

# Map My World

Jeremy Hale

**Abstract**—The Udacity Robotics Nanodegree project Map My World is a simulation of a small robot mapping two different environments. The robot uses a RGBD camera and a laser scanner with the RTAB-Map method of SLAM (simultaenous localization and mapping) to create a map of the environments. ROS and Gazebo/RViz are the main tools employed.

**Index Terms**—Robot, IEEEtran, Udacity, L<sup>A</sup>T<sub>E</sub>X, Localization.

## 1 INTRODUCTION

THIS project demonstrates the use of SLAM in simulation using ROS and specifically the RTAB-Map implementation of SLAM. SLAM is a technique to solve the problem of localization while building a map from sensor inputs. In this simulation, a small two-wheel robot with RBGD camera and laser scanner is manually navigated around an environment. During navigation, the robot uses its sensor inputs to build a map of the environment.

The problem of mapping applies to many different situations as a robot may commonly find itself in an environment for which it does not have a map a priori. Even if a map were to exist, most environments in the world are not static. In homes and offices, for example, furniture or other significant objects or obstructions may be moved and would differ from prior maps.

## 2 BACKGROUND

Any robot that needs to navigate autonomously needs a map in order to plan a route from the origin to the destination. This map could be a "world" map and the robot could use GPS to navigate, but for any indoor application or an outdoor application where obstructions are likely to move, a map will either not exist or not be sufficiently helpful for navigation. Therefore, the robot must determine the map itself and also determine its location relative to this map.

For the robot in discussion here, a 2D map is sufficient, because our robot moves only in 2D space. However, for airborne robots like drones a 3D map is necessary because they move in three dimensions.

During the classroom work for this project, several varieties of SLAM were discussed such as FastSLAM and GraphSLAM. The algorithm used in this project is RTAB-Map (real-time appearance-based) which is based on Graph-SLAM.

## 3 MODEL CONFIGURATION

The robot was based on the Udacity robot from the "Where am I?" project. It was a small box-shaped chassis with 2 driven-wheels on the sides and 2 stabilizing wheels fore and aft. The robot has a Hokuyo laser scanner and a camera attached on the front. In the previous project, the camera was a simple RGB camera. This was changed to an RGBD camera to satisfy the project requirements.

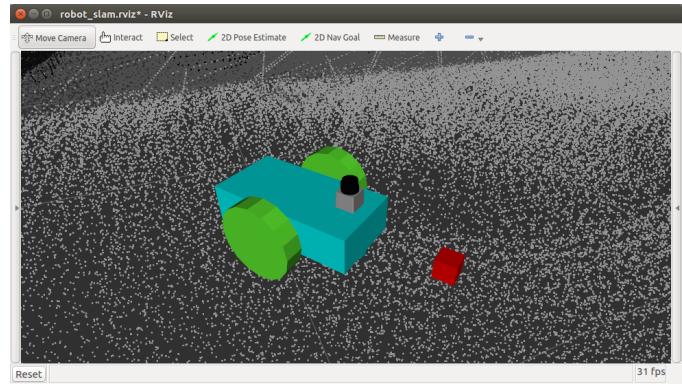


Fig. 1. Robot

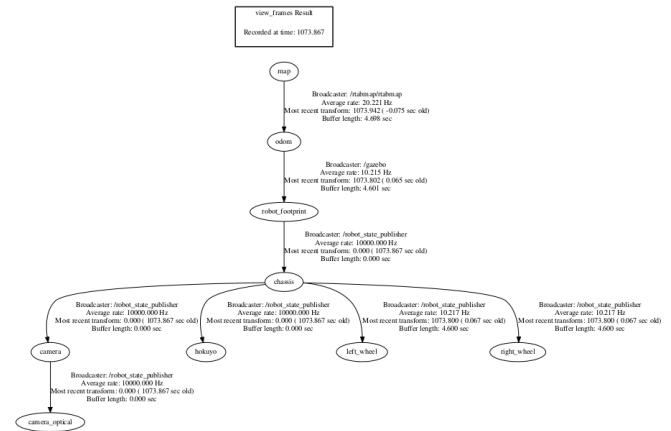


Fig. 2. TF Frames

## 4 WORLD CREATION

The personal "world" is supposed to be a back alley. There is a large brick wall, a pickup truck, a dumpster and a fire hydrant.

