’s work submitted for a prior project [2], SLAM is the process of a robot incrementally constructing a worldmap simultaneous to localization within that map, to allow the robot to develop a globally consistent estimate of pose with respect to environment. Accurate pose estimation is a fundamental requirement for autonomous robotics, allowing robots to operate in previously unseen environments (such as with self-driving cars, or unmanned inspection drones for hazardous areas), and providing benefits in tasks like path planning, where priority can be given to previously unsearched areas, or to previously traversed paths therefore assumed to be “safe”.

SLAM is a challenging problem, as the ultimate accuracy of the worldmap depends heavily on the accuracy of localization, and localization requires an accurate world-map, however, several techniques exist to achieve reliable outcomes in reasonable amounts of time, and with reasonable compute power, falling in to five broad categories:

* Extended Kalman Filters (EKF)
* Sparse Extended Information Filters (SEIF)
* Extended Information Form (EIF)
* FastSLAM
* GraphSLAM

SLAM solutions perform either *online* SLAM, or *offline*:

* Online – maintains an estimate over the map and only the present robot pose
* Full SLAM – maintains a posterior over both the map and the entire robot path

## Extended Kalman Filters (EKF-SLAM)

EKF SLAM makes use odometry of feature detection to estimate current robot pose (Online SLAM) in an iterative 5 step cycle:

1. State prediction: given input controls, the current pose of the robot is estimated
2. Measurement prediction: given the best estimate pose prediction, what measurements are expected to be observed
3. Measurement: capture sensor readings
4. Data association: associate measurements to specific landmarks based on measurement proximity to the landmark
5. Update: update the estimated pose given the measurements and measurement predictions.

EKF-SLAM suffers from two key problems [3]:

* Computation time is non-linear ( O(*n*2) for *n* landmarks) with the number of landmarks to be measured, and
* The use of a techniques to linearize inputs leads to propagation of covariance when linearizing an inherently non-linear model (e.g. many real-world systems

## Sparse Extended Information Filters (SEIF-SLAM)

An Extended Information Filter maintains a feature matrix representing the strength of link between features (e.g. measurements over time of a feature relative to the robot’s pose). On observation, it can be seen that most links in the matrix are, once normalized, close to zero, and there are a relatively small number of elements in the matrix with strong links – in short, the normalized matrix is *sparse*.

An online algorithm, SEIF-SLAM exploits this to maintain a sparse information matrix representing only strong links between nearby features. As the matrix is sparse, updates occur in constant (O(*n*)) time [4].

## Extended Information Form (EIF-SLAM)

Although similar to EKF-SLAM, EIF-SLAM is not incremental, calculating a full SLAM posterior on each execution. Because the entire dataset is known at the point of calculation, it’s possible to improve the linearization of the model, and as the EIF-SLAM iterates map construction, it is less susceptible to errors, and produces maps of higher accuracy than EKF. Finally, EIF-SLAM be applied to maps of many orders of magnitude larger than EKF-SLAM [5].

## FastSLAM

## GraphSLAM

# Scene and Robot Configuration

## Robot model

The robot model was based on a previously designed model rob\_bot, created for a prior project submission. The robot was modified to remove the front-facing monocular camera and replace it with a top-mounted front-facing RGBD camea modelled on a Kinect device powered by Openni2 drivers. The hokuyo laser finder was raised by approximately 10cm to ensure the beams were not interrupted by the Kinect camera. The resulting frames are as follows:

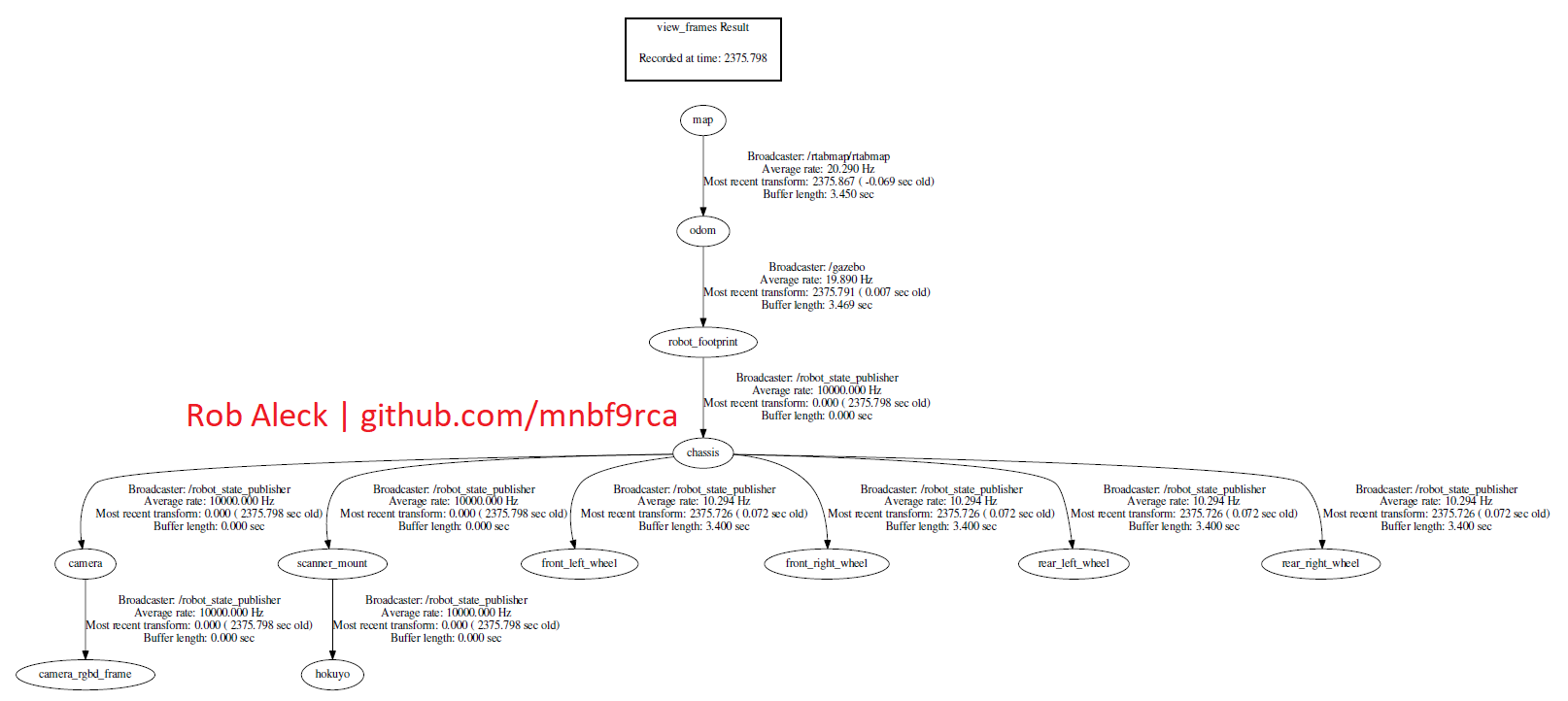


Figure 1 frame map for rob\_bot

## Custom scene generation

A training scene, called kitchen\_dining.world, was provided, and is shown in Figure 2 as seen on initial launch. A custom scene was generated using an out of the box Gazebosim model “cafe". The model was further customised by adding furniture and people. During tuning (see results), it became apparent that the floor texture was highly repetitive across much of the model space, and so additional textures in alternative directions were added to the floor to provide additional visual differentiation for word selection.

