SLAM for Robots

Navin Rahim, navin.rahim@gmail.com

Abstract—This paper discusses the simultaneous localization and mapping(SLAM) of a robot in an environment. A robot equipped with a RGB-D camera was used to perform RTAB-map SLAM using ROS packages in a kitchen worlds and a custom world. The robot performed SLAM and created accurate 2D and 3D maps of the worlds.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, Localization.

1 Introduction

ROBOTS require understanding of the world around it to perform its tasks. Part of the understanding is the map of the world where the robot has to operate and determining where exactly the robot is in that map.

Although maps can be predefined and fed to the robot, this is not feasible in real world conditions. The robot might be operating in a dynamic environment or performing actions in a emergency scenario and in these cases providing a map to the robot is difficult. Simultaneous Localization and Mapping(SLAM) techniques are useful in these scenarios.

SLAM involves creating the map and localizing the robot within the map on the fly. It utilizes the sensor data and creates a understanding of the environment on the go and adjusts the map based on various inputs from the different sensors.

2 Background

SLAM is a necessity for robots that have to operate in an unknown environment. It helps the robot to map the environment and localize itself which can later be used for tasks such as path planning.

SLAM involves various challenges. The major challenge is that neither the map or the pose of the robot are not provided. Combined with the uncertainties in the sensor measurements, SLAM becomes a hard problem to solve even though highly useful.

SLAM involves finding the map of an environment given the sensor data and control signals. SLAM can be categorized into online SLAM and full SLAM. While Online SLAM involves calculating the map at every instant from the previous map and current measurement and controls, full SLAM focuses on collecting the entire sensor measurements and controls and calculating the map from it.

SLAM can also be categorized on its nature - Continuous and Discrete SLAM. Continuous SLAM focuses on determining the continuous poses and the object locations. Discrete SLAM focuses on the correspondence between objects across frames. It tries to answer whether the robot has seen the current configuration before or not.

2.1 Occupancy Grid Mapping

Occupancy Grid Mapping is a mapping technique which implements mapping in a robot. It divides the map into grid cells and provides information about whether the grid is occupied, free or unknown. It implements a binary Bayes filter.

2.2 Grid-based FastSLAM

FastSLAM is a SLAM technique that uses particle filters for mapping an environment. Each particle will have the robot pose, weight and a map of the environment. FastSLAM uses these particles to estimate the trajectory that the robot is moving in. It estimates the map using a low dimensional Extended Kalman Filter approach. Although FastSLAM is easy to implement, it is inefficient since particle filters generate sample inefficiency.

Grid-based FastSLAM adapts FastSLAM to grid maps to solve this inefficiency. It also contains the particles with poses, weight and map. The particles hold information about the trajectory the robot has taken. It then utilizes the Occupancy Grid Mapping to estimate the map using the known poses.

2.3 GraphSLAM and RTAB-map

GraphSLAM solves the full SLAM by using a graph based approach. The nodes of the graph are the robot poses and the edges are the constraints. Constraints are either measurement of motion. Motion constraints represent the motion of the robot and measurement constraints represent the measurement to a object in the environment. The measurement and controls from a robot are used to build the graph with the constraints. The constraints contains the noise from motion and sensor measurement. The graph is then optimized to reduce the overall error thus providing the best possible estimate of the environment as a map.

RTAB-map, or Real Time Appearance Based Mapping, is a GraphSLAM implementation for SLAM. It utilizes data collected from the vision sensors to perform SLAM. From each frame of the image from the sensors, RTAB-map identifies features and associates them to a bag of words. When a similar image as seen before appears, a loop closure appears. Loop closure is where the optimization happens, allowing to fix the map thus far.

RTAB-map allows for real-time mapping as it is optimized for large scale and long term SLAM. Thus, loop closures are done in real-time in an efficient way.

This project implements RTAB-map ROS package to perform SLAM in two different worlds created in simulation.

3 Scene and Robot Configuration

3.1 Kitchen World

The Kitchen world was provided by Udacity. It is built in the Gazebo simulator. Fig.1 shows the kitchen world.

Fig. 1: Kitchen World

3.2 Custom World

A custom world was created for performing SLAM for a robot. The custom world was built using the Gazebo simulator. Objects were added from the Gazebo models. The idea was to create a car garage with other objects such as cardboards and number boxes. Features were added so that the robot could identify for performing loop closures.

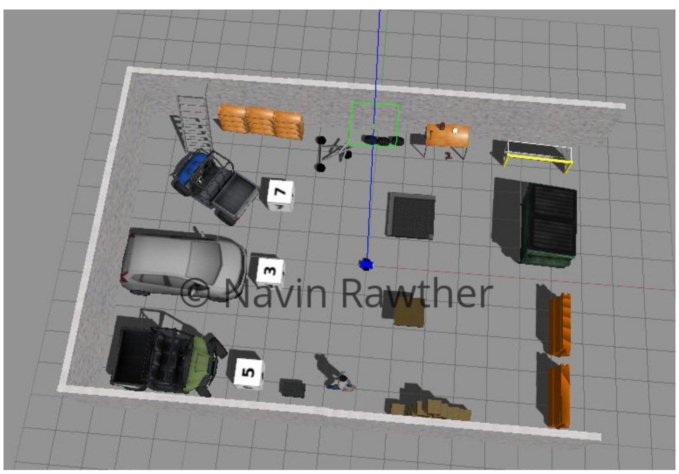

Fig. 2: Custom World

3.3 Robot Configuration

The robot used for performing SLAM is the Udacity bot. This was previously used to perform localization in another project. The bot has a Hokuyo laser and a RGB-D camera attached to the base. It was two wheels on the sides and two caster wheels for stability. Fig.3 show the robot used in this project.

