Mapping in ROS using RTAB-Map

Merrick Campbell

Abstract—Mapping is an important part of robot navigation, whether maps are preloaded onto a platform for Monte Carlo Localization or as part of a SLAM pipeline. This paper applies Real Time Appearance Based (RTAB) Mapping to map two different environments, an interior Kitchen World and an exterior Custom Firehouse Environment. Both worlds could be successfully mapped in simulation using a differential drive robot with an RGBD camera and a laser scanner. For each world, 2D and 3D maps of the environment were created. Loop Closure was achieved 303 times for the Kitchen World and 27 times for the Custom Firehouse Environment.

Index Terms—Robot, Udacity, Mapping, ROS, RTAB-Map.

1 Introduction

For centuries, humankind created and utilized maps to explore the world. Having a map, whether a human or a robot, greatly improves environment understanding and localization. (Naturally, robots utilize a different *type* of map than humans.) In robotics, mapping can be accomplished using some combination of odometry, camera images, and laser scan data. These maps are formed by identifying distinct points between frames called landmarks and determining the relationship between these landmarks. Figure 1 shows a simulated robot with camera data and a depth point cloud at the start of a mapping session.

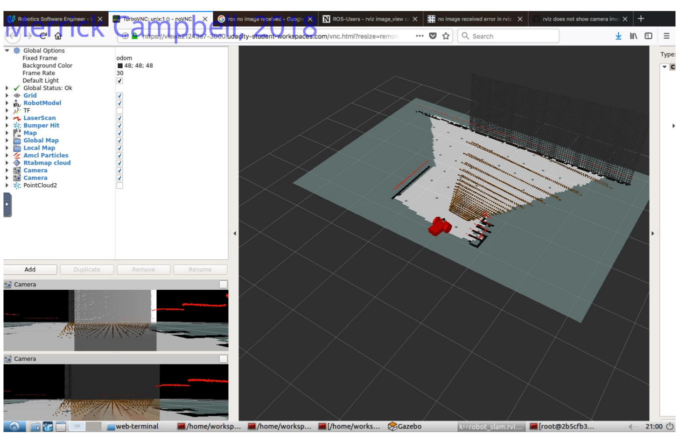

Fig. 1. Robot model at the start of the mapping session

While there several different mapping algorithms and techniques exist, this paper will explore Real Time Appearance Based (RTAB) Mapping. Additional information about RTAB and other mapping processes can be found in Section 2. The details of the simulated robot and world environments will be discussed in Section 3 and the simulation results will be covered in Section 4. Section 5 contains additional performance discussions and context. Final conclusions are drawn in 6.

2 Background

Mapping is the process of constructing a representation of the world from sensor inputs. Loop closure occurs when a robot can identify the current location from information gained during a prior visit. Closing the loop corrects from errors introduced from odometry that have propagated throughout the map.

When loaded onto a robot, the robot can use the map for localization to determine it's pose (position and orientation). Both of these techniques can be performed simultaneously in a process called SLAM (Simultaneous Localization And Mapping) to create a world map and determine pose in real time. RTAB is one of several mapping algorithms that can be used either solely as a mapping tool or part of a SLAM pipeline.

2.1 Real Time Appearance Based (RTAB) Mapping

RTAB Mapping is an efficient mapping technique that primarily uses visions sensors to map the environment. This process can be accomplished with a monocular camera, but additional depth information from either an RGBD camera or a laser scanner is required to derive geometric distance information. Features are extracted from the image data using Speeded Up Robust Features (SURF) and then added to the map database as landmarks. For RTAB Map, loop closure uses a ”bag-of-words” approach.

2.2 Other Mapping Techniques

2.2.1 Gridmapping

Another form of mapping is Gridmapping. In its simplest form, a grid map is a binary map that stores information about free (0) and occupied (1) space. This style of mapping typically produces two dimensional information that can be partially extended into a third with a little bit of additional information in either a Height or Elevation Map.

2.2.2 Octomap

Octomap is an efficient probabilistic mapping framework. [1]. This framework represents free and obstructed space in 3D using efficient compression models. The probabilistic approach helps with sensor fusion from several potentially noisy inputs. This library can be used without ROS as a standalone C++ library, potentially resulting in less system bloat for embedded systems.

