Map My World

Snigdha Dongre

Abstract – In this project we create a 2D occupancy grid and 3D octomap from a provided simulated environment. Furthermore, we then create our own simulated environment to map as well. For this project we used the rtabmap_ros package, which is a ROS wrapper (API) for interacting with rtabmap. The robot uses Real-Time-Appearance-Based Mapping (RTAB-Map) GraphSLAM algorithm to map its environment.

F

1 INTRODUCTION

Mapping is an essential component of developing robots that can navigate in environments that they have never seen before. Effective mapping allows the robot to dynamically detect different environments and enable further object detection and can also be used to navigate the robot autonomously. Mapping is critical in environments which are unexplored or where the surrounding change often. To create map of an environment successfully, the robot needs to solve the problem of localization (estimating the position of robot relative to map) and mapping (construct a map of the environment given robot's position). In this project we use a Simultaneous Localization and Mapping (SLAM) based algorithm to construct a map.

2 BACKGROUND

As a map is required for localization but robot's location data is required to construct a map, a SLAM based approach is often considered to be a chicken and egg problem. SLAM algorithms can be divided into 5 different categories:

- Extended Kalman Filter SLAM (EKF)
- Sparse Extended Information Filter (SEIF)
- Extended Information Form (EIF)
- FastSLAM
- GraphSLAM

For this project we consider using either a Grid Based FastSLAM algorithm or a GraphSLAM algorithm.

Grid Based FastSLAM: This algorithm uses occupancy grid mapping (2D algorithm in which each grid cell is identified as Undiscovered Zone, Free Zone or Occupied) and FastSLAM which uses a custom particle filter along with a low dimensional Extended Kalman Filter.

GraphSLAM: This algorithm uses a graph-based approach to represent position, features and constraints. GraphSLAM solves the full SLAM problem by recovering the entire path and map.

- GraphSLAM generally provides better result as compared to FastSLAM as GraphSLAM is able to retain information from past locations. GraphSLAM also improves upon the need of onboard processing capability. Therefore, for this project we have decided to use RTAB-Map (Real-Time Appearance-Based Mapping), which is a Graph-SLAM approach that uses loop closure with Visual Bag-of-Words for optimization. This process checks if a robot has seen a location before and uses that to adjust the map based on the path loop completed.

3 SCENE AND ROBOT CONFIGURATION

Kitchen - Dining Scene:

This scene was provided and was used to test the mapping algorithm. This scene is feature rich and has a kitchen dining table with chairs, a table and storage cabinet etc.

Fig 3.1 Kitchen dining world

My World Scene:

This is a custom world which was created by me in gazebo. This world has a walking man,2 tables and a few cupboards etc. Ample space was provided for efficient navigation.

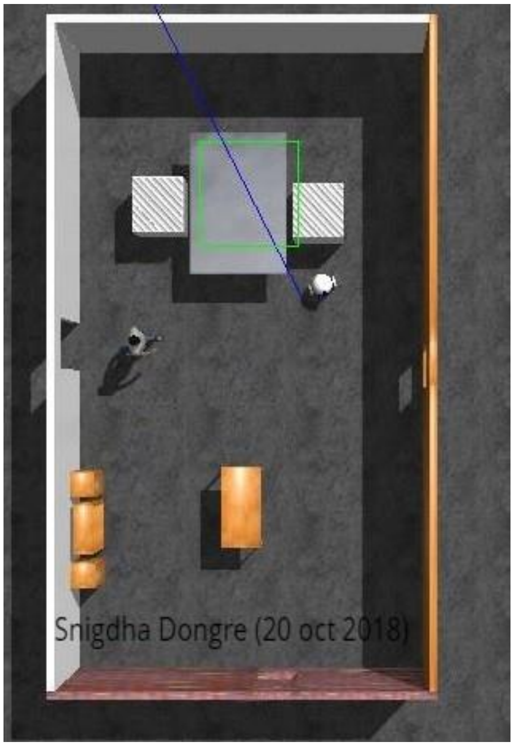

Fig 3.2 My world

Robot Configuration:

The robot model used in the previous project was modified with an RGB-D camera. A transform node was created to allow the rotation and translation of RGB-D camera to improve mapping .TF frame for the robot model can be seen below.

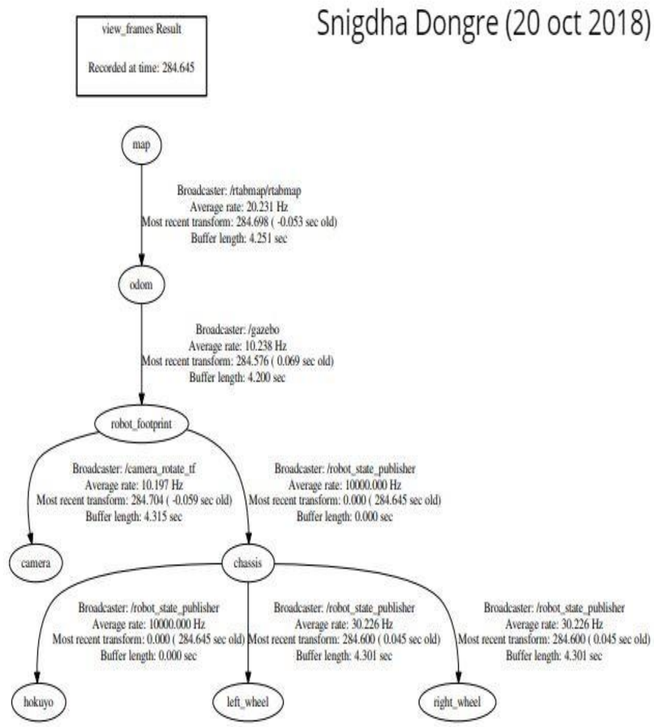

Fig 3.3 TF tree

4 RESULTS

Kitchen - Dining Scene:

Kitchen-Dining world simulation is launched in gazebo. The robot is navigated in the world to create a 2D and a 3D map.

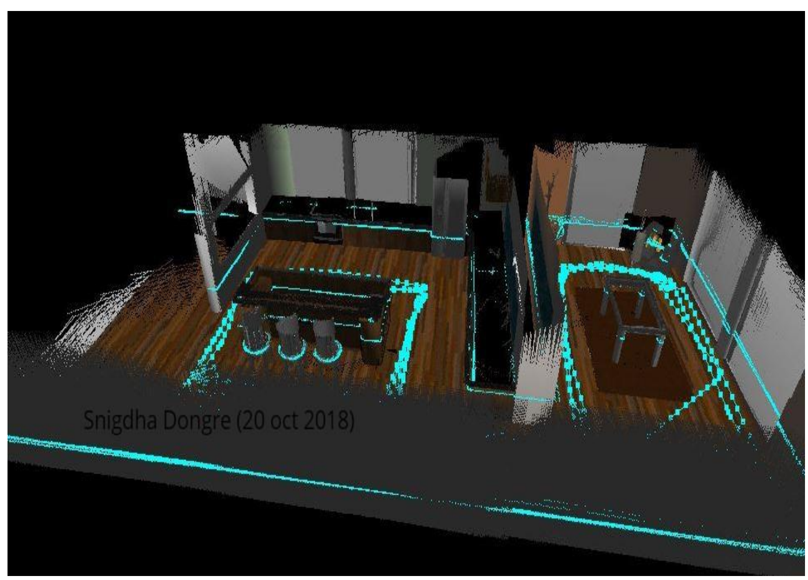

Fig 4.1 kitchen dining 3D Map

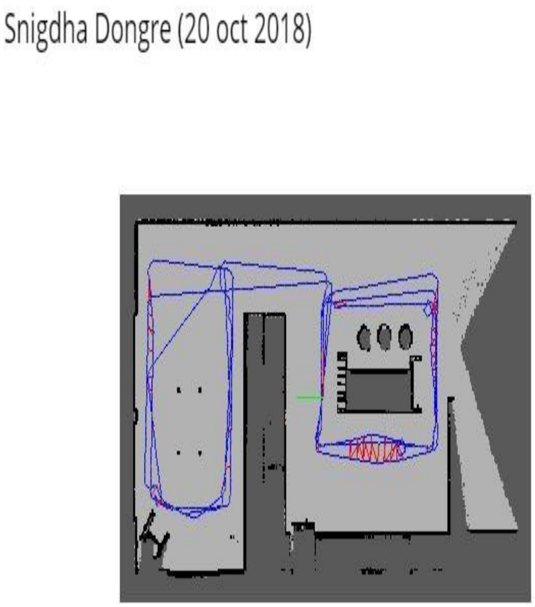

Fig 4.2 kitchen dining world grid map

My World Scene:

My custom My_world simulation is launched in gazebo. The robot is navigated in the world to create a 2D and a 3D map.

Fig 4.3 My world 3D map

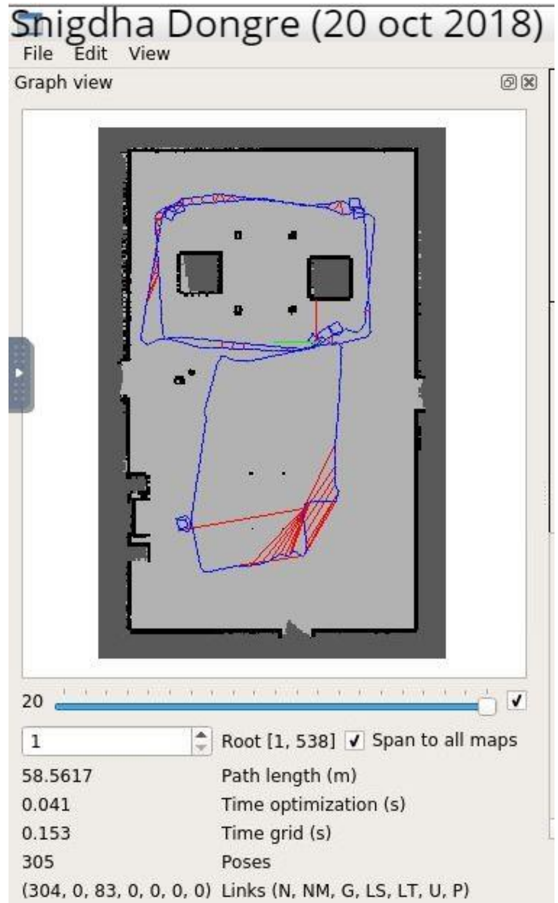

Fig 4.4 My world grid map

5 DISCUSSION

The robot was navigated in both the worlds. The robot was able to map the 2D boundary accurately. The robot was also able to map both the worlds in 3D; however, it struggled a bit to map the tables properly due to the low height of the RGB-D camera. The kitchen dining world mapping was more accurate as compared to the my_world custom world possibly due to more features in the kitchen dining world. More space in the custom world also seems to have complicated the mapping process. The real time performance of the mapping was also inconsistent at times as the path taken and distance from the objects greatly affected loop closures. Sometimes coming too close to the object also delocalized the robot.

6 CONCLUSION / FUTURE WORK

The robot was successfully able to create good quality maps using the RGB-D camera. However, the quality of mapping can certainly be increased by adding more sensors. There is lot of room for improvement as we can add features like visual odometry, obstacle detection, wifi signal strength mapping, finding objects in 2D and 3D and Multisession mapping.

REFERENCES

[1] Udacity, Robotics software engineering nanodegree, Map My World Project