## 5 RESULTS

### 5.1 Kitchen & Dining

The occupancy and 3D maps for the "Kitchen and Dining" world are shown below. There are also images showing the 3 closures.

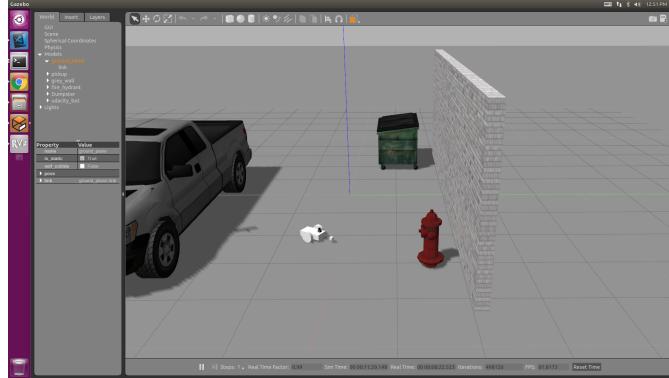


Fig. 3. Alley

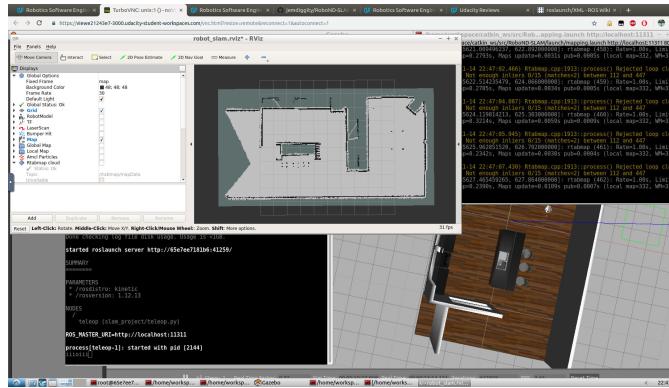


Fig. 4. Kitchen &amp; Dining Occupancy Map

## 5.2 Back Alley

The occupancy and 3D maps for the "Back Alley" world are shown below. There are also images showing the 3 closures.

## 6 DISCUSSION

The mapping performance for the robot in the "Dining and Kitchen" world was satisfactory. Once the robot had retraced its path, the RTAB-Map algorithm detected closures easily with subjectively little distortion.

