MAP MY WORLD PROJECT

POORNIMA L NATHAN

Table of Contents

Abstract

This project aims at implementing Simultaneous Localization and Mapping to a robot that has been previously developed. A 3D map of the environment that the robot has been run is to be generated. Also, the same robot, developed earlier, is run in a new environment to discuss its effectiveness in mapping. The robot is additionally equipped with a new RGBD sensor to enable it to generate the 3D map based on the reading from the new sensor.

Introduction

The previous project involved the robot localizing itself in a known environment. But the reality is that no environment is going to be the same. In due course, there will be changes occurring in the environment that will need to be accounted for by the robot.

In this project, the robot is let to perform simultaneous localization and mapping in an environment that is not yet known to the robot. This then creates another map in occupancy grid and and OctoMap using the SLAM algorithm.

Background

Mapping your environment is the next logical step after your robot localizes itself. This project is about helping the agent understand its environment and map them so it is aware of its surroundings. This will also address the kidnapped robot problem. Addressing the problem of the robot mapping its environment and localizing itself can be covered in 5 types of algorithms:

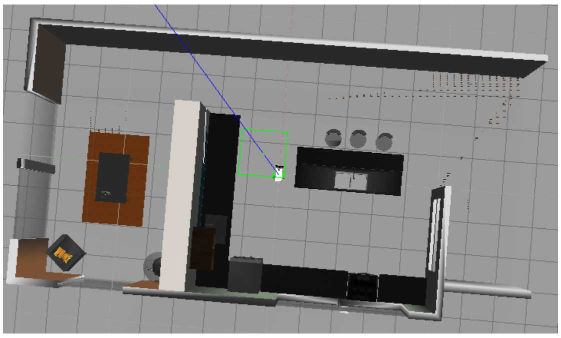
- Extended Kalman Filter
- Sparse Extended Information Filter
- Extended Information Filter
- FastSLAM
- GraphSLAM

The FastSLAM algorithm uses the particle filter approach with a low dimensional extended Kalman filter to solve the SLAM problem. Real Time Appearance Based Mapping is used to help us with this problem. Mapping the environment poses two different challenges –

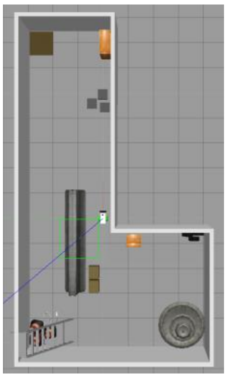
- Unknown map and poses
- Huge hypothesis space

Scene and Robot Configuration

The initial world that the robot needed to be run into was provided in the student files. Changes were made to the existing localization project to the launch file to open the robot in this world.



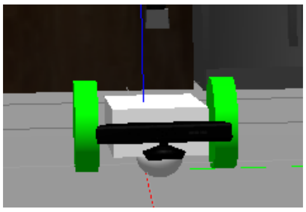
Post this, as the second part, a new Gazebo environment was built from scratch. At first 4 walls were built using the building editor. This was then opened in the model editor. Then random items were added from the Gazebo model database. The resultant was a garage-like setting that had items in it.



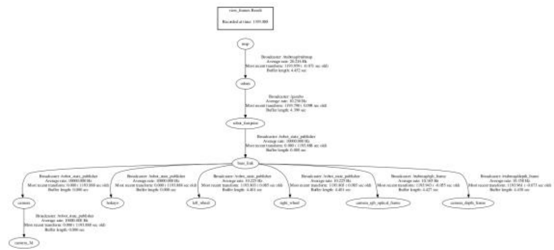
This design was implemented to better understand the SLAM algorithm. We will need to understand if it understands the 'L' shape of the environment as it is or as a loop. If you notice closely, a long table is placed strategically in the middle of the garage to enable a loop-like setting.