Link	Shape	Specification	Remarks
Chassis	Box	0.4 x 0.2 x 0.1	
Casters	Sphere	Radius = 0.05	Front and Back
Wheels	Cylinder	Radius = 0.1 Length = 0.05	Left and Right
Camera	Box	0.05 x 0.05 x 0.05	Connected to chassis at [0.2,0,0,0,0,0]
Hokuyo Laser	Box	0.1 x 0.1 x 0.1	Connected to chassis at [0.15,0,0.1,0,0,0]. Mesh file used for visualization.

TABLE 1: Udacity bot specs

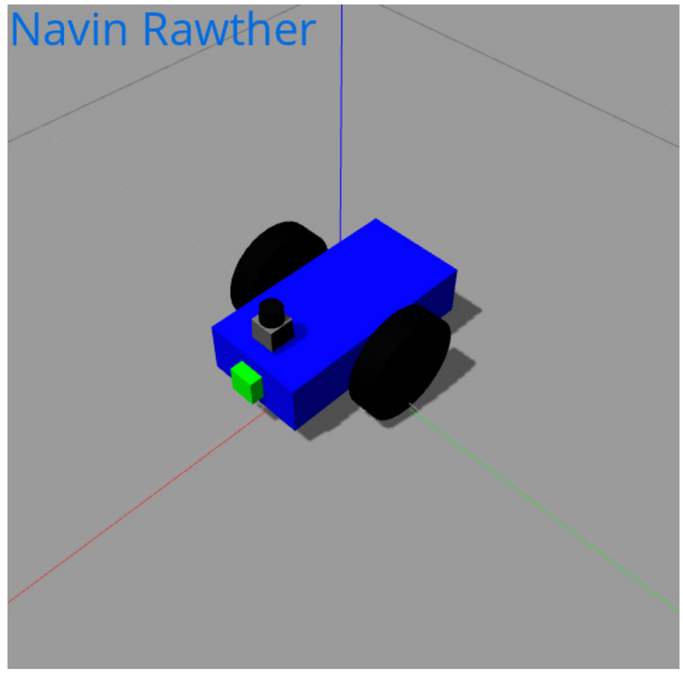

Fig. 3: Udacity bot

3.4 Package Structure

A package was created for incorporating the launching of the robot, SLAM node and navigation purposes. The package is named slam_project. The robot models are placed in the udacity_bot package. turtlebot_teleop package contains the files that are used for teleoperating the robot in the worlds.

4 Results

4.1 Kitchen World

The robot was launched in the kitchen world and was moved around multiple times to generate the map with appropriate loop closures.

4.2 Custom World

The robot was spawned in the custom built map in gazebo. The robot was moved across the map a few times to obtain the map.

5 Discussion

The 3D map and 2D occupancy grid obtained in both the worlds were accurate. The map obtained in kitchen world