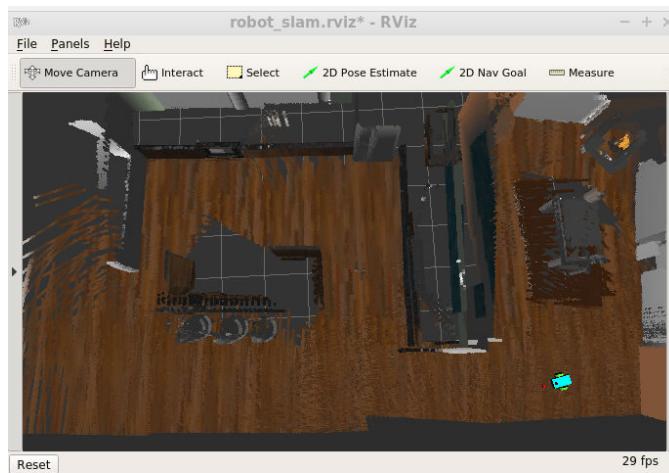


Fig. 5. Kitchen &amp; Dining 3D Map

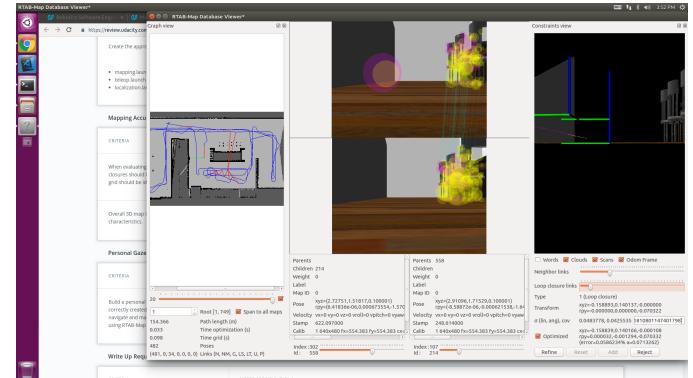


Fig. 6. Udacity World Closure

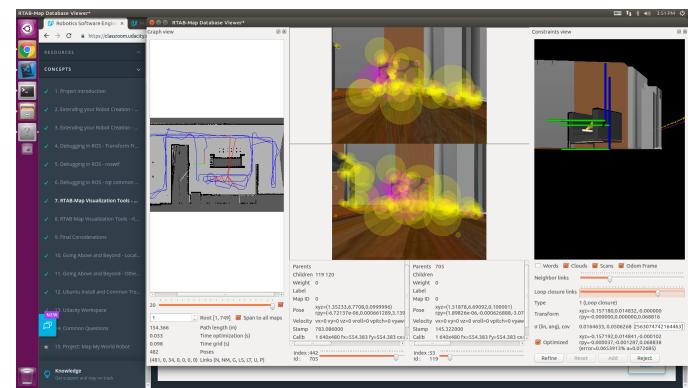


Fig. 7. Udacity World Closure

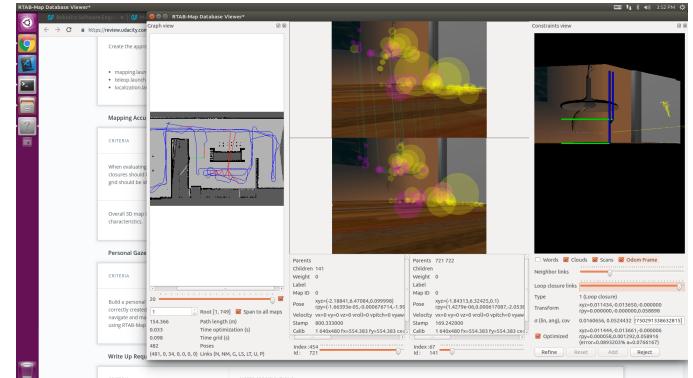


Fig. 8. Udacity World Closure

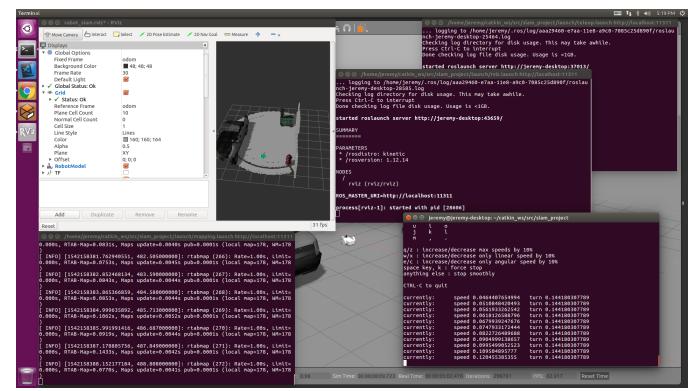


Fig. 9. Back Alley Process

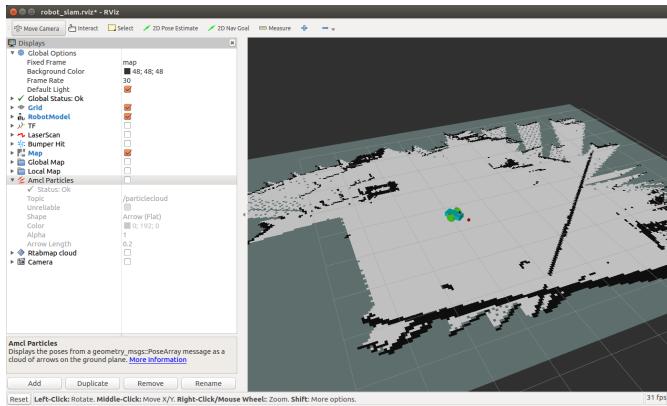


Fig. 10. Back Alley Occupancy

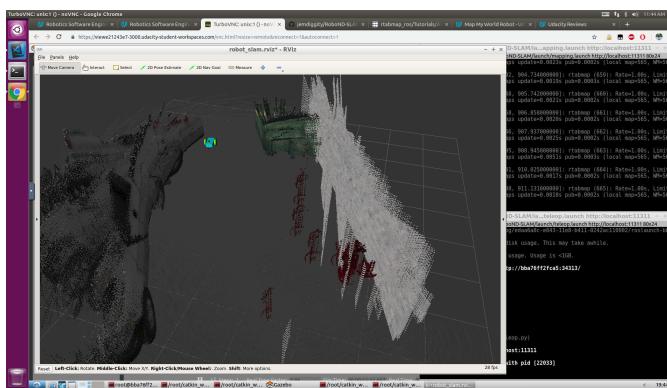


Fig. 11. Back Alley Map

Mapping the “Back Alley” world proved to be much more difficult. The mapping algorithm easily mixed up the rear and front wheel of the pickup truck causing mapping errors. A image from “rtabmap-databaseViewer” shows a closure between different truck wheels. Even with a bright orange traffic cone placed beside the wheel, the algorithm still had issues discerning the back and front wheels.

Perhaps RTAB-Map can be tuned to have better performance. Adjusting the parameters for RTAB-Map was attempted as per the ROS RTAB-Map tutorial, but met limited

