Map My World

Diana Baicu

Abstract - In this paper, the process of applying a Simultaneous Localization And Mapping algorithm to a mobile robot in a simulated environment is presented. The implemented approach is based on the Real-Time Appearance-Based Mapping technique, through which both the 2D and 3D maps of the environments are obtained. The maps are constructed through correlating the visual data from an RGBD camera, along with data from a laser scanner. Two types of environments are mapped, a smaller one, with plenty of distinct features, and a larger, rather repetitive one, the results being discussed in the following sections. The implementation is built on ROS, employing the visualization and debugging tools.

1 Introduction

Given that the last project has been focused on performing localization on a provided map, this time the objective is to develop a more versatile application for a mobile robot, in a real-world scenario it having to identify the position and be able to create a representation of the environment by its own. This scenario would be necessary even for situations where an approximate model (or plan) of the building is present, since in a dynamic environment, inhabited by people, objects are unlikely to keep their position or appearance observed at a certain point for a long period of time.

The used robot for this task is represented by a small-dimensions wheelchair, which first tries to explore a kitchen and dining room environment, and then a custom working space for robotics development.

In order to carry the "cartographic" task, the RTAB-Map SLAM algorithm will be deployed, the robot navigating the spaces through manual commands. The outputs of interest are the 2D occupancy grid and the 3D octomap (which can be consequently transformed to a 3D point cloud).

2 Background

SLAM or Simultaneous Localization and Mapping is meant to solve two of the most fundamental problems of a mobile robot that navigates in unfamiliar environments, namely constructing a map of its surroundings and localizing itself with respect to that map. It is implied that both the map and the robot's poses are unknown. At the same time, it is assumed that the measurements taken are characterized by noise.

There are various problems one has to face when trying to obtain a map, starting with the repetitive nature of places, which produces perceptual ambiguity. Thus, it cannot be clearly stated which location some features belong to. At the same time, the size of the environment to be mapped can raise difficulties, especially if it exceeds the perceptual space of the robot, but also due to the large amount of data that needs to be saved on board. In the mapping process, odometry information also plays an important role for the representation of the robot's path, but error inevitably accumulates over time, situation which is particularly problematic for travelling in cycles. The error arises from the fact that the point where the loop has been started is highly unlikely to be similar or very close to the ending point.

There are various alternative algorithms for mapping, starting with the Occupancy Grid Algorithm, to SLAM version, such as FastSLAM, Grid-based FastSLAM and GraphSLAM. In a broad sense, there are two types of SLAM algorithms, online SLAM, which produces the instantenous map and pose, and the full SLAM, which keeps track of the poses and constructs the entire map, the three mentioned methods all solving the full SLAM problem. It is to be mentioned that all SLAM algorithms take into account the movement commands.

2.1 Occupancy Grid Algorithm

The Occupancy Grid Algorithm is intended to construct a better map by knowing the poses of the robot and, moreover, assuming that they are non-noisy. It is common for the mapping phase to happen after SLAM, in the form of a post-processing step. It uses the robot poses filtered from SLAM. It is mainly used for 2D maps, due to the fact that the computational needs are much higher when it comes to 3D spaces. Thus, it relies on uniformly partitioned 2D spaces, which encodes the binary information of the cell being occupied or free. In order to estimate the occupancy of each cell, it uses a binary Bayes filter.

2.2 FastSLAM

FastSLAM solves the full SLAM problem and also the correspondences are know, which means that the robot know whether it has been in the same location before. It is comprised of two steps, first a trajectory estimation, which uses a particle filter, such as the Monte Carlo method, and consequently a map estimation, where a low dimensional Kalman Filter for the independent features of the map is used, each feature being characterized by a local Gaussian.

2.3 Grid-based FastSLAM

Grid-based FastSLAM is also a full SLAM algorithm, which combines the FastSLAM to the grid occupancy mapping. It is able to recover the full path and the complete map by adding a few more steps accounting for the grid approach. It first of all samples the pose of the robot, knowing the current control commands as well, then uses the Occupancy grid mapping presented above to estimate the map, to finally attributing importance weights given the current pose and map estimation.

