Map My World : Performing SLAM with ROS

Vijayasri Iyer

Abstract—This paper presents an attempt to solve the robot navigation problem by using the popular method known as SLAM (Simultaneous Localization And Mapping). SLAM is the method of performing robot navigation in an unknown environment. The experiment presented in this paper aims to solve the above problem in two different simulated Gazebo environment using ROS (Robot Operating System), with a custom differential drive robot model. The Graph SLAM method, along with its implementation as a ROS package and challenges to setup as well as usage of the package are discussed. The results of the experiment will fuel further research in autonomous robot navigation.

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

1 Introduction

In robotic systems the navigation problem consists of three important sub-problems; localization, mapping and path-planning. Localization is the process of a robot estimating its location in an environment, based on a combination of its prior knowledge of the environment and sensor measurements, while mapping is that branch which deals with the study and application of ability of a robot to localize itself in a map / plan and sometimes to construct the map or floor plan. Both of these problems are dependent on each other, for accurate results. The SLAM (Simultaneous Localization And Mapping) is an approach to solving both these problems in simultaneously, in real-time. It is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. The SLAM problem can be solved using a number of approaches, using a number of algorithms in each. In this paper, the Graph SLAM approach is used to solve the Full SLAM problem, thus allowing the robot to estimate its entire navigation path.

2 Background

The SLAM problem can be divided into two distinct problems, the Online and the Offline SLAM problem. The nature of the SLAM problem can be continuous or discreet depending upon the environment. The Online problem deals with estimating the robot path, based on the only the current state i.e, the robot's current state depends only on its immediately previous state. The Offline or Full SLAM problem involves estimating the entire path of the robot given its previous and current states. There are various algorithms that solve each of these approaches. The FastSLAM approach is a popular approach used to solve the full SLAM problem. This approach has 3 variants namely FastSLAM 1.0 2.0 which assume known landmarks and Grid-Based FastSLAM, which is non-landmark based algorithm. An advantage of this approach is that it uses a particle filter or an occupancy grid map algorithm to estimate it's position. Another approach to solving the Full SLAM problem is Graph SLAM. It is by far, one of the most complicated SLAM algorithms since it can require a lot of computation. This approach uses four types of data namely, Poses, Features, Motion constraints and Measurement constraints. The work done by the algorithm is split into two parts, the Front-End and Back-End. The Front-End deals with constructing the graph using the sensor and odometry measurements collected by the robot. It also performs the task of solving the data association problem i.e checking whether a location has been visited before or not. The Back-End deals with graph optimization and is responsible for producing the output containing the most probable poses of the robot. This approach, although effective, is a 2D SLAM approach. In this paper, a 3D Graph based SLAM approach called RTABMAP is used. This method is explained in detail in the upcoming section.

2.1 RTABMAP

Real-Time Appearance Based Mapping, also known as RTABMAP is a 3D graph SLAM approach that uses visual odometry and sensor measurements to approximate it's map. The Front-End of the algorithm uses the sensor and odometry measurements to perform loop closure detection, whereas the Back-End is used for Graph Optimization and 2D/3D Map Generation. Loop closure is a method to solve the data association problem is mentioned above. In RTABMAP, loop closure is done using a method called SURF or Speeded Up Robust Features. Here, each feature has a descriptor associated with it which is a representation of the pixels that make up a feature and the point where the feature is located, is split into smaller square sub regions. From these sub-regions the regions of regularly spaced sample points are calculated and compared. In order to compare the large amount of features, they are clustered into groups of similar features or synonyms. Collection of these clusters represents the vocab. when a feature descriptor is matched to one in the vocab it is called quantization. now the feature that is linked to a word, can be referred to as a visual word. When all features in an image are quantized the image is now a bag of words each word keeps track of the images that it is associated with, for efficient retrieval of data. A matching score is given to all images containing the same words. this is managed in the form of a table called the inverted index. A bayesian filter with the hypothesis that the image is seen before is used to evaluate the scores. once the score crosses a certain threshold then a loop closure is detected. An illustration of visual bag of words approach can be seen in fig 1.

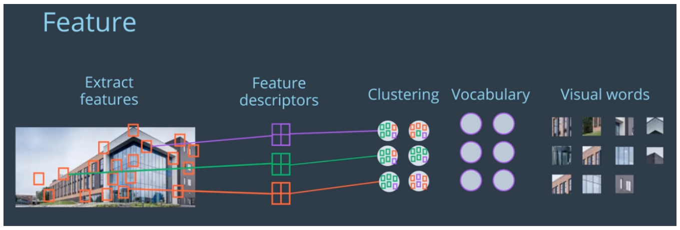

Fig. 1. Visual Bag of Words

The primary advantage of using the RTABMAP approach includes Loop Closure Detection, efficient Memory Management and Graph Optimization techniques such as Tree-based network optimizer, General Graph Optimization etc. An illustration of the memory management in RTABMAP algorithm is given below in fig 2.

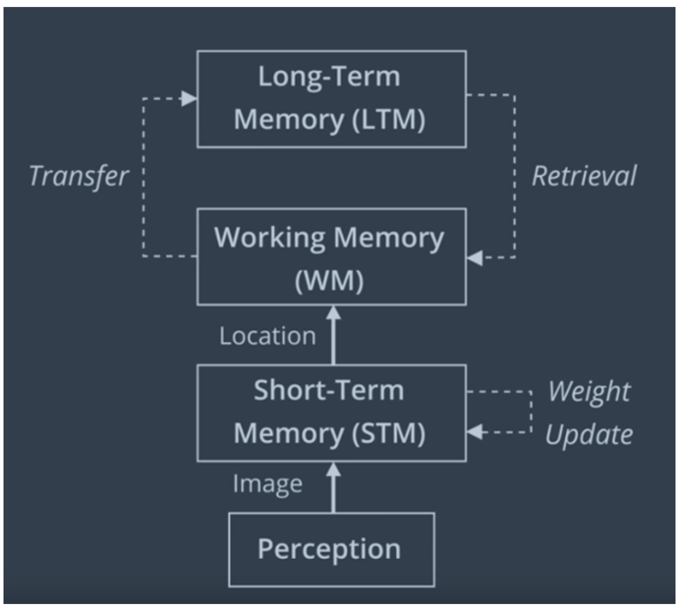

Fig. 2. Memory management

The reason RTABMAP is used for this task, is because of the ability of the algorithm to leverage visual data for generating the map. This makes it easier to perform SLAM using a relatively cheap sensor such as a stereo camera or a kinect, instead of an expensive LIDAR sensor.

3 Scene and Robot Configuration

3.1 Achievements

You should describe what you achieved for SLAM in the project with the kitchen world and your own world.

3.2 Robot Design

3.2.1 Model design

3.3 Personal World

3.3.1 Robot Performance

4 Results

Present an unbiased view of your robot's performance in both worlds and justify your stance with facts. Do the final maps look reasonable? Is the quality of the maps high? In which world did the SLAM perform better?
For demonstrating your results, it is incredibly useful to have some watermarked charts, tables, and/or graphs for the reader to review. This makes ingesting the information quicker and easier.

5 Discussion

This is the only section of the report where you may include your opinion. However, make sure your opinion is based on facts. If your robot performed poorly, make mention of what may be the underlying issues. If the robot runs well, which aspects contribute to that? Again, avoid writing in the first person (i.e. Do not use words like "I" or "me"). If you really find yourself struggling to avoid the word "I" or "me"; sometimes, this can be avoid with the use of the word one. As an example: instead of : "I think the robot could not map the" try: "one may believe the localization performance is poor because the sensor layout is not able to provide enough information for localization". They say the same thing, but the second avoids the first person.

5.1 Topics

- In which world did robot performed better?
- Why it performed better? (opinion)
- What types of scenario could SLAM be performed?
- Where would you use SLAM in an industry domain?

6 Conclusion / Future work

This section is intended to summarize your report. Your summary should include a recap of the results, did this project achieve what you attempted, how would you deploy it on hardware and how could this project be applied to commercial products? For Future Work, address areas of work that you may not have addressed in your report as possible next steps. This could be due to time constraints, lack of currently developed methods / technology, and areas of application outside of your current implementation. Again, avoid the use of the first-person.

Fig. 3. Robot Revolution.

6.1 Subsection Heading Here

Subsection text here.

6.1.1 Subsubsection Heading Here

Subsubsection text here.

- example 1
- example 2

1) example 1
2) example 2

TABLE 1
Table

One	Two
Three	Four

6.2 Modifications for Improvement

Examples:

- Base Dimension
- Sensor Location
- Sensor Layout
- Sensor Amount

6.3 Hardware Deployment

1) What would need to be done?
2) Computation time/resource considerations?