Map My World

Beshari Jamal

Abstract—The purpose of this project is to apply simultaneous localization and mapping principles to localize and make the robot able to generate a map of its environment. The robot is extended from a previous creation made in the localization project. However, this project employs SLAM through Real-Time Appearance-Based Mapping (RTAB-Map). Consequently, the sensors are re-messaged and rewired to provide the required sensor information for RTAB-Map. Sensors incorporate an RGB-D sensor(camera and a depth sensor), 1D scanner and odometry such as IMU and wheel encoders.
The robot is then launched and tele-operated around the place and generate the environment map in 2D and 3D. After that, a personal environment is created and mapped using the same procedure.

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

1 Introduction

Mapping, whether in 2D or 3D could be one of the most useful functionalities of mobile robots. In simple terms, it is localizing outside information to the robot (or each other), the robot then learns its environment that way. While also some robots are provided with maps of their environments, their environment may be altered or rearranged. Then, a robot with the right sensors can map obstacles, free space or any depth and visual feature, and therefore, can also plan a safe trajectory path for navigation, also could compare its local map with a global map to localize itself globally.

In this project, an RTAB-Map (Real-Time Appearance-Based Mapping) is utilized. RTAB-Map is a type of SLAM that utilizes an RGB-D camera sensor and a rangefinder. It also uses appearance-based loop closure detection. The map is then stored in a database that can be inspected by an RTABmap-database viewer.

2 Background

When a robot meets a new setting where there is no provided map, it needs to be able to create this map and localize its pose using it. This methodology is the Simultaneous Localization and Mapping. Various types of mapping algorithms can be implemented. These include Occupancy Grid Mapping, Fast-SLAM, and GraphSLAM. This project focuses on the 3D Graph-based SLAM approach with RTAB-Mapping.

SLAM algorithms are not easy to implement, specifically for 3D mapping. Large amounts of noise may present irreversible error into the data. Noise could be gained from Odometry used to keep track of robots motion or the sensory data. SLAM algorithms also face challenges as the map gets more significant, especially if some parts of the map looks like another part such as in indoor mapping. These challenges could be overcome by proper tuning of the packages' parameters involved. As the map gets larger, the required memory size gets more substantial and additional memory management is needed. Choosing the right SLAM algorithm for the robot purpose mitigate these hurdles.

2.1 Mapping Algorithms

The FastSLAM approach combines SLAM and localization using a custom particle filter approach. Each particle holds the trajectory of the robot and a with map features, these features are estimated by local gaussian. To solve all these independent problems each ending to estimate features of the map, Fast-SLAM finally uses low dimensional Extended Kalman Filters to localize the robot and map the environment efficiently. The main problem of FastSLAM is it assumes there is a landmark. Consequently, it is unable to model an arbitrary environment. However, Grid-Based Fast-Slam combines both Occupancy Grid Mapping and FastSLAM to model the environment without predefining any landmark, thus, mapping any arbitrary environment.

Graph-based SLAM solves the full SLAM problem by recovering the entire path and map and uses all information recorded. This approach allows it to consider dependencies between the current and previous poses. A significant benefit of this method is accuracy and the significantly reduced on-board processing. Graph-SLAM achieves that by following a (visual-based) graph-based approach to represent poses, features from the environment, the corresponding probabilistic motion, and measurement constraints. Then it tries to solve for the uncertainty of those probabilities in the map.

2.1.1 RTAB-Mapping

RTAB-Map is an RGB-D Graph-Based SLAM approach based on an incremental visual-based loop closure detector. A loop closure happens when Robot revisits a location and recognizes a feature or a landmark, then readjusts the map accordingly, when readjusting the map, the robot is solving for the probability uncertainties in its previous movements. The loop closure detector uses a bag-of-words feature detecting approach to determinate how possible a new perception comes from a previous location that has been visited, or a new location. When a loop closure proposal is accepted, a new graph constraint is added to the maps visual graph; then a graph optimizer goes to minimize the error in the map. The loop closure occurs in real-time, so as the map largens, a memory management approach is applied to limit the number of locations used for loop closure detection and graph optimization, Prioritizing real-time constraints.

3 Configurations

3.1 Robot Configuration

The robot model was expanded from the localization project and had its camera swapped with an RGB-D camera. It is as seen on figure 1, with two driving wheels on the sides, two caster wheels at the front and the back of the robot. A range sensor and the RGB-D camera are in the middle of the front half of the robot. The robot is small, light, a little more than 16 kg and has a broad base with a footprint radius of 0.2 m and 0.06 for wheel radii.