2.2.3 Maplab

Maplab is yet an other mapping and localization framework bundled by ETH-Zurich [2]. Unlike RTAB mapping, Maplab is a visual-inertial framework that uses camera images and pose information (instead of encoder odometry) for mapping. Loop closure and map construction can be performed either ”online” in realtime or ”offline” on a server.

3 Simulation Configuration

The simulations conducted involved a small robot inside two different world environments, a stock Gazebo interior Kitchen Model and a custom exterior Fire Department HQ Building. Each of the environments were mapped using RTAB Map.

3.1 Robotic Platform

The robotic platform used a differential drive system for locomotion while mapping the environment. There is a single caster at the rear of the robot while two drive wheels at the front of the robot provide the locomotion. The robot has a LIDAR scanner and a RGBD camera at the front of the robot for mapping (Figure 2).

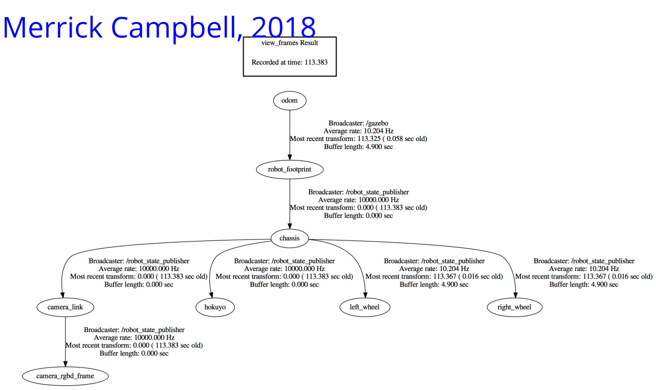

Fig. 2. Robot Frames

3.2 World Environments

3.2.1 Kitchen Environment

The first world mapped was an interior kitchen environment from Gazebo (Figure 3). This environment represents a typical home or apartment environment that a physical version of the robot could traverse.[1]

3.2.2 Custom Environment

The second world mapped was an exterior scene at a fire department (Figure 4). The world contained the fire house, a fire engine, a fire hydrant, dumpster, and a brick wall. While this is not a typical environment for the robotic platform, it is useful to compare RTAB's performance in large and small spaces with different obstacles.

1. Special Thanks to Udacity Slack user *froohoo* for pointing out that collisions were not enabled properly in the Udacity provided kitchen model.

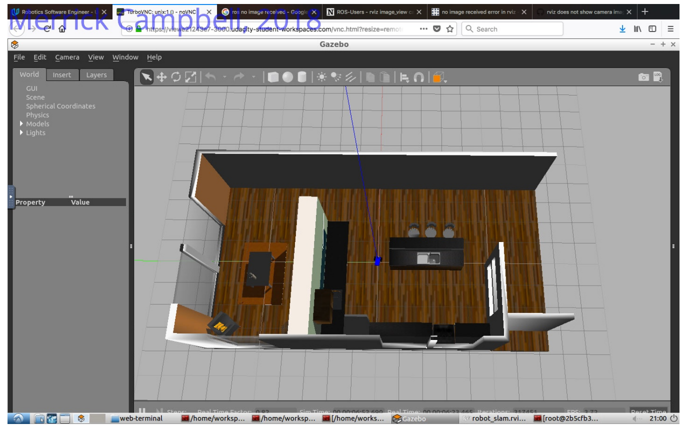

Fig. 3. Kitchen World Environment

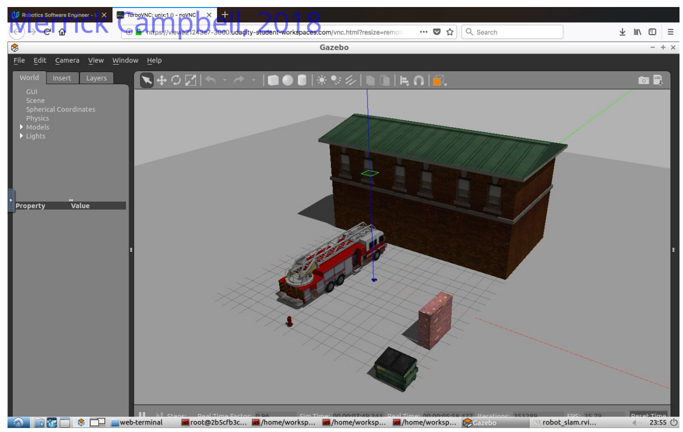

Fig. 4. Custom World Environment

4 Results

The robot platform successfully mapped both the indoor and outdoor worlds. The indoor map resulted in 303 loop closures while the outdoor map resulted in 27 loop closures.