2.4 GraphSLAM

In GraphSLAM, the robot's poses and its environment are represented using graphs. The graph is constructed using the recorded odometry and the measurements from the on-board sensors, step which is referred to as the front-end of the algorithm. Meanwhile, the back-end is responsible to outputting the most likely configuration of the robot's poses and of the map with its features, taking as an input the constructed graph with its constraints. Is also includes an optimization step in order to find the smallest error of the configuration as described by the constraints. Since the constraints are non-linear, they must first be linearized before using the information, for the linear approximation a Taylor series being employed. Is an iterative algorithm, which uses the solution of the previous step, meaning the best approximation for the poses and the map features, as an estimate for the current step. The solution converges after a few iterations.

RTAB-Map, or Real Time Appearance Based Mapping is based on the vision information taken from cameras to construct the map. The loop closure procedure, through which it is determined whether the robot has visited or not a certain location before, is used. It can find a correspondence between the current location and a previously visited one by applying the so-called bag-of-words approach. Firstly, the features from an image are

extracted by a feature descriptor, such as SURF. The features are then organized into vocabularies, a vocabulary clusters similar features, or synonyms. Matching a feature to a vocabulary is achieved through quantization.

For the memory management aspect, RTAB-Map makes the distinction between the Working Memory, with the most recent and frequently observed locations, the Long Term Memory, where all the other locations are transferred, and the Short Term Memory, where the node for the current acquired image is constructed.

3 Scene and robot configuration

The constructed scene was indented to recreate a larger environment compared to the provided kitchen and dining room one, in order to test the mapping behavior under different conditions. Thus, the created world represents a working space with a few robots and parts for robots, along with multiple tables and shelves, as it can be seen in the image below. Mainly, the objects in the environment have been placed marginally, leaving a larger space towards the middle of the room.

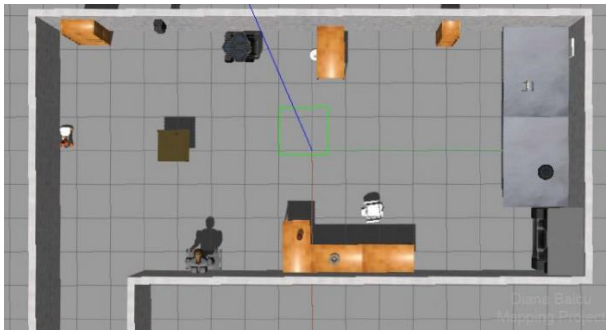


Figure 1. Custom world – lab 1

Due to the fact that the mapping process has proved to be problematic for the created world, as it will be described in the section below, another simplified world has also been attempted. It was meant to have smaller dimensions than those of the previously created world, and also provide a more feature-rich environment. The top view of the second custom world can be seen in Figure 2.

Figure 2. Custom world – lab 2

Regarding the custom robot, the model for the localization project has been further used and modified. The normal camera has been replaced with an RGBD camera, which has been kept at the lower side of the wheelchair platform. As well, a LIDAR has been placed on the feet support.