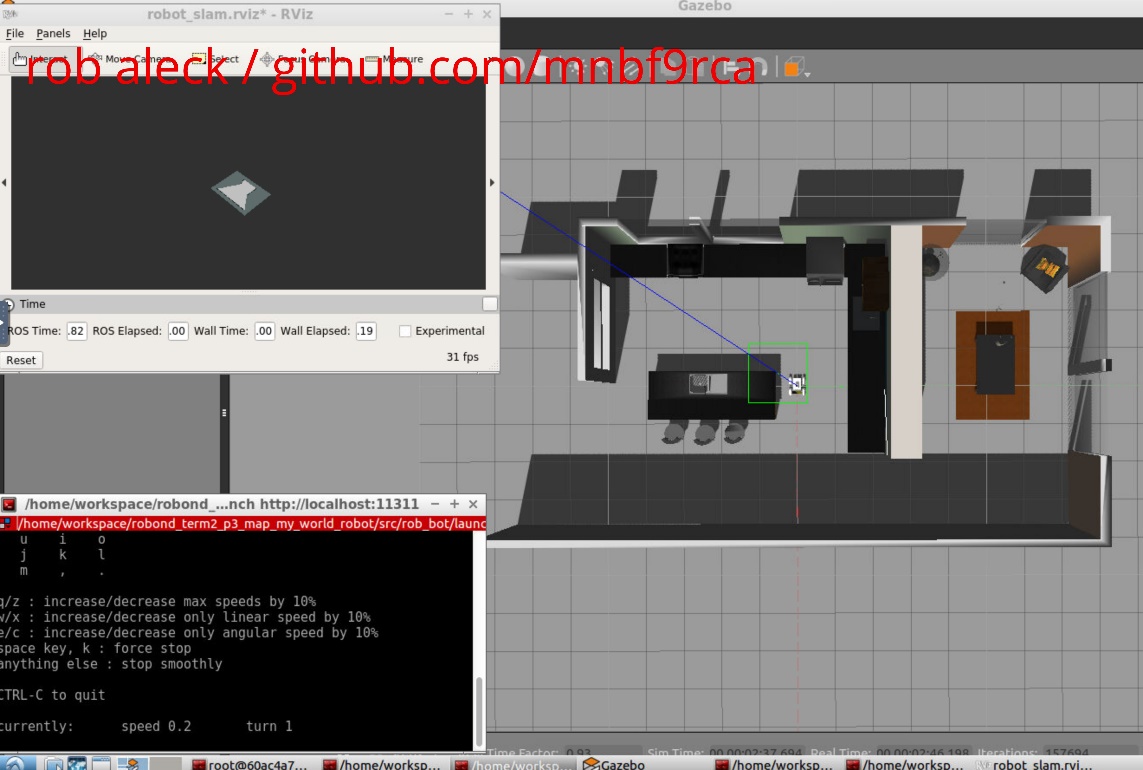


Figure 2 kitchen\_dining.world viewed from above

Scene and robot configuration - Student explains how the gazebo world was created by providing an overview of the layout of items in his/her customized Gazebo world. Student also describes the robot's parameters, sensor features, and reasoning on the package structure.

# Results

## Provided world

A single pass around the room was sufficient to achieve 32 global loop closures across 228 poses, as can be seen in Figure 1.

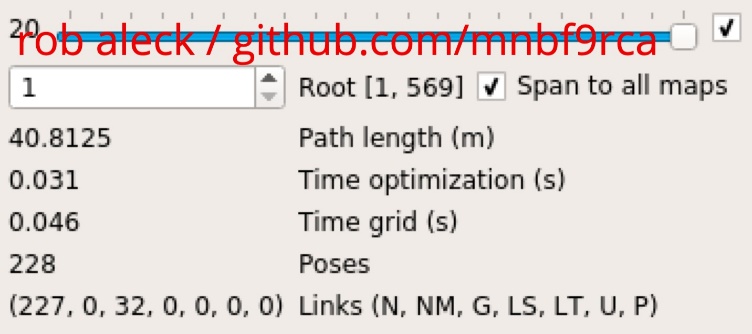
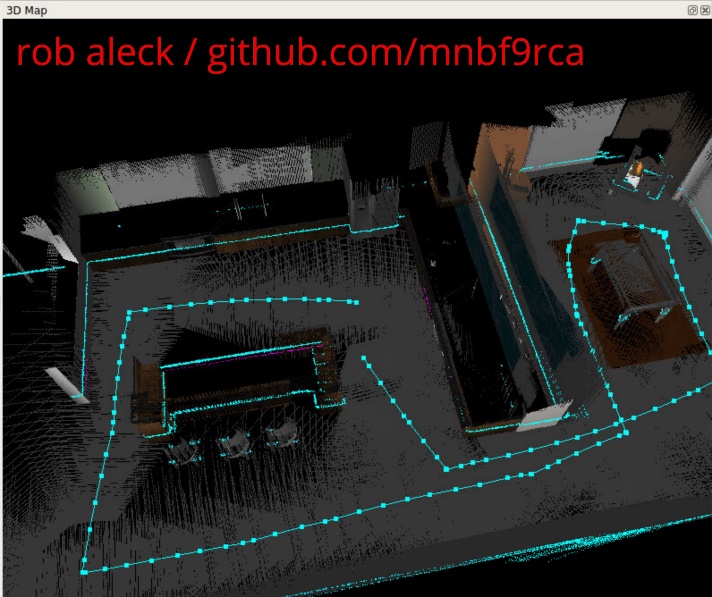


Figure 3 Summary of loop closures from rtabmap-databaseViewer for provided world

The resulting 2D and 3D maps are clearly recognizable.

