Map My World

Harrison Seung

Abstract—The following report outlines the Map My World project of Term 2 for Udacity's Robotic Software Engineering nanodegree. Utilizing ROS and Gazebo, the requirement is to develop a ROS package which will generate a 2D occupancy grid and a 3D octomap in two separate environments, one supplied and one self created. The 'udacity bot' robot created in a previous project is leveraged with upgrades to the sensors to supply sensor messages for the RTAB-Map SLAM approach. The robot is teleoped around the supplied environment to generate the 2D/3D map and data confirmed using rtabmap-Database-Viewer. After successfully mapping of the supplied environment, the same approach is applied to the self-created map as well.

Index Terms—Map My World, Robotic Software Engineering Degree, Udacity.

1 Introduction

In robotic mapping and navigation, Simultaneous Localization and Mapping (SLAM) is the process of generating an unknown map of an environment while simultaneously keeping track of a robots position within it. In the following project, a robot developed in the 'Where am I?' project is deployed in simulation to demonstrate the SLAM approach called RTAB-Map (Real-Time Appearance-Based Mapping). RTAB-Map is a RGB-D SLAM approach based on a global loop closure detector with real-time constraints. It is used to generate a 3D point cloud of the environment and to create a 2D occupancy grid map for navigation.

2 Background

If a robot is placed in an environment with an unknown map, it needs to figure out how to map the environment and localize itself within it. This is achieved through SLAM, and some of the most popular SLAM algorithms used by roboticists today are Occupancy Grid Mapping, Grid-Based FastSLAM, GraphSLAM, and RTAB-Map.

2.1 Occupancy Grid Mapping

The Occupancy Grid Mapping is a 2D algorithm which takes a known environment and decomposes it into grid cells. A binary Bayes filter is then implemented to estimate the occupancy of each cell individually. This algorithm requires a static environment or known landmarks and is not suitable for real time mapping.

2.2 Grid-based FastSLAM

The Grid-based FastSLAM algorithm is an extension of the FastSLAM algorithm. The environment is modeled without predefined landmarks by implementing occupancy grid maps. It solves the sampling motion and importance weight techniques through Monte Carlo Localization (MCL) and map estimation through occupancy grid mapping.

2.3 Graph-SLAM

The Graph-SLAM algorithm solves the full SLAM problem, recovering the entire map and pose instead of the just the most recent, enabling it to consider dependencies between the current and previous poses. Graph-SLAM constructs a graph of robot poses, environmental features, motion constraints and measurement constraints. The Graph-SLAM approach is more accurate than FastSLAM as it uses all the data at once whereas FastSLAM uses small portions of data gathered through particle location.

2.4 RTAB-Map

The algorithm that is used for this project is Real Time Appearance Based Map or RTAB-Map, a Graph-SLAM approach which uses data collected from vision sensors to localize the robot and map the environment. In appearance based mapping, a process called loop closure is used to determine if a robot has seen a location before. As the robot explores its map, each image is compared to each new image and thus the complexity increases linearly as duration increases. RTAB-Map optimizes this approach to allow the loop closure process to complete prior to the next image being acquired using an idea called Visual Bag of Words. Visual Bag of Words causes the loop closure detection to be checked against a constrained set of images in short term memory while older images are moved into long term memory, reducing complexity and computation time.

2.5 2D and 3D mapping

The importance of performing both 2D and 3D mapping is due to environmental topographical and on board computational constraints. While 3D mapping will give a more accurate view of a robot's environment to prevent collisions due to height constraints, 3D mapping is more computationally expensive and thus use more energy, a major priority constraint for mobile robots. Additionally, 2D mapping is faster to perform than 3D and cheaper in terms of computational cost and hardware cost.

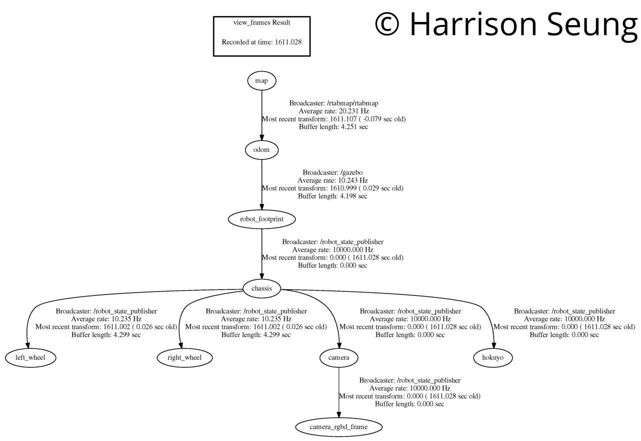

Fig. 1. Transform Tree

3 Scene and Robot Configuration

The robot model used was based on the 'Udacity_Bot' created in the previous 'Where am I?' Udacity project. The camera was replaced with a kinect RGB-D camera by modifying the existing 'udacity_bot.xacro' and 'udacity_bot.gazebo' files. A diagram of all the coordinate frames are provided in the tree below.