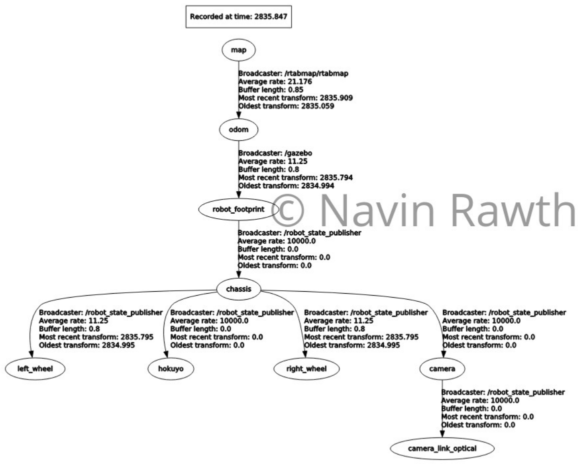

Fig. 4: Transform tree

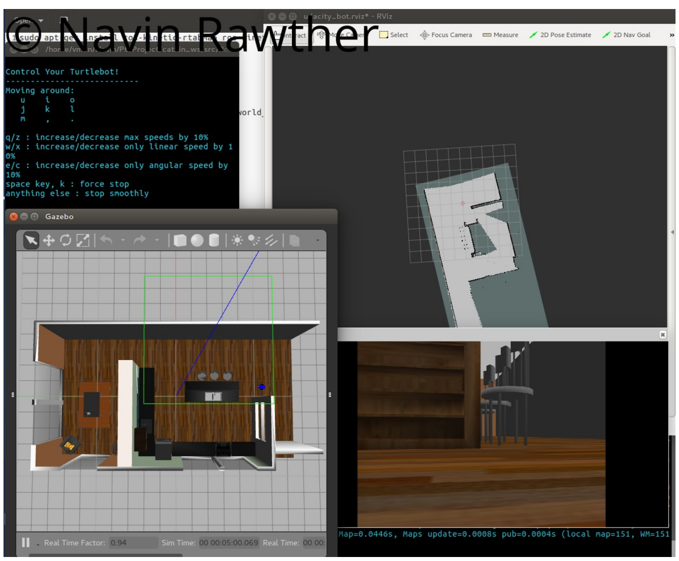

Fig. 5: Kitchen world SLAM process

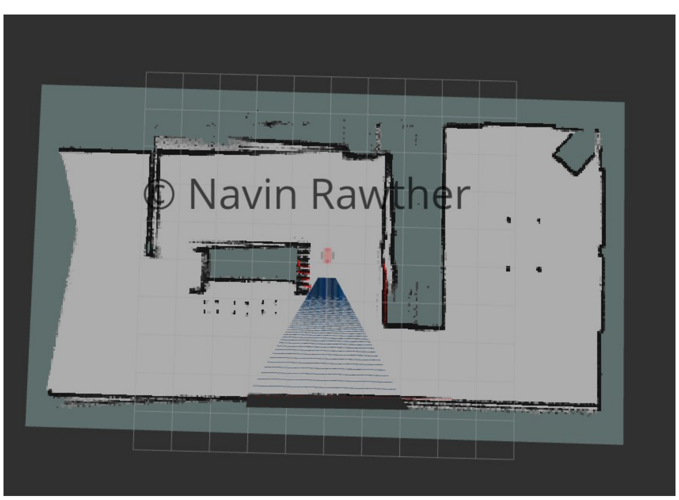

Fig. 6: Kitchen World - 2D map

Fig. 7: Kitchen World - 3D map with robot trajectories

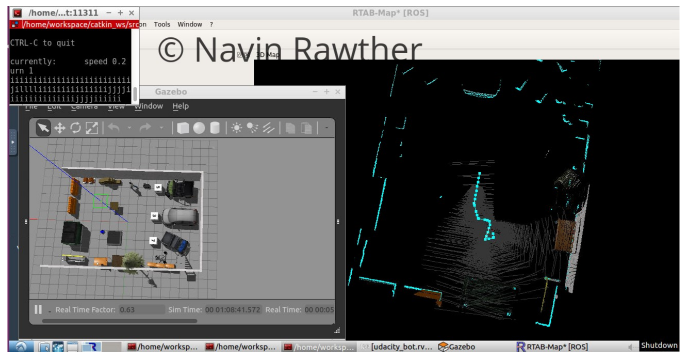

Fig. 8: Kitchen world SLAM process

were the best among the two worlds. Kitchen world had solid objects placed across the floor and had distinct features. The custom world creation was a difficult process. Too less objects made the loop closure happen at different places mostly because the wall of the world had similar features. More objects were placed so that distinct features could be extracted by the RTAB-map ROS package. This helped in creating a good map for the custom world as well.

The map was obtained in two rounds of the robot around the world. Map was formed quickly and more than 20 loop closures were obtained in both the worlds. Performance of

Fig. 9: Custom World - 2D map

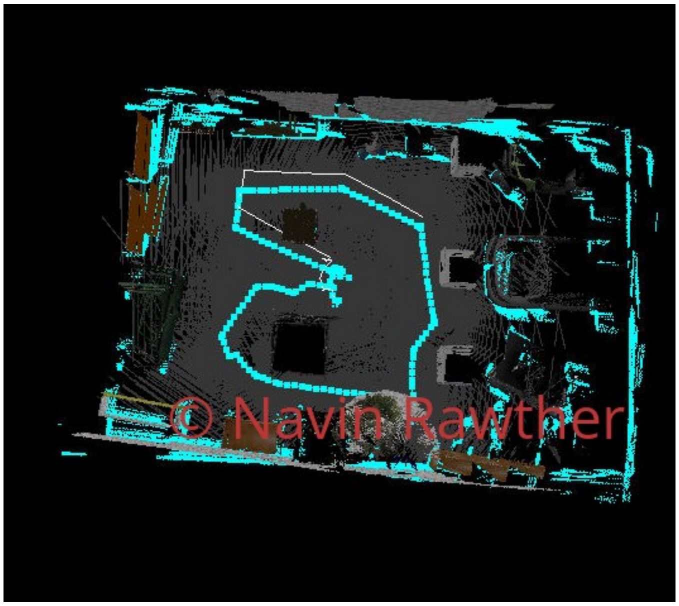

Fig. 10: Custom World - 3D map with trajectories

map creation was difficult in places with identical features.

6 FUTURE WORK

Th implemented SLAM technique is really good for real world robots as it quickly forms the map. Thus, this technique can be implemented in robots that operate in indoor worlds. Different challenges could be examined such as working on real sensors and tf problems. The technique could also be implemented in an outdoor robot to assess the performance in such cases.