Results - The student should include the images for mapping process, final map (2D/3D) for both Gazebo worlds.

*Figure 4 3D map generated by ??? for provided world*



# Discussion

Discussion - The student explains how the procedure went and methodologies to improve it. The student should compare and contrast the performance of RTAB Mapping in different worlds.

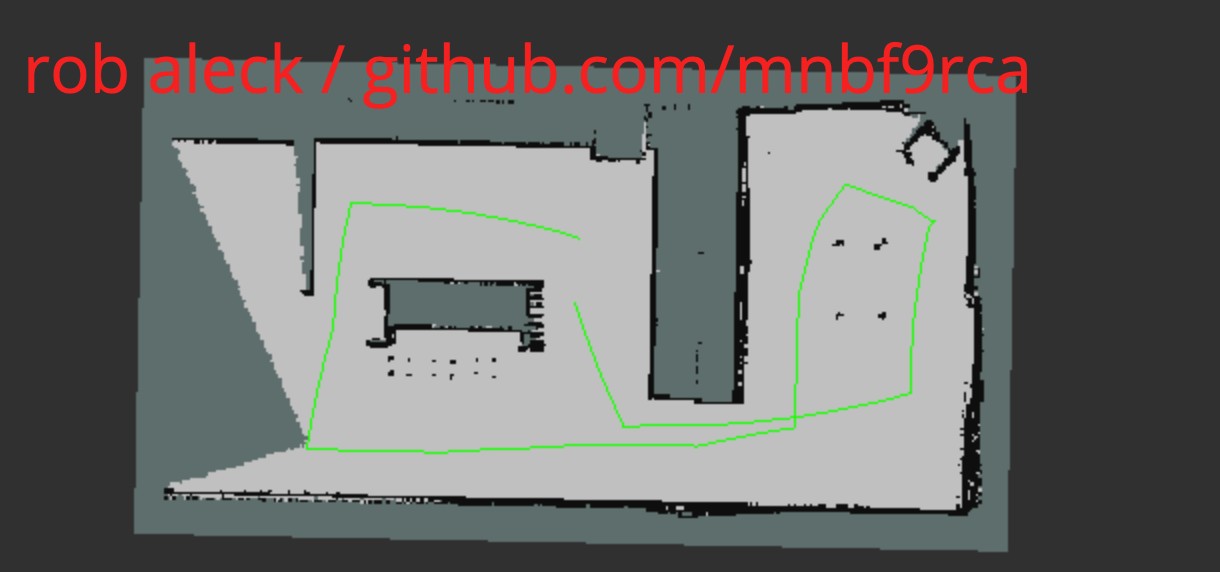


Figure 5 3D map generated by ??? for provided world

# Future Work

"Future Work - The student can discuss how they would like to leverage this tool in robotics. The student identifies other areas where mapping could be done and for what reason. Such as simulated room or physical place.

Try other localization techniques to avoid matching floor

# References

|  |  |
| --- | --- |
| [1] | M. Labbé, “RTAB Map - Real-Time Appearance-Based Mapping,” 2017. [Online]. Available: http://introlab.github.io/rtabmap/. [Accessed 21 07 2017]. |
| [2] | r. aleck, “github/mnbf9rca/robond\_term2\_p2\_where\_am\_i,” 2019. [Online]. Available: https://github.com/mnbf9rca/robond\_term2\_p2\_where\_am\_i. |
| [3] | M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem,” *Proceedings of the AAAI National Conference on Artificial Intelligence.,* pp. 593-598, 2002. |
| [4] | S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *The International Journal of Robotics Research,* vol. 23, no. 7-8, 2004. |
| [5] | S. Thrun, W. Burgard and D. Fox, “11: THE EXTENDED INFORMATION FORM ALGORITHM,” in *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, MIT Press, 2005, pp. 267-268. |