3.1 Gazebo Worlds

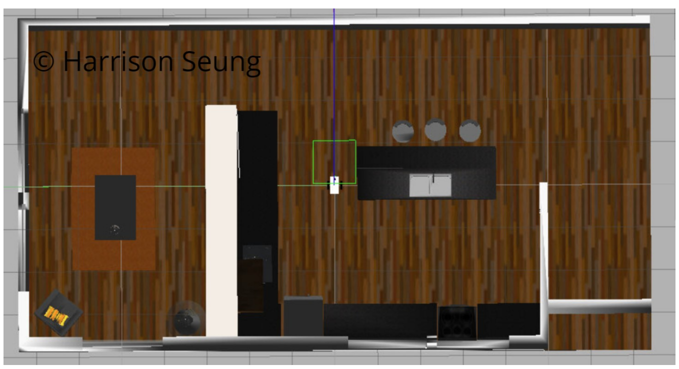

Fig. 2. Kitchen Dining World (supplied)

Fig. 3. Cafe World (custom)

Two gazebo worlds were created: an Udacity supplied world called 'kitchen_dining.world' and a custom world called 'cafe.world'. The objects in the 'cafe.world" were selected to ensure sufficient space was left for the robot to navigate while supplying sufficient landmarks for the robot to identify unique locations. The ROS package structure was setup similar to the previous project and is presented below.