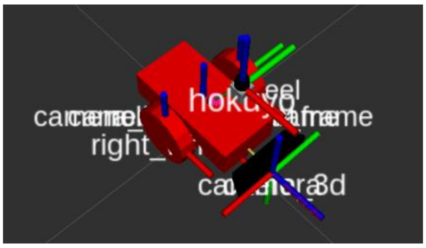
Robot Configuration

This is the same model used in the localization project for all practical intents and purposes. Also, the classical camera is replaced with an RGBD sensor.



The transform tree is used to determine of the linking of the topics for each sensor is done correctly. The following diagram confirms the same.

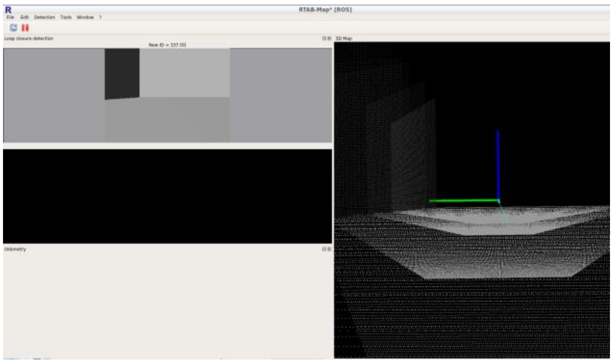




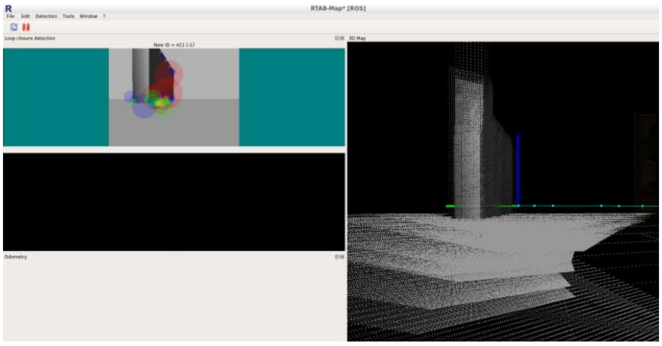
Results

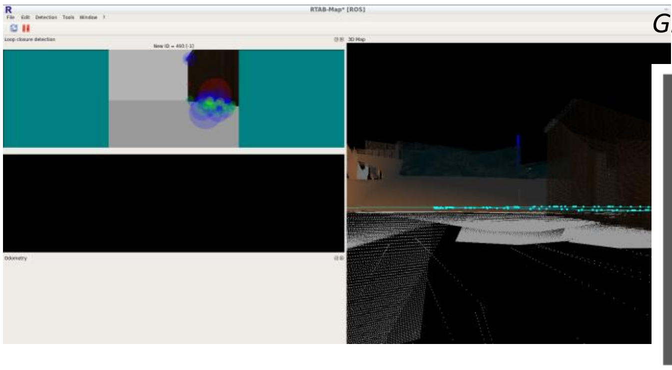
The following pictures clearly depict the way the SLAM algorithm works. The more features that the camera encounters, the easier it is for the robot to detect the environment and identify loop closures.

Loop/feature detection with almost nothing in the camera's view

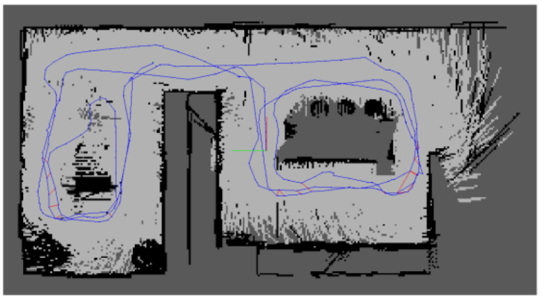


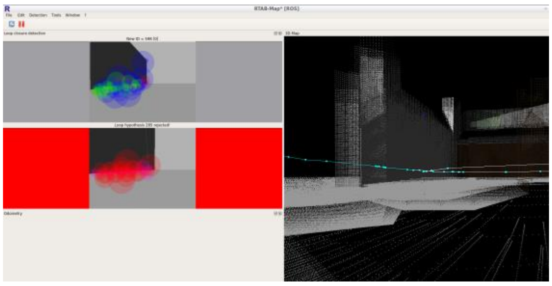
Loop/Feature Detection With Just Walls In The Camera's View

