Robotics Map My World

Amay Yadav

Abstract—This project is to demonstrate localization for a Robot and a Simultaneous Localization and Mapping (SLAM) algorithm was applied to two simulated environments to generate maps. In the simulation, a mobile robot with a RGBD camera and a laser range finder was used to roam around the environment and map it. The SLAM algorithm used is Real Time Appearance Based Mapping (RTAB-MAP), a type of GraphSLAM algorithm that uses traditional computer vision techniques to solve the correspondence problem i.e. correlate the images to one another in an environment. There are two simulated environments (1) The provided kitchen dining environment and (2) The office environment which has been built for this project specifically. In both environments, the package RTAB-MAP package was utilized to generate the map. This paper presents the findings of running the algorithm and its accuracy and advantages in the real-world.

Index Terms—Robot, Localization, SLAM, GraphSLAM, RTAB-MAP, Robotics Nanodegree, Udacity.

1 Introduction

Mobile Robots need to map their location in reference to their surrounding and in doing so build map of their environment to be effective in tasks like transportation, picking up and moving crates in a warehouse, automated vacuum cleaning or even self-driving cars. The creation of an accurate map by the mobile robot itself allows it be truly independent and it can operate in new and complex environments only based on their on-board sensors. Learning maps under pose uncertainty is often referred to as the Simultaneous Localization and Mapping (SLAM) problem [1].

GraphSLAM is one of the well known SLAM algorithms as the name suggests it is a graph-based formulation. Solving a graph-based SLAM problem involves the construction of a graph whose nodes represent robot poses or landmarks and in which an edge between two nodes encodes a sensor measurement that constrains the connected poses. Once such a graph is constructed, the crucial problem is to find a configuration of the nodes that is maximally consistent with the measurements. It explores how the SLAM methods works in a simulated environment. The robot uses a RGB-D camera and estimates its trajectory and map features poses as it moves through the environment. If the SLAM algorithm is successful, the robot would be able to produce a map with recognizable features in the surroundings and its trajectory.

2 Background

In Simultaneous Localization and Mapping (SLAM) the robot needs to map itself in respect to its surroundings and also map its environment solely based on the sensor inputs which in this case is a RGB-D camera and a laser range finder. Now, when we initially start up the robot we only have some input data and it's easier to map what's right in front of the robot, but once the robot starts moving in some environment like a house it gets a whole lot of sensor input data from different points of view and it needs to coalesce to formulate a 3D view of its surrounding environment.

In this project we are going to use the GraphSLAM algorithm to map the robots' environment in both the gazebo simiulated environments. The GraphSLAM algorithm represents the SLAM problem as a graph where the poses of the trajectory and the measured locations are the nodes of the graph and the estimated motion and measurement distances between these nodes are the edges. The algorithm solves for the best configuration of the graph to satisfy the constraints as much as possible, thus solving the Full SLAM problem [1].

RTAB-Map (Real-Time Appearance-Based Mapping) is a RGB-D, Stereo and Lidar GraphSLAM approach based on an incremental appearance-based loop closure detector. The loop closure detector uses a traditional bag-of-words approach to determinate how likely a new image comes from a previous location or a new location. The RTAB-Map algorithm detects loops in the trajectory with features such as SURF or SIFT. When a loop closure hypothesis is accepted, a new constraint is added to the maps graph, then a graph optimizer minimizes the errors in the map. A memory management approach is used to limit the number of locations used for loop closure detection and graph optimization, so that real-time constraints on large-scale environments are always respected. In this project we have demonstrated the effectiveness of RTAB-Map in mapping the two environments.

The goal of this project is to do the following:

1) Build a localization project to make the necessary changes to interface the robot (mounted with camera and laser range finder) with RTAB-Map.
2) Launch the robot and teleop around the room to generate a proper map of the supplied kitchen dining environment.
3) Rewards R.
4) Build a simulated environment in this case it's an office environment with tables and chairs. Launch the robot and teleop around the room to generate a proper map of the created office environment.