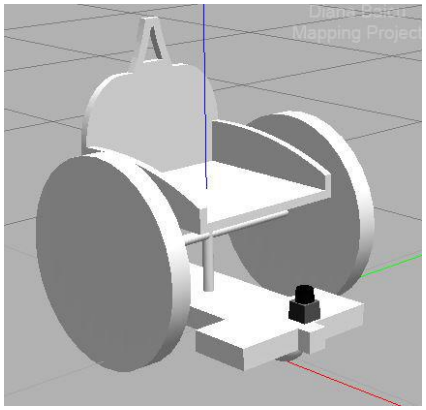
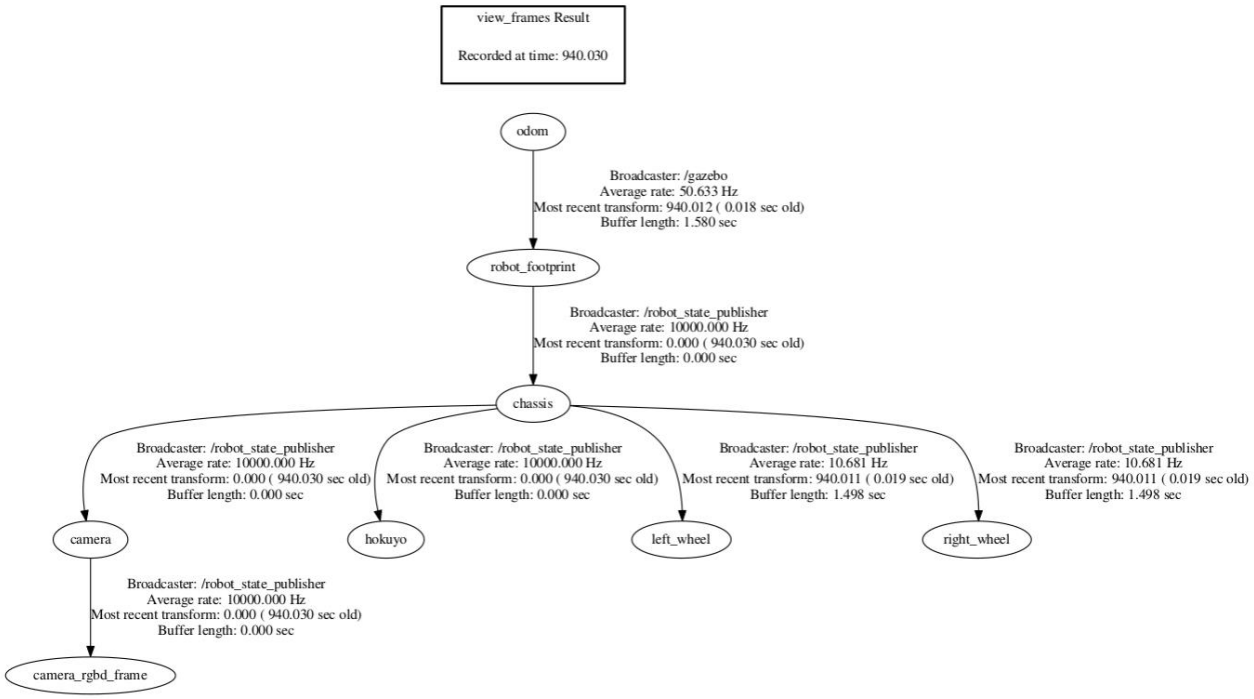


Figure 3. Robot model

While teleoperating the robot through the environment it has been observed that a significant slide either to the left or to the right was accumulating while moving forward, problem which has been ameliorated by analyzing the model's collision, inertia and contacts visuals in Gazebo and consequently modifying the robot's URDF.

The transform link tree is as in the image below:



4 Results

Both the 2D and the 3D maps of the Kitchen and Dining room environment have been well constructed, the pictures below showing the initial world (Figure 4), the obtained 2D map (Figure 5) and the octomap converted to a point cloud (Figure 6).

Figure 4. Kitchen and dining room world

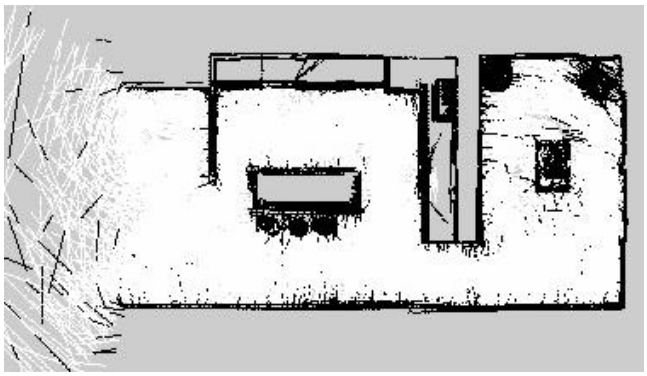


Figure 5. Kitchen and dining room obtained 2D map

Figure 6. Kitchen and dining 3D point cloud

The mapping for the larger environment has arisen several difficulties, due to the fact that after a navigating only a

part of the environment, the map would distort at some arbitrary point, obtaining a representation as in Figure 7. After the distortion point, the map could not have been correctly recovered. What most likely happens is that false positive loop closures might be identified while navigating.

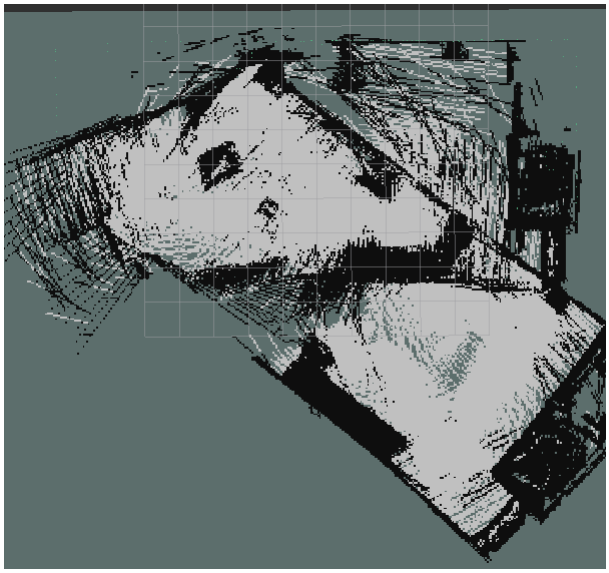


Figure 7. Drifting in lab 1 world

Three main solutions for this behavior have been discovered:

- Setting the "RGBD/OptimizeMaxError" parameter to 0.1 instead of the default 1.0 value. This parameter represents the optimization error ratio threshold above which a loop closure is rejected and helps in the dectection of whether a wrong loop closure has been added to the graph.
- Synchronizing the update frequencies of all the important topics for mapping. /scan, /camera, /odometry and also the detection rate of RTAB-Map have been set to 10Hz.
- Reducing or increasing the *queue_size* parameter, depending on the environment. Thus, for the smaller world, a queue size of 100 produces good mapping results, while for the larger world a queue size of 50 has proven to give better result.

The outputs both worlds are presented in the pictures below. As it can be observed, the accuracy is lower compared to the provided environment, especially for the 3D representations.

Figure 8. Lab 1 world obtained point cloud

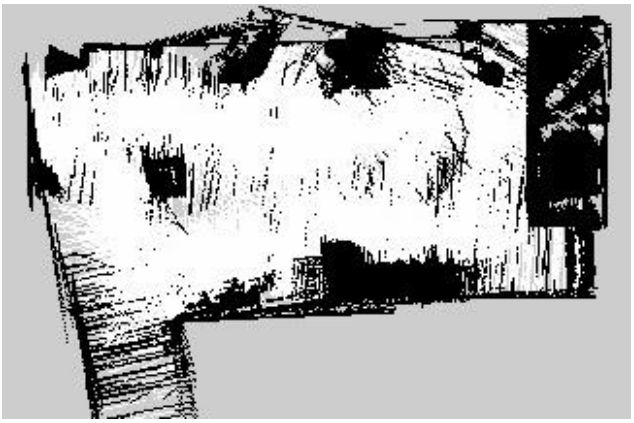


Figure 9. Lab 1 obtained 2D map

Figure 10. Lab 2 obtained 3D point cloud

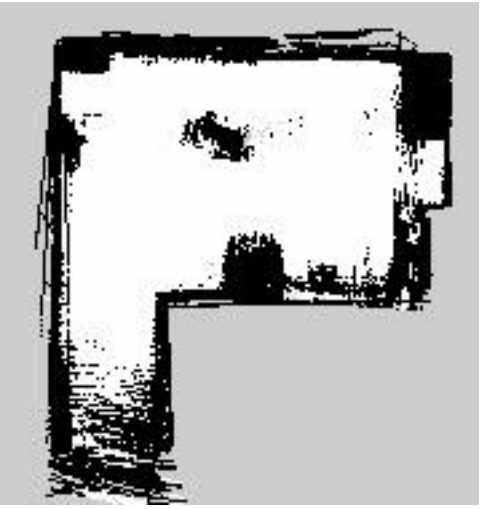


Figure 11. Lab 2 obtained 2D map

At the same time, the map of the first world is less accurate in terms of geometry and was circled only once, whereas the smaller room has been circled three times and on-the-spot turns were performed for obtaining a better view.

5 Discussion

The difference in result and mapping parameters tuning effort when it comes to the provided map and the custom created ones are most likely owed to the features that each type of environment is composed of. Whereas the first environment contains distinct pieces of furniture, windows and other objects, the larger custom environment has mostly simple walls and wide, repetitive spaces between objects. Thus, the loop closure procedure might have a difficult task in finding correspondences.

6 Conclusion. Future work

The RTAB-Map algorithm can be applied to a real robot by being deployed on a platform such as Jeston TX2. In order to construct a map, additional components are required, such as an RGBD camera and a Lidar, but the odometry information of the robot should also be known. The map obtained from RTAB-Map can also be obtained and then postprocessed so that it can then be used a more exact representation of a real-world scenario.

I am interested in performing SLAM in indoor environments, where the accuracy and efficiency of the navigation depends on the map provided. However, I believe that modifying the parameters of RTAB-Map is necessary in order to obtaine a quality representation.