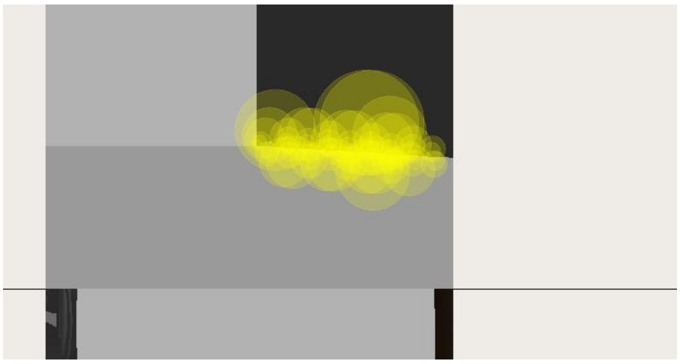


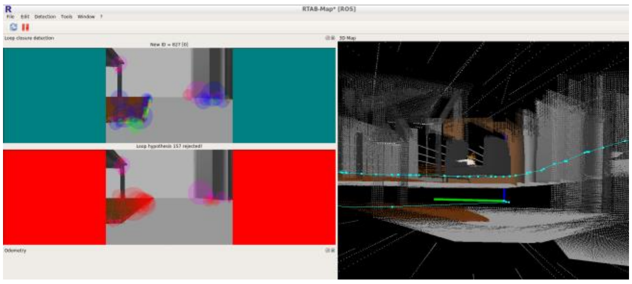


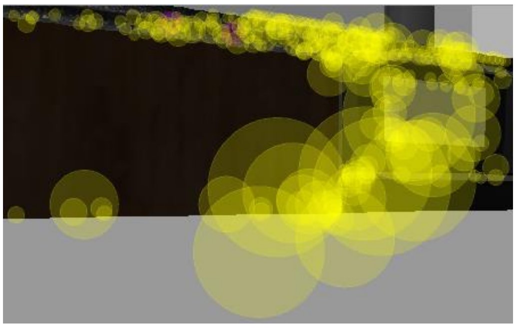
Graph View

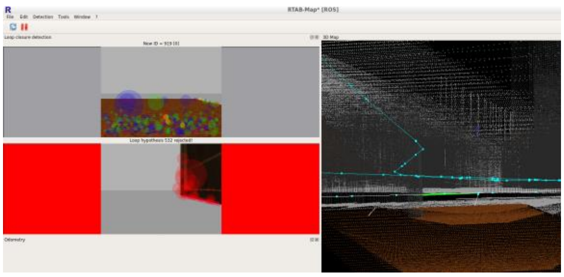


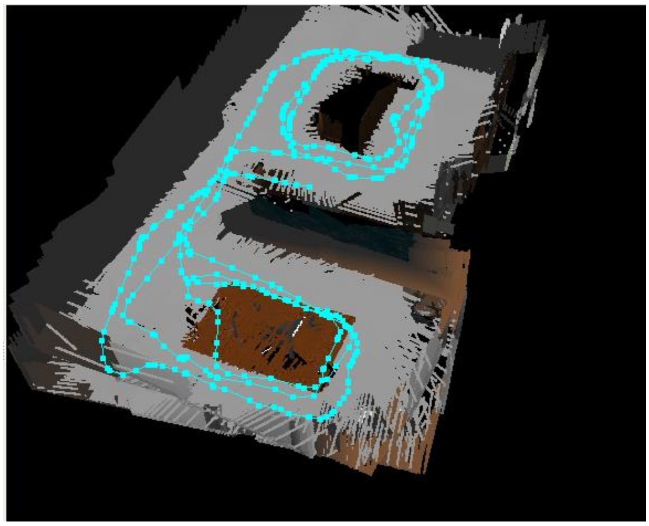


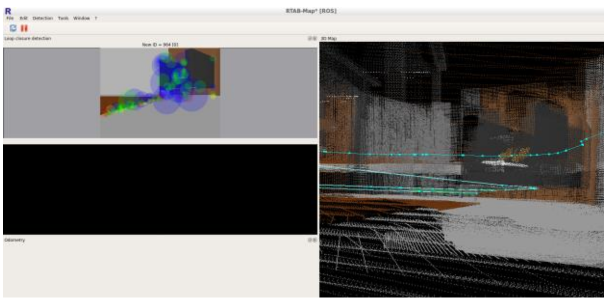


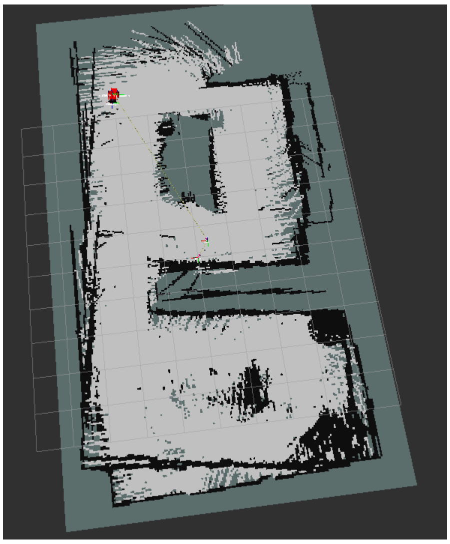