4.1 Kitchen Map

4.1.1 2D Map

The 2D map created by RTAB map contained 303 loop closures. Figure 5 shows the resulting map, with the robot's traversed path in blue and the loop closures in red.

4.1.2 3D Map

The tool RTAB Map Database Viewer can be used to visualize the 3D dimensional map of the environment. The kitchen is clearly visible in Figure 6. The reconstructed map minimizes odometry errors from the raw data and repairs the noisy geometry. For reference, Figure 7 shows the raw point cloud in RVIZ.

4.2 Custom Map

4.2.1 2D Map

The 2D map of the firehouse exterior created by RTAB map contained 27 loop closures. Figure 8 shows the resulting map, with the robot's traversed path in blue and the loop closures in red. This exterior map was much larger than the kitchen map. The majority of the world was traversed only once, but a small section in the bottom right was traversed several times to ensure a few loop closures. Though the fire truck's tires were mapped in the 2D map, the truck's chassis was not represented in the 2D map.

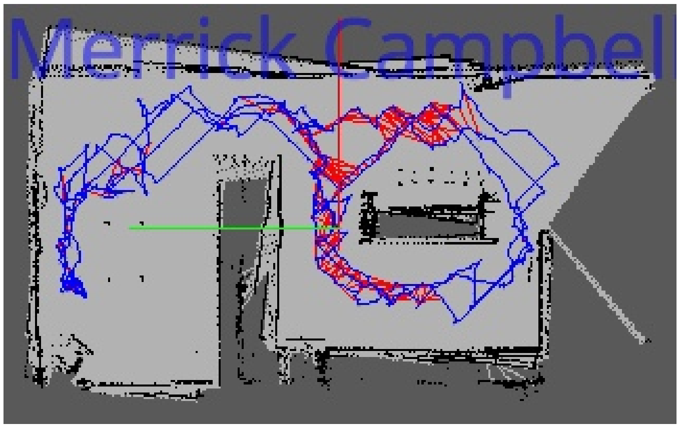

Fig. 5. Kitchen World - 2D Map

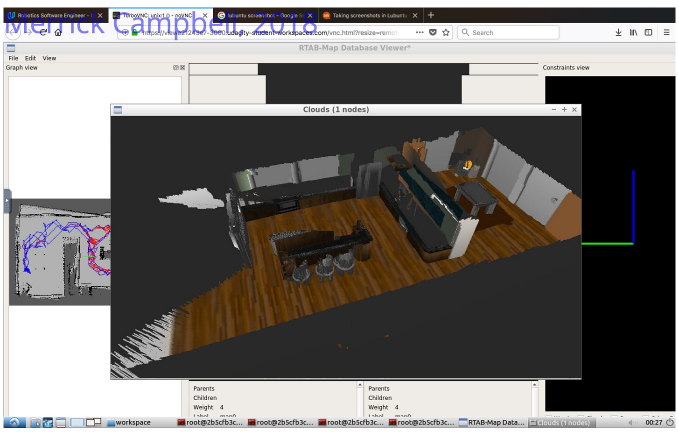

Fig. 6. Kitchen World - 3D Map

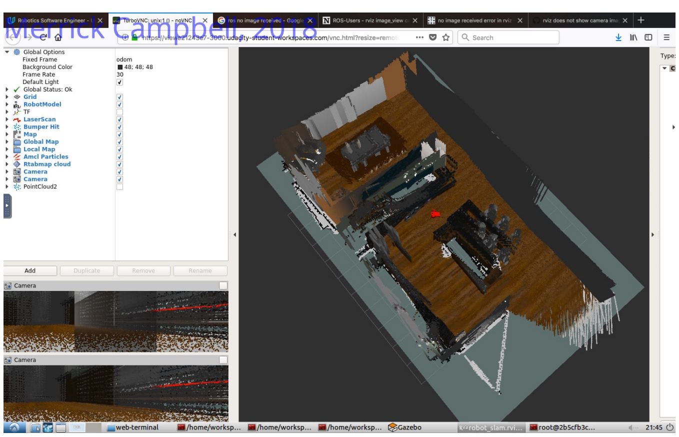

Fig. 7. Kitchen World - RVIZ

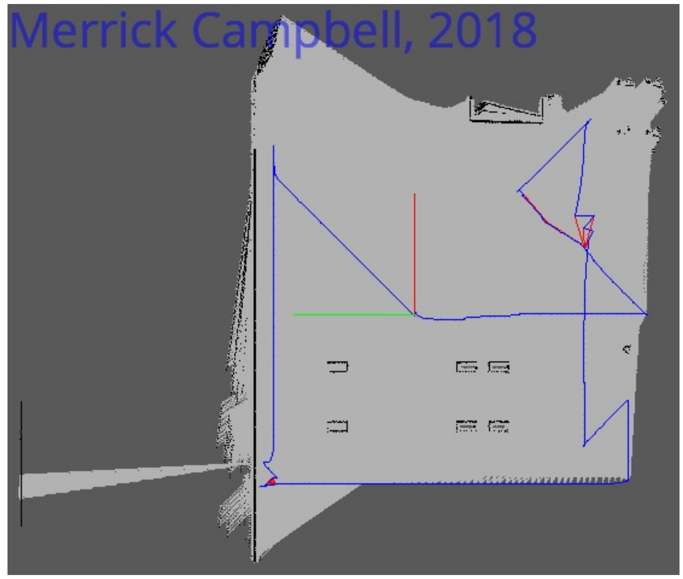

Fig. 8. Custom World - 2D Map

4.2.2 3D Map

Like the kitchen 3D map, the firehouse 3D map was completed with a similarly level of fidelity. Unlike the firehouse 2D map, the 3D map clearly shows the entire firetruck. Figure 9 shows th 3D map of the custom firehouse world.

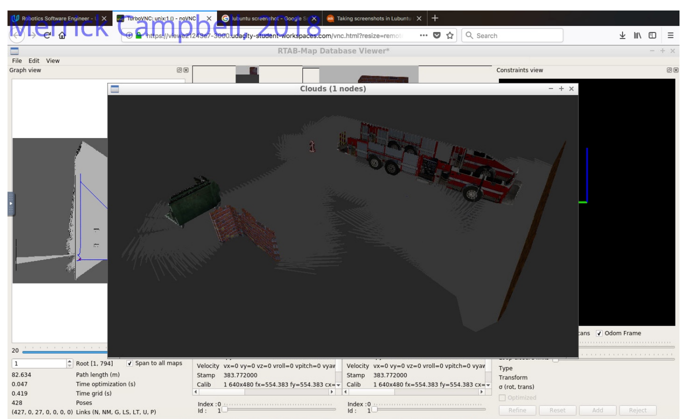

Fig. 9. Custom World - 3D Map

5 Discussion

Both of the worlds were successfully mapped by the robot. The Kitchen World had more loop closures than the Custom World (303 vs 27). This was due partially to the fact that the Kitchen World was smaller and the robot could perform more laps within a similar time window. However, though the Kitchen is completely mapped in the first environment, the robot should have driven farther out of the kitchen to fully map the side room.

