Map My World

Dale Cassidy

Abstract—Utilized ROS packages to implement SLAM inside provided and created maps in Gazebo and RViz simulation environments. Controlled a mobile robot to traverse through environments to help create loop closures. Created 2D occupancy grids and 3D octomaps of environments.

Index Terms—Robot, Gazebo, RViz, SLAM.

1 Introduction

For robots to be useful they must be able to map unknown environments. The purpose of the project is to map an unknown environment without a map. SLAM which stands for Simultaneous Localization and Mapping solves the problem of being able to both localize and map in unknown environments where a map is not provided.

The robot for this project is operating in simulated 3D. The provided environment is a kitchen dining environment and the second created environment is a custom made cafe. SLAM is important in mapping and navigating both of these worlds. Even if the robot has a map, the features within the space may change and the robot needs to know this in order to successfully navigate the area to reach a goal without hitting obstacles.

In more detail, the mapping part of SLAM assumes the robot's poses having access to only movement and sensor data. SLAM is more challenging than just localization and mapping, because in SLAM a robot must construct a map of the environment while simultaneously localizing itself relative to that map without a map or poses. There is also noise in the measurements and movements making the problem even more challenging. It's difficult because the map is needed for localization and the robot's pose needed for mapping.

2 Background

SLAM is an important topic in Robotics since the robot needs to know where it is before it can figure out how to reach it's goal and it needs to be able to create and update internal maps of unknown and/or changing worlds.

In resource constrained systems, 2D mapping is very beneficial because it's less computationally expensive. However, robots do live in a 3D worlds so it's important that all the 3D structures are accurately mapped. 3D mapping provides the most accurate way to avoid colliding into obstacles and most accurate navigation for mobile and path planning.

In this project's lectures, 2 approaches were demonstrated- 1 for FastSlam and the other for GraphSlam. FastSlam is adapted to grid maps resulting in an occupancy grid mapping algorithm. In occupancy grid mapping you can map any arbitrary environment and divide it into a finite number of grid cells. And, GraphSlam implements realtime appearanced based mapping (RTAB-Map).

FastSlam uses a particle filter along with a low dimensional Kalman filter to solve the SLAM problem. Each particle holds a guess of the robot's trajectory as well as its own map. The largest issue with the FastSlam algorithm is that it relies on known landmarks. However, this is not something that can always be counted on in unknown environments.

GraphSlam uses constraints to represent relationships between the robot poses and the environment and then tries to resolve constraints to produce the most likely map given the data. It can do this because it implements full SLAM which takes into account the robot's entire trajectory. GraphSlam, unlink FastSlam, does not need to know any landmarks. RTAB-Map was chosen for this project for it's computational and memory efficiency as well as it's effectiveness in mapping 3D environments.

3 Scene and Robot Configuration

For the custom configuration, a cafe was chosen. There are 6 added desks as well as a couple of coke cans on the floor and a construction cone on the ground. A small red brick wall was added as a divider. There is also an added cabinet in the corner.

The robot used has a laser range finder to map 3D as well as an RGBD camera. The package has 5 subdirectories- scripts, worlds, meshes, urdf and launch. The scripts' directory holds the teleop script to control the robot. The worlds' directory holds both the supplied kitchen dining world as well as the custom cafe world. The meshes directory holds the hokuyo mesh. Urdf holds the robot specification files. The launch directory holds the robot description's launch file, as well as launch files for the world, mapping and launching rviz with custom additions such as the map and robot model.

4 Results

The first 3 pictures, figures 1 to 3, show the kitchen dining world, its 2D occupancy grid map and 3D map consecutively. The second set of 3 images, figures 4 to 6, show the cafe world and its 2D occupancy grid as well as 3D map consecutively.

Fig. 1. Kitchen dining

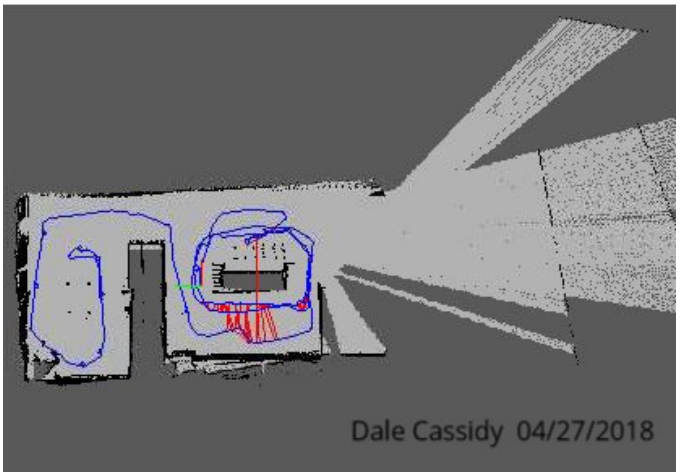

Fig. 2. Kitchen Dining Occupancy Grid

The following 2 images, figures 7 and 8, show the tf frames of the custom robot and the rqt graph demonstrating the published and subscribed messages in the ROS package:

5 Discussion

The mapping for the given kitchen dining world went very well. There were 147 global loop closures from just a couple of loops around the environment. Also the 3D map clearly shows the features of the room.

The mapping for the custom cafe did not go nearly as well even though there were 120 loop closures. Because there was not enough variation in the features (6 tables are identical) and not enough overall features, RTAB-Map was not able to successfully map the area. The map started to rapidly degrade when the robot was turned around a few times in place whereas this did not happen with the robot in the given kitchen dining world.

6 Future work

Overall, the robot mapped the given kitchen dining environment well. However, it did poorly mapping the custom cafe environment. Given more time, I would create a custom cafe with more features so that the RTAB-Map converges on a more feature rich rendering of the environment.

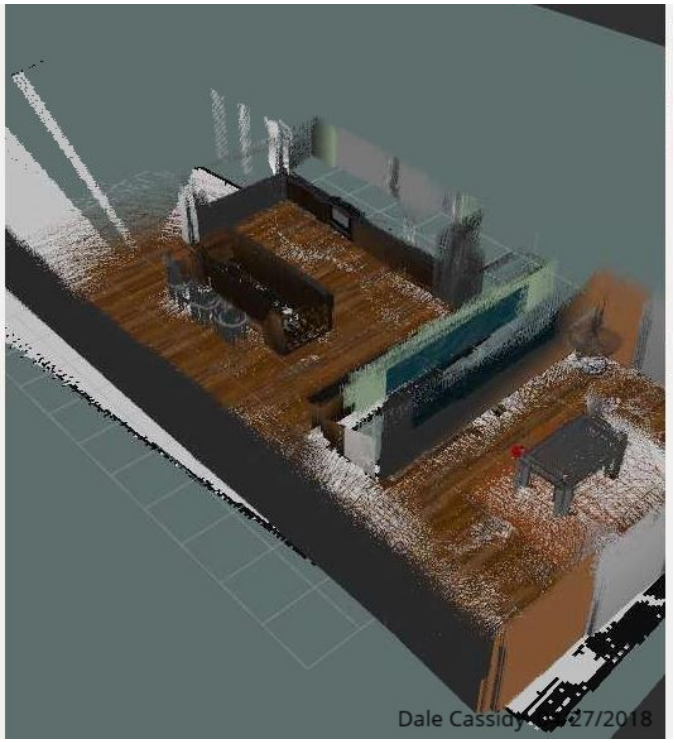

Fig. 3. Kitchen Dining 3D

Fig. 4. Cafe

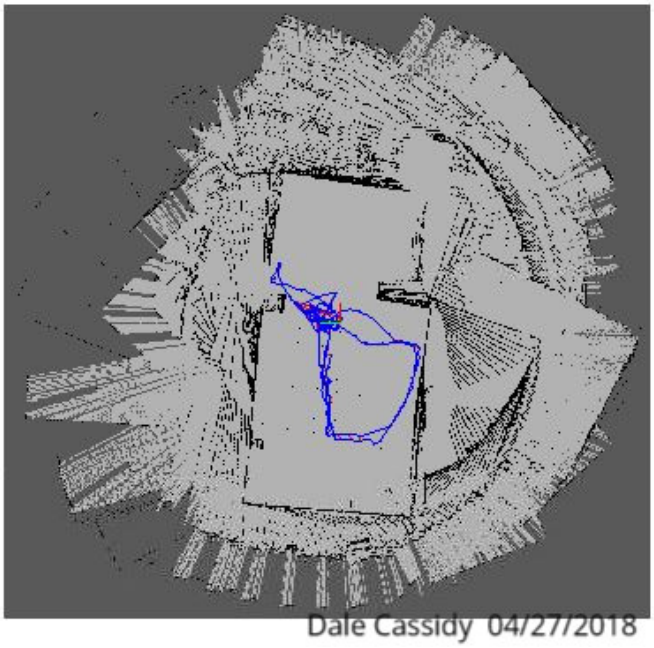

Fig. 5. Cafe Occupancy Grid

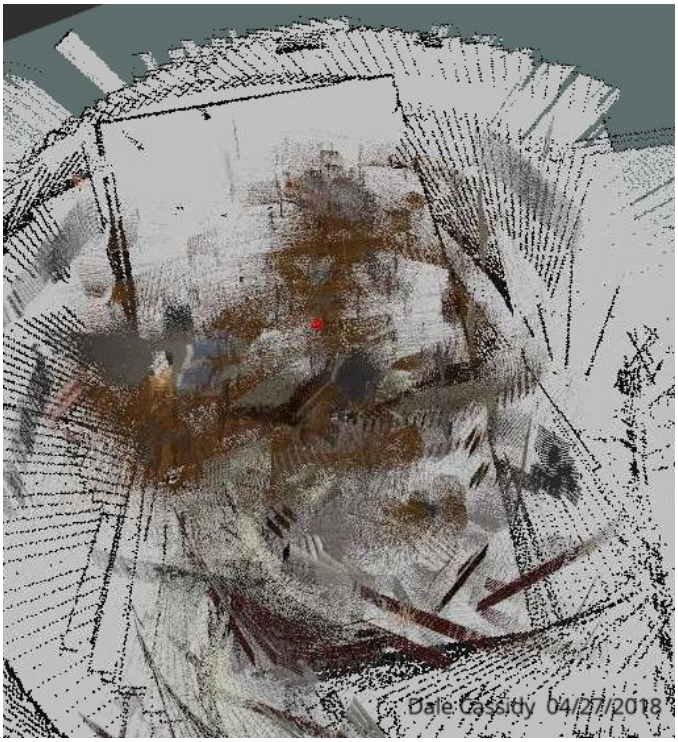

Fig. 6. Cafe 3D

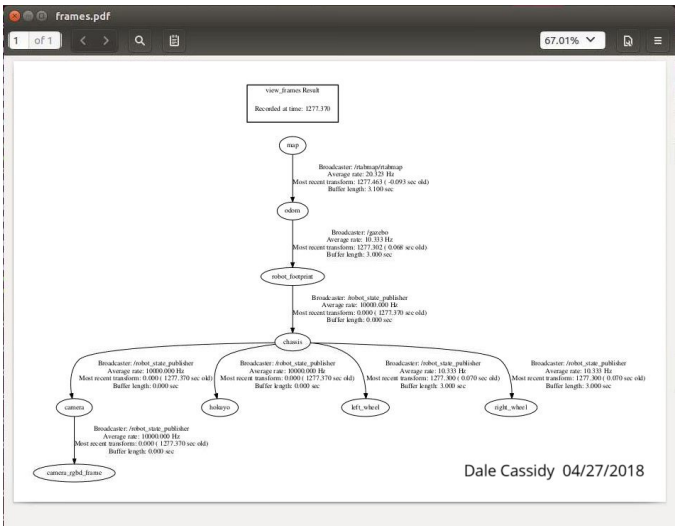

Fig. 7. TF Frames

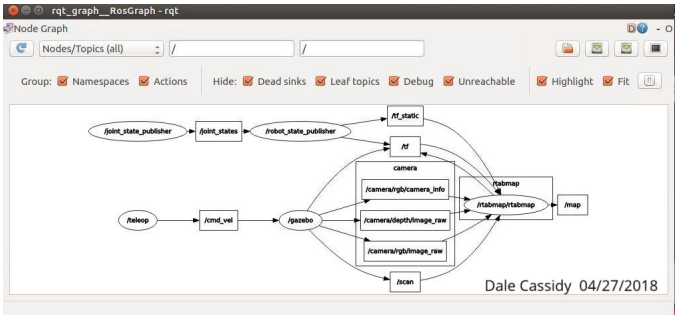

Fig. 8. RQT graph

Another environment that would be fun to apply RTAB Mapping would be on a flying drone in a 3D simulated environment.

Also, the Jetson TX2 was used on this project. Even though the script speeding up the clocks was used, it still did not run as fast as expected. Given more time, it would be interesting to spend more time on just increasing the computational efficiency of RTAB-Map on the Jetson TX2.

REFERENCES

[1] L. Lamport, *LATEX: a document preparation system: user's guide and reference manual*. Addison-wesley, 1994.