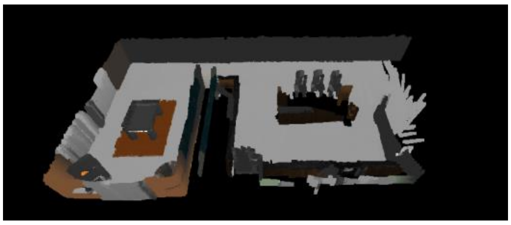








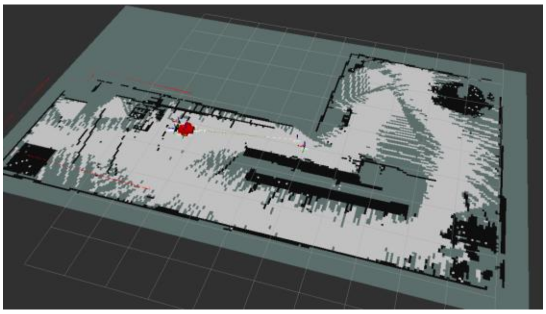




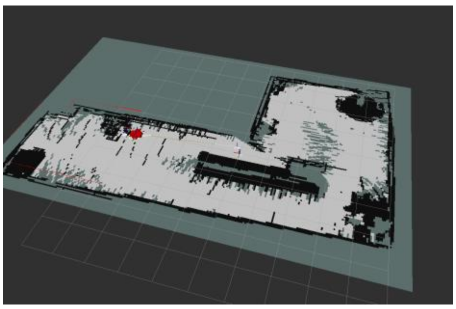
This behaviour was even more prominent in the new environment where there are a lot of items placed such that the robot can spot them easily for feature detection purposes.