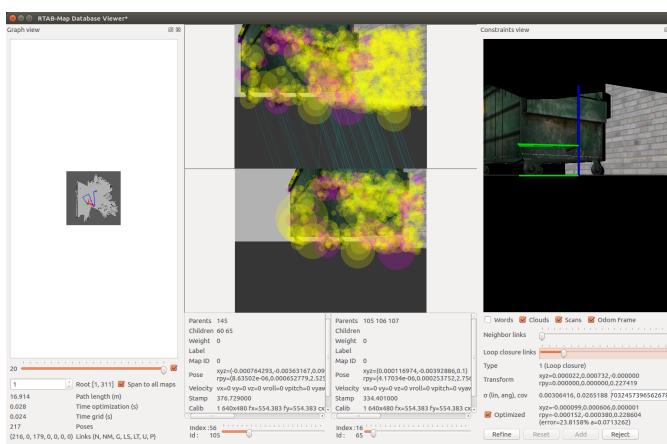


Fig. 12. Alley Closure 1

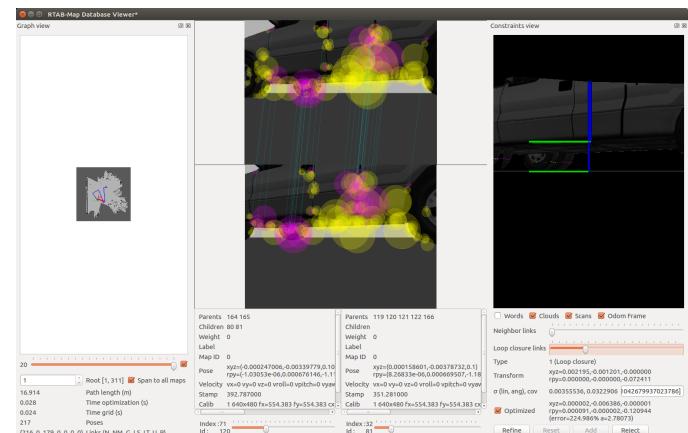


Fig. 13. Alley Closure 2

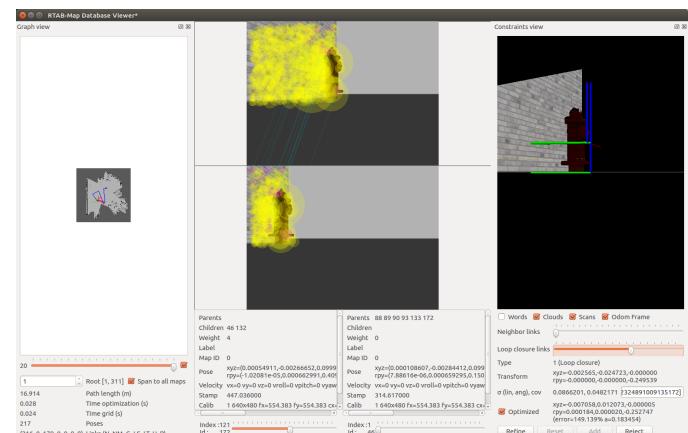


Fig. 14. Alley Closure 3

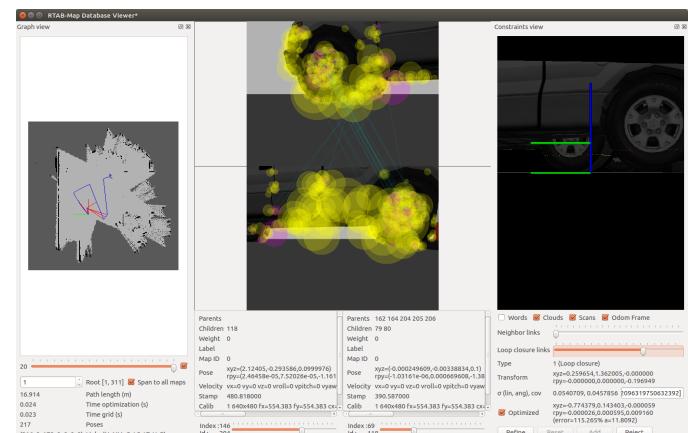


Fig. 15. RTAB-Map Confused

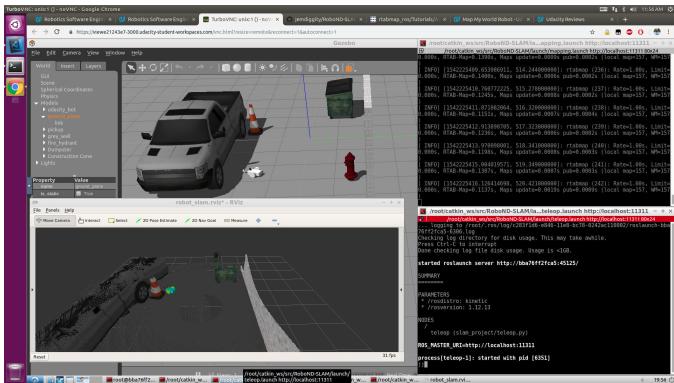


Fig. 16. RTAB-Map Confused with Cone

success. There was also some confusion about setting up the ROS environment on a local machine and whether RTAB-Map 1.8 or 1.9 would work.

Navigating the small robot around was painfully slow. It took around 10 minutes to complete 3 circles of the environments.

## 7 CONCLUSION / FUTURE WORK

Considering that the project on SLAM was only a simulation, it seems that it would be challenging to optimize and get satisfactory results in the real world.

In future, more automation would be helpful. Manually navigating the robot around the environment was extremely time consuming. It would be much more efficient to define a path for the robot to follow.

Another issue is a lack of useful documentation on elements of ROS and RTAB-Map.