The 2D map of the Custom Firehouse World included just the tires of the fire truck and did not mark that area as an obstacle. While a small robot may be able to fit underneath, the test robot would not fit. In a real world environment, the fire engine may be a dynamic object and should not be included in the static world map anyway. Ideally, the robot's local planner should be able to detect the fire engine as an obstacle and avoid a collision.

6 CONCLUSION / FUTURE WORK

This paper has demonstrated the usefulness of RTAB Mapping in creating a map of a robot's environment. RTAB map was able to construct both 2D and 3D maps of the Kitchen World and the Custom Firehouse World with 303 and 27 loop closures, respectively.

Mapping is necessary for robotic navigation and path planning. Access to an accurate map improves a robot's ability to reach a target destination. If a preloaded map is present on a robot, the platform's computational resources could be focused on other tasks such as gesture recognition. Future work with RTAB map could focus on using the generated map for Monte Carlo Localization. In this scenario, the robot would perform localization using the 2D map generated by RTAB map. Alternatively, RTAB map could be used for the map creation portion of a SLAM pipeline. Potentially, RTAB map could be applied to a real robot with an RGBD camera and laser scanner.

REFERENCES

[1] Armin Hornung and Kai M. Wurm and Maren Bennewitz and Cyrill Stachniss and Wolfram Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees," *Autonomous Robots*, 2013. Software available at urlhttp://octomap.github.com.

[2] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, 2018.