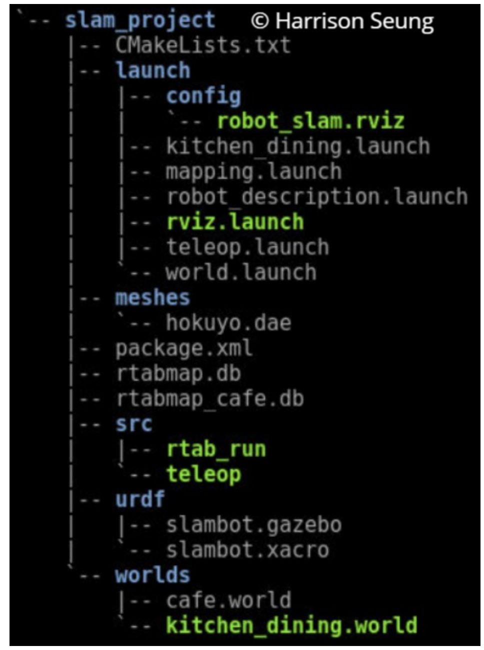

Fig. 4. ROS Package File Structure

4 Results

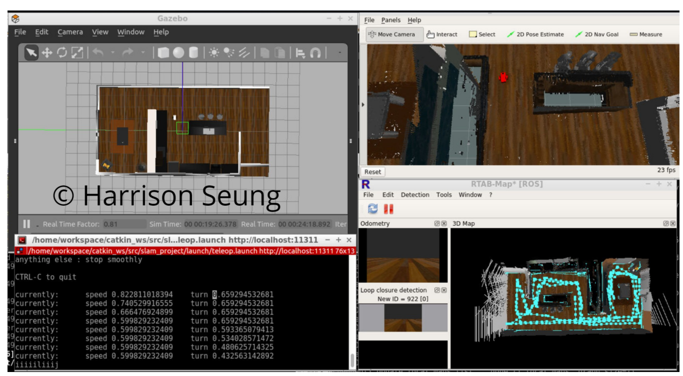

Fig. 5. Kitchen Dining capture view

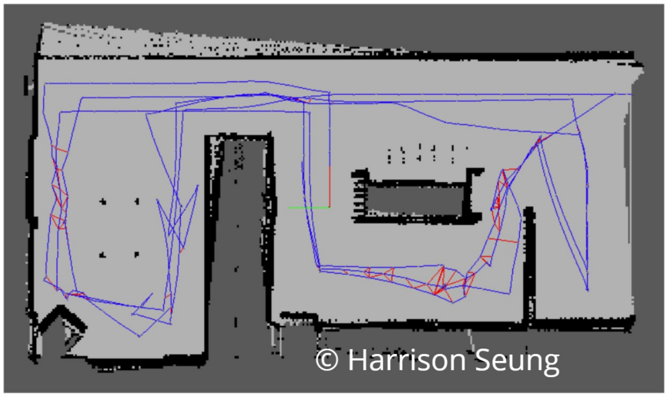

Fig. 6. Kitchen Dining 2D map

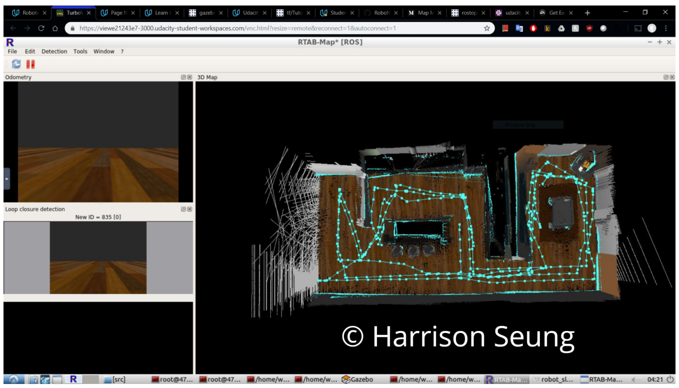

Fig. 7. Kitchen Dining 3D map

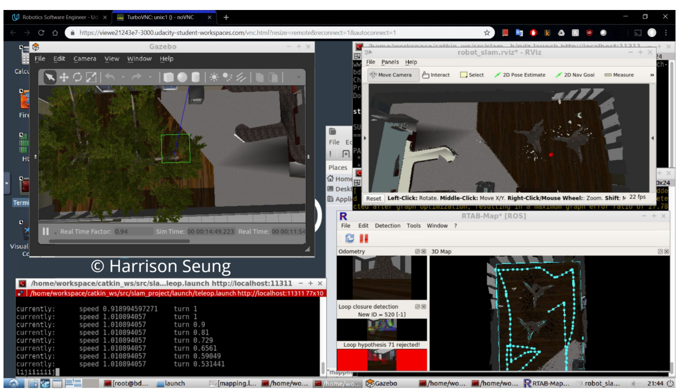

Fig. 8. Cafe capture view

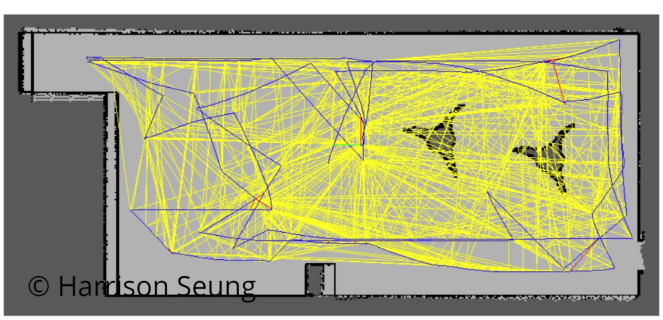

Fig. 9. Cafe 2D map

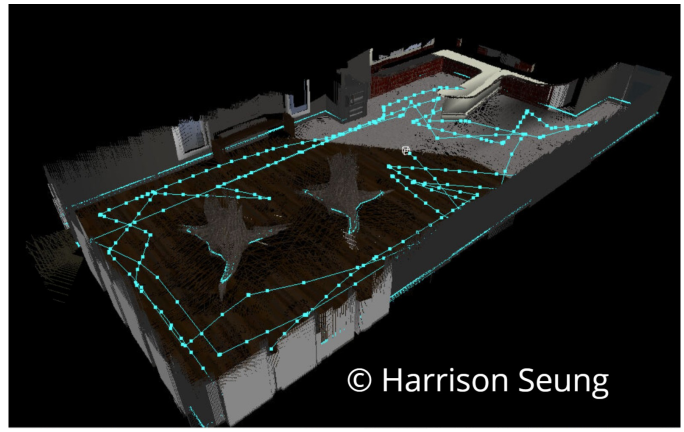

Fig. 10. Cafe 3D map

5 Discussion

The robot was manually driven through each environment using keyboard teleop commands. For the kitchen dining world, the robot was driven around the same path for 2-3 cycles until each feature of the map became obvious. In larger pathways such as the kitchen region, the robot would be rotated several times to scan surrounding areas that may not have been in the robots field of view when moving in a straight path. The cafe world started as a predefined gazebo model with no objects in the wood paneled area. Initial attempts at mapping this open environment caused significant distortion on the generated the 2D and 3D maps. Trees were added in the center of the cafe to allow better pattern recognition of different locations. Additionally, the Loop Closure Constraint was modified to include both Visual and ICP (Iterative Closest Point) resulting in an increase in precision of loop closure and elimination of distortion. Over multiple runs for each environment, the robot was successful in recreating the same 2D and 3D maps for both environments.

6 Future work

SLAM is a powerful tool in robotics and the RTAB-Map algorithm provides a cost efficient method to implement this using cost effective RGB-D sensors. Potential applications are quickly surveying hazardous environments for humans such as compromised buildings and infrastructure due to natural disasters. On a consumer side, this is an easy application for a home service robot to navigate around a home residence for cleaning or delivery.