Once we accomplished the above mentioned tasks we should have a map of the environment that we can view using rtabmap-databaseViewer.

Fig. 1. Gazebo model for kitchen scene.

3 Robot Model Configurations

The robot has four wheels and round chassis for easy turning. We can easily teleop with this kind of robot as because it allows us to turn on the spot if we crash into any walls. It is mounted with a kinect RGB-D camera and a laser range finder. The camera and laser are both stacked on top of the chassis at a certain height to have a longer sensing range.

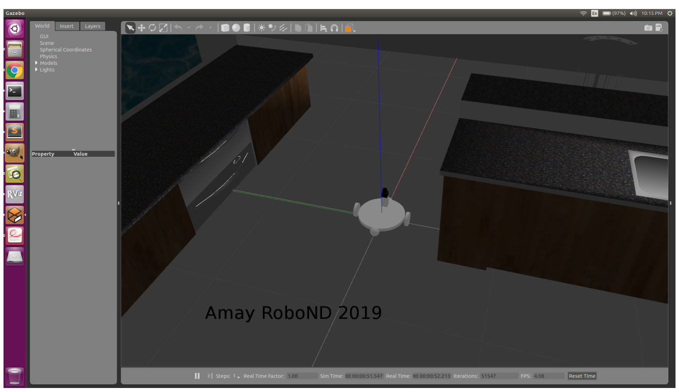

Fig. 2. Robot Model.

4 Scene Creation

The first environment has already been provided; it's a Kitchen dining area and sort of a enclosed room on both sides of the kitchen.

The second environment is an office area with some desks and chairs and tables and a small room and also a whole lot of doors. It is an enclosed space so the robot can map the environment completely.

5 Results

5.1 Kitchen Dining Scene

The generated mapping for the kitchen dining environment is very accurate. The boundaries of the map are properly generated. Most of the surrounding environment is very well mapped and depicted in rtabmap-databaseViewer for

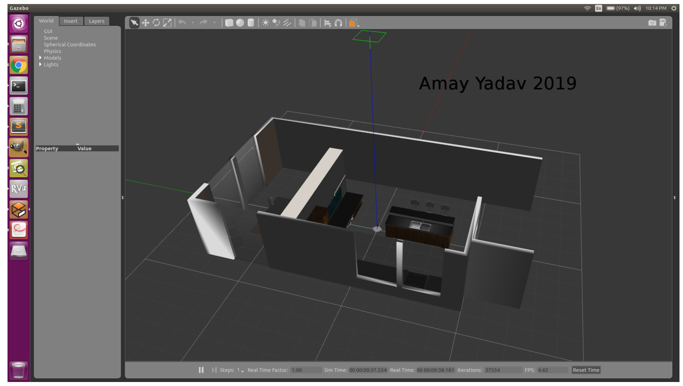

Fig. 3. Kitchen Scene.

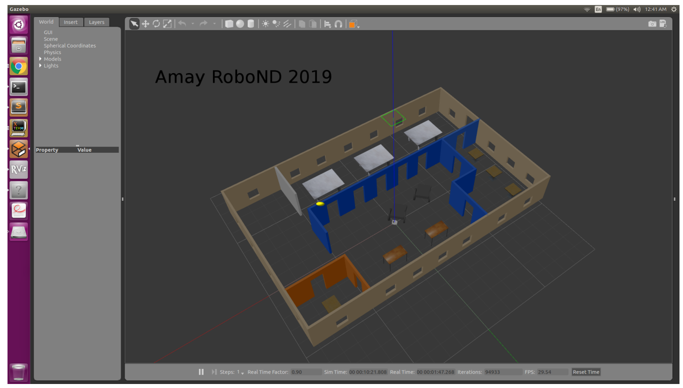

Fig. 4. Office Scene.

instance the walls, chairs, table etc. The robot is inspect elements in the environment from a certain fixed height and so it has crawling toddlers view. The right side being open area does not have straight boundary.

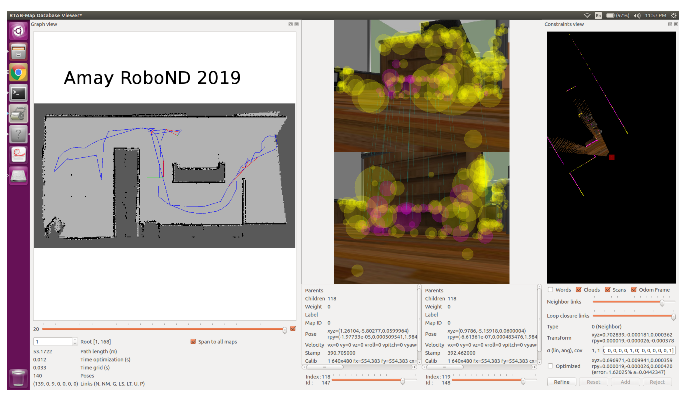

Fig. 5. RTAB-Map of Kitchen Environment.

5.2 Office Scene

The office area depicted in the images above does not have a lot of items in it and the robot has perfectly mapped the closed area and all the doors. The walls are very well covered. Moreover, the robot is able to map the legs of the tables quite effectively because the robot doesn't high enough view. There's two wheelbots at ground level and the robot has perfectly mapped those. The results are pretty good and all of the environment has been covered.

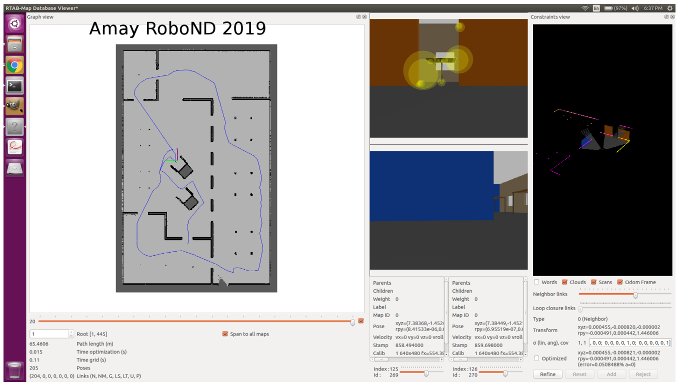

Fig. 6. RTAB-Map of Office Environment.

6 Discussion

The Robot is able to map both the environments quite well. In both of the scenes there were a lot of repeated objects so the robot which could lead to false positive on loop closures. In spite of that the robot was able both the worlds quite well. It was a slow process of teleoping the robot across the environments but after circling around and coming back to the same we had the complete map of the environment. The RTAB-map algorithm can probably tuned or we can preprocess the images to be classified before hand but then that would beat the process of having a robot in an alien environment.

RTAB-map assumes the objects to be fixed in the environment and that's why the results are so good. If we introduce a lot of same moving objects in the environment then RTAB-map will not be able to function as well. That's a pitfall of the algorithm and it'll probably require an object tracking algorithm to keep track of moving objects.

7 Conclusion / Future work

As any animal needs to be aware of their environment to roam around in it freely and do certain tasks. It is equally essentially for a robot to know it's environment and also it's own position in it for e.g. robotcs to shelf and transport crates or even self-driving cars. SLAM is very important for sending a rover to other planets where we won't have any maps available.

For future work, the algorithm can be enhanced to be applied to multiple robots in cohesion so that they can work together for e.g. multiple cranes on wheel in a shipping yard responsible for picking up containers. The algorithm could be modified to have a 3D view of the environment using multiple cameras.

References

[1] M. M. Sebastian Thrun, *The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures*. The International Journal of Robotics Research, vol. 25, no. 5-6, pp. 403429, 2006.