RTAB-Map ROS Package and SLAM

Nicholas Stepanov

Abstract—Mapping is another problem in robotics which goes hand in hand with localization. The problem of mapping is primarily important for robots that get deployed in unknown or undetermined areas. The software pipeline needs to recognize when a robot sees the same place and correctly create a map of the environment. The most sophisticated algorithms use mapping in parallel with localization to perform Simultaneous Localization and Mapping (SLAM). This is even more of a complex problem, but dealing with it requires solid mapping and localization configurations. This project introduces a ROS package that uses rtabmap ROS package for mapping two different worlds deployed on a sample bot in simulation.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, SLAM, RTAB-Map, ROS.

1 Introduction

The problem of mapping with known poses can be solved primitively through Occupancy Grid Mapping using Bayesian filters. This is a probabilistic approach that updates probability distributions based on sensor calculations and their uncertainties. Fast-SLAM and Graph-SLAM are used for performing mapping and localization in an unknown environment. In larger areas with many similar features those algorithms are usually unreliable. In these cases it might be better to create a map first to localize in. If such map is unavailable, rtabmap ROS package can be used to create a map of unknown environment. It uses Bag of Words - a collection of images from the sensors that are split into unique pieces to be identified later. This creates the ability for the robot to detect loop closures - approaching same place from opposite sides, - a common situation during mapping. Such a technique increases the accuracy of mapping significantly, but requires much more computational power. Depending on hardware setup the Bag of Words could be used together with SLAM to improve its performance.

2 Background

Different approaches to mapping differ in details, but the general idea remains the same. A robot has to synchronize its odometer readings with the sensory inputs to create a valid environment map. While simpler probabilistic approaches can be used to map a relatively simple world with a decent end result, more sophisticated methods should be used for complex environments to correctly identify loop closures. This adds extra complexity to the algorithm and, therefore, requires additional computational resources. The complexity of both SLAM and Bag of Words enabled mapping increases linarly with the map size increase to accommodate for new features to be checked on with each new map piece. [1]

2.1 Fast-SLAM

Fast SLAM approximates robot's position based on its movement and uncertainty. It then updates the map with the new information, and repeats the process based on current estimate of robot's position. This approach saves computational power because it does not have to propagate back to the robot's initial position each time the map needs an update. However, such an approach also leads to an ever increasing uncertainty of robots position and orientation and, therefore, makes it harder to correctly identify loop closures. Fast SLAM should be used in environments with low to medium complexity.

2.2 Graph-SLAM

Graph SLAM creates a map of the environment in a similar fashion that a Fast SLAM algorithm. It adds extra certainty to the mapping process by recalculation of robot's path starting from the origin for every new place a robot finds itself in. While this process will generally improve the quality of the map, it should be done using high power GPU's because math behind it will include back propagation and updates for each position node in the Graph SLAM and is very similar to the back propagation used in training Neural Networks. This is why Graph SLAM can be used in environments of medium complexity to create an accurate map and localize on it. However, the complexity of Graph SLAM will increase at a higher rate than Fast SLAM with map size increases, which might become an issue with larger environments.

2.3 RTAB-Map using Bag of Words

While SLAM is very useful for localizing and mapping at the same time, larger maps could be too complex for the existing hardware to deal with. RTAB-Map ROS package uses algorithm known as Bag of Words to create unique items using pieces of the camera input. Each new iteration this bag is updated to include additional input, or to match against an existing entity. This improves mapping process to be able to combine the knowledge of robot's odometer readings with sensory inputs to better understand locations of loop closures. This can be used either during SLAM or separately to create the map of environment. Uniqueness of the environment is crucial for the algorithm to perform well, and if such is satisfied, the algorithm would successfully map even large areas.

- Probabilistic Occupancy Grid Mapping: Simple and Cheap, Inaccurate
- Fast-SLAM: Increasing complexity, Increasing uncertainty, Relatively cheap
- Graph-SLAM: Increasing complexity, Decent computational power consumption
- RTAB-Map: Increasing complexity, Decent computational power consumption, Works better with environments that have locations distinguishable from each other

3 SIMULATIONS

The udacity bot model from Localization project is used to map two different environments using RTAB-Map ROS package. Done on Udacity's GPU servers such simulations require additional power for creating the environment itself. The provided package could be used to recreate the maps, or create a map of a new environment. Results are discussed along with the possibilities for future improvements.

3.1 Achievements

The robot created a perfect map for the supplied environment. The world created by me was more simple and had more similar features in different locations on the map. This made the mapping process perform less well than in the supplied environment.

3.2 Gazebo world

A Gazebo world was created using standardly available items. This world is and exposition of different robotic projects located in a circle.

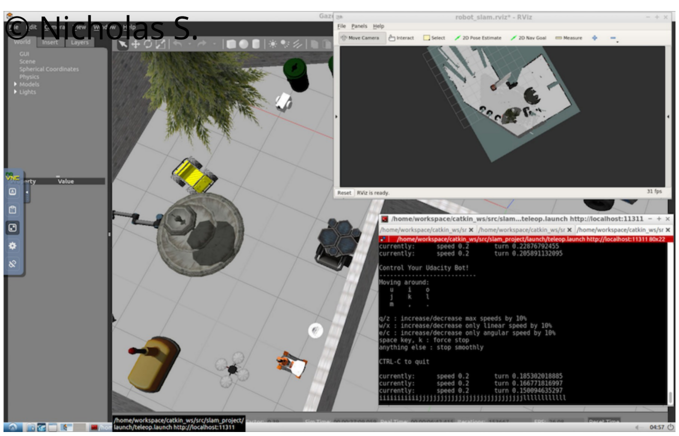

Fig. 1. My world, its mapping process and resulting 2D/3D maps

4 WORK DONE AND RESULTS

After configuring the package which included creating .launch files for launching the simulated world, launching the mapping algorithm and launching robot's teleoperation, the map could successfully be created. Additional work included setting up the correct topic names and remapping RTAB-Map package inputs and outputs, modifying udacity bot to have an RGBD Kinect camera with relevant publishers. The camera base frame needed to be modified in order to correctly display the sensor data inside RTAB-Map.

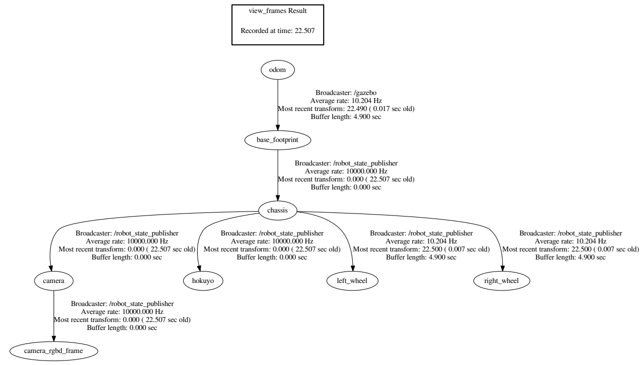

Fig. 2. Robot TF Tree

4.1 Mapping Results

As expected, the supplied environment performed much better than the one created by myself. The resulting map had no discrepancies from the original environment and can be easily used for localization in it.

Fig. 3. 3D map of the provided environment

4.1.1 *Kitchen Dining*

Mapping the kitchen dining environment was relatively easy because the environment is both simple and its different parts are easily distinguishable. The server was running smoothly because of that, and did not get stuck (lags) for it to complete a lot of computations.

4.1.2 *My World*

My World was little trickier to map, and a complete map has never ended up being uncorrupted. The reason behind that was a great amount of similarity between different places inside the environment. Less detailed maps of that environment were produced without corrupting the file. It

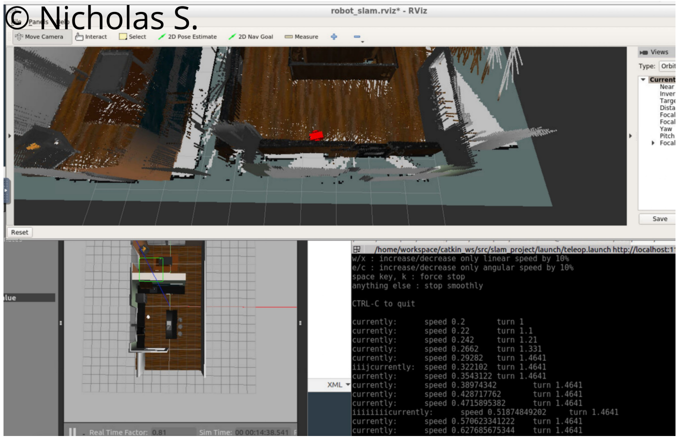

Fig. 4. Mapping Supplied Environment