It is worth noting that when installing the RGB-D camera, the depth sensor part of the camera needed a re-orientation as the initial orientation -which was not the same as the camera orientation- did satisfy the scanning needs and was pointing upwards. A transform as shown in 2 transform was applied to correct that.

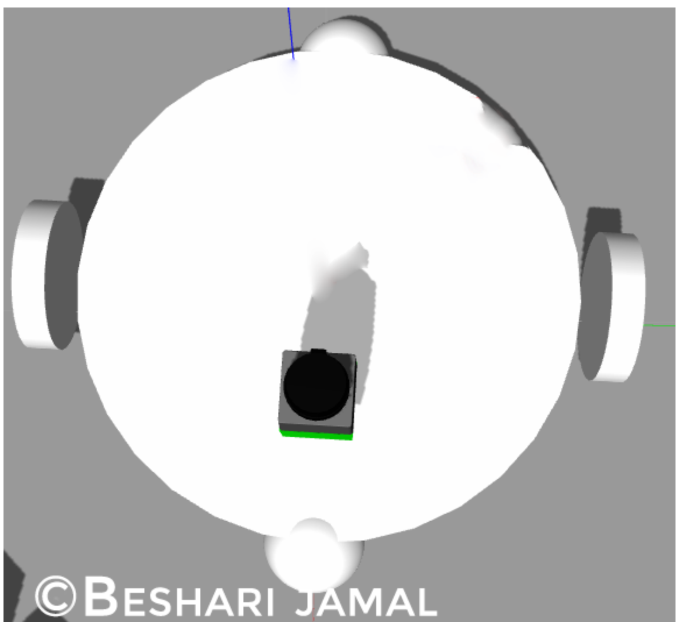

Fig. 1. Personal Model

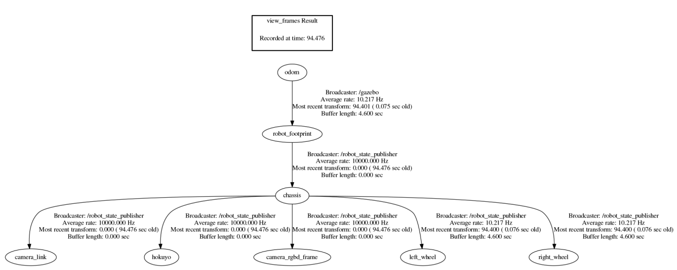

Fig. 2. Transform frames between the different robot links

3.2 Scene Configuration

The provided scene is a kitchen Dining space with an extra room as shown in figure 3. The apartment had two different spaces divided by semi-wall, one included the kitchen and the other a seating area.

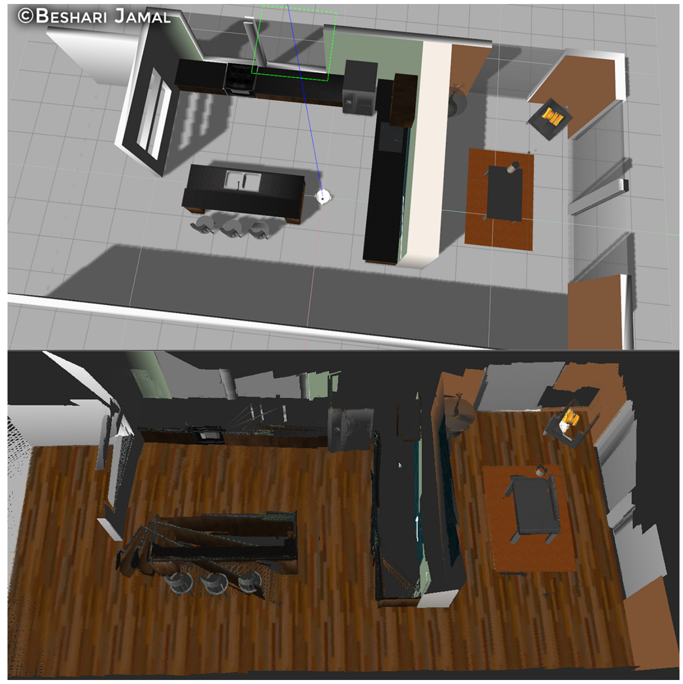

Fig. 3. Top: Kitchen Environment, Bottom: Mapped environment

The custom world consists of a rectangular room with 17 unique objects inside. The objects are aligned along the walls, to make the different wall-sides identifiable to the robot. Therefore the robot can apply the right loop closures as it maps the area.

4 Results

The final 2D maps results can be seen in figures 5 and 6. The 3D maps are in figures 3 and 4.

5 Discussion

The robot mapping performances were excellent. In mapping the kitchen dining area, the robot succeeded to map all the walls, the tables, the chairs and some more features, the robot only missed a spot under a table, and it erred in mapping the table. This is because the robot was driven mistakingly towards the wall due to the slow computer processing speed, crashed and rolled over, yet the generated map had 21 loop closures. In mapping the personal map, different items had to be placed successively to test whether the robot gets the right loop closures. Beside some unscanned tiny areas behind the barriers, everything was scanned correctly with more than 60 loop closures

The real-time performance of the mapping was varied. This was due to the difference between the two environments' size, mapping area, and the number of features. The mapping of the personal environment was distinctly faster.

6 Conclusion / Future work

Mapping is a critical function in many robotics applications. Different robots can use mapping for different purposes.

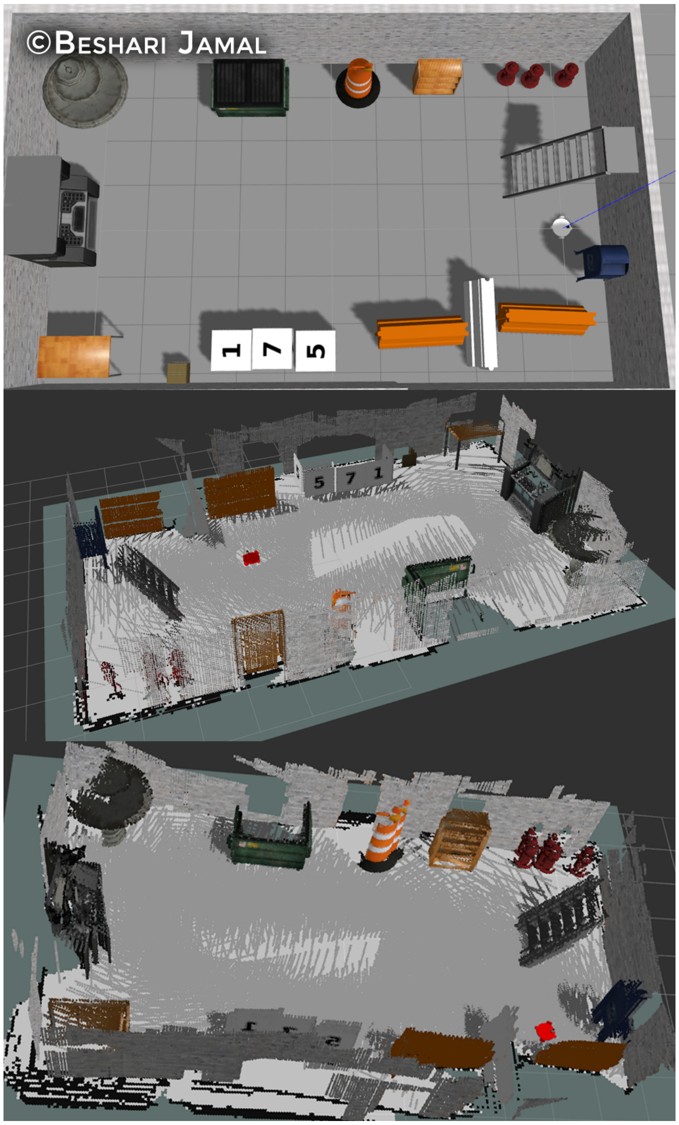

Fig. 4. Top: Personal Environment, Bottom: Mapped environment with one and loop and two loops around the environment

A drone can map an environment for, navigation, data collection, and analysis. Such drones as Kespry, are currently being used in terrain mapping in the construction and mining industries. An autonomous robot can apply mapping to determine how to navigate. Robots also implement mapping when trying to perform pick-and-place operations.

Conclusively, The ROS packages: Teleop, RTAB-Map, were coded and generated, and all nodes were wired correctly, successfully performing 2D and 3D mapping using RTAB-Map. The robot was able to generate characteristic maps of the kitchen dining room and the personal custom world. However, There is always room for growth. Further Improvements includes implementing a more robust SLAM algorithm and fine-tuning the RTAB-Map parameters to be able to recognize loop closures with high regards to odometry data. Further work can include multi-robot autonomous localization, mapping, and inspection.

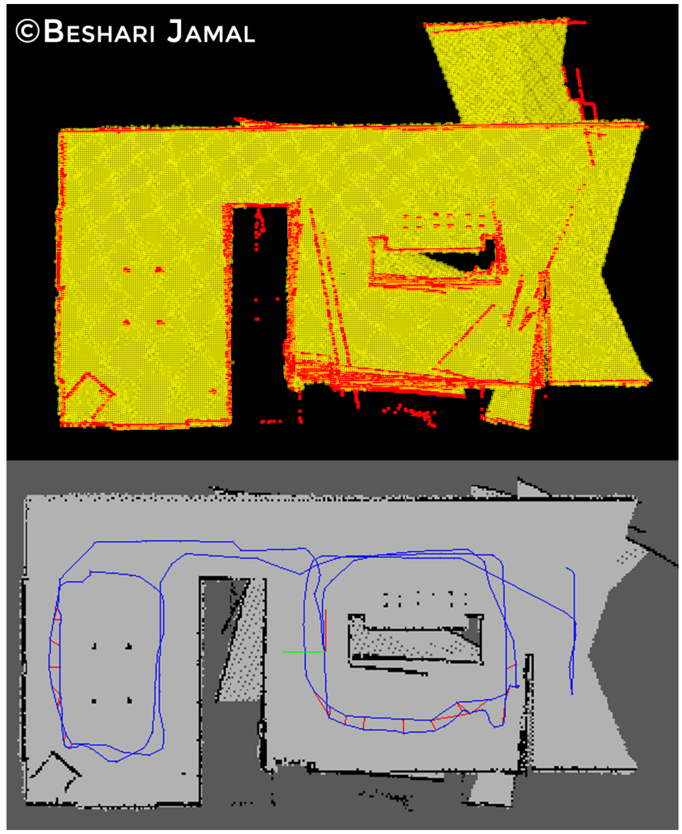

Fig. 5. Two figures showing the mapping process and the final occupancy grid map for mapping the kitchen_dining map

REFERENCES

[1] Udacity, 2017 *Robotics Software Engineer Nanodegree Program: Term2: SLAM*. https://www.udacity.com/ Accessed: June 18, 2018.

[2] Real-Time Appearance-Based Mapping, 2017 *ROS: History*. http://introlab.github.io/rtabmap/ Accessed: June 18, 2018.

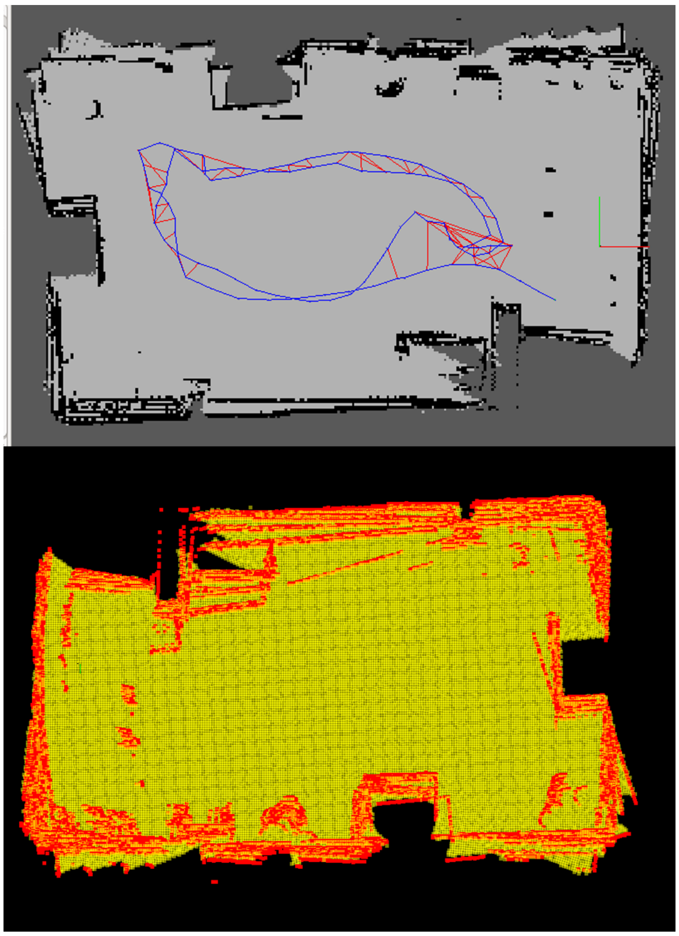

Fig. 6. Two figures showing the mapping process and the final occupancy grid map for mapping the personal map