Map – New Environment

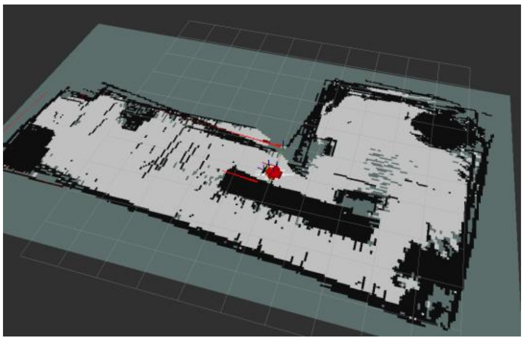
Loop 1



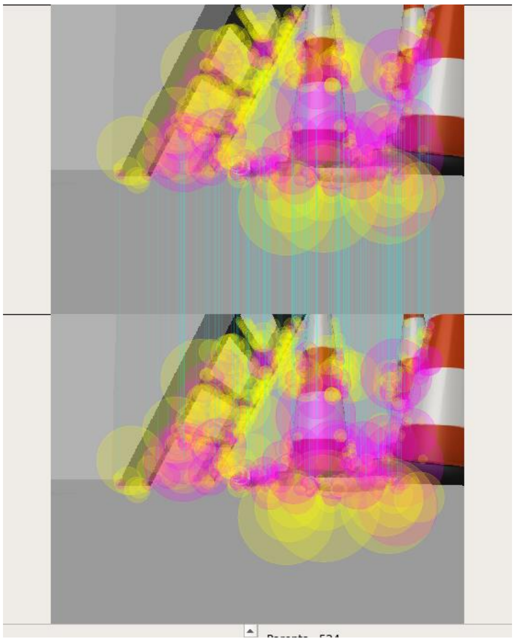
Loop 2

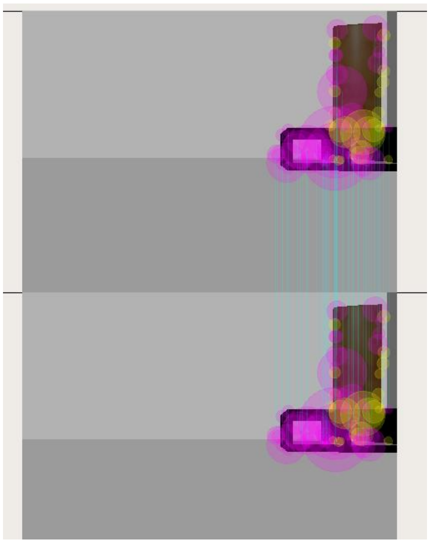


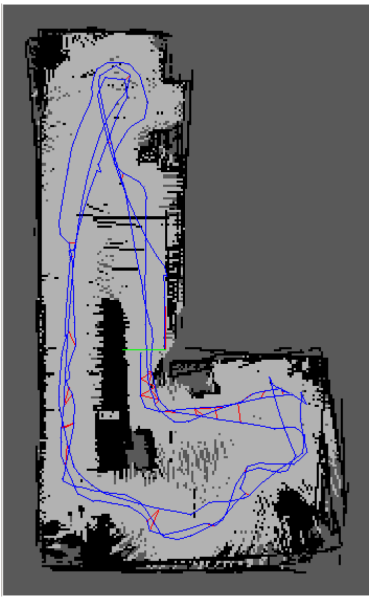
Loop 3



Feature Detection – New Environment







Discussions

The robot was able to map both the environments successfully. One interesting observation was that the robot was able to capture only objects in its line of vision. This meant objects that were the height of the robot. This behaviour leads us to believe that features that don’t qualify this requirement are not used to define features or loops. For example a clock on the wall. Also, the bot needed to moved at a slow speed for it to capture all the features in enough detail. If it is moved at a high speed, not only does the robot find it difficult to capture features, but also, features that are very similar, like the dining chairs on the opposite ends of the table, are considered for loop closure. If you notice in one of the pictures above, the camera detects a lot of features on the carpet as well, which helped localize the robot easily around that area. It is also evident that more loops translated to a clearer map, this could also be overcome by increasing the range of the laser rangefinder and fitting a camera with a better resolution. Nevertheless, the robot was able to map both the environments successfully.

Future Work

SLAM has so many applications in robotics from a Roomba to defence applications. I personally find this very useful for search and rescue applications. It can be extended to swarm robotics where a group of robots can coordinate with each other to search, locate people/property that needs to be rescued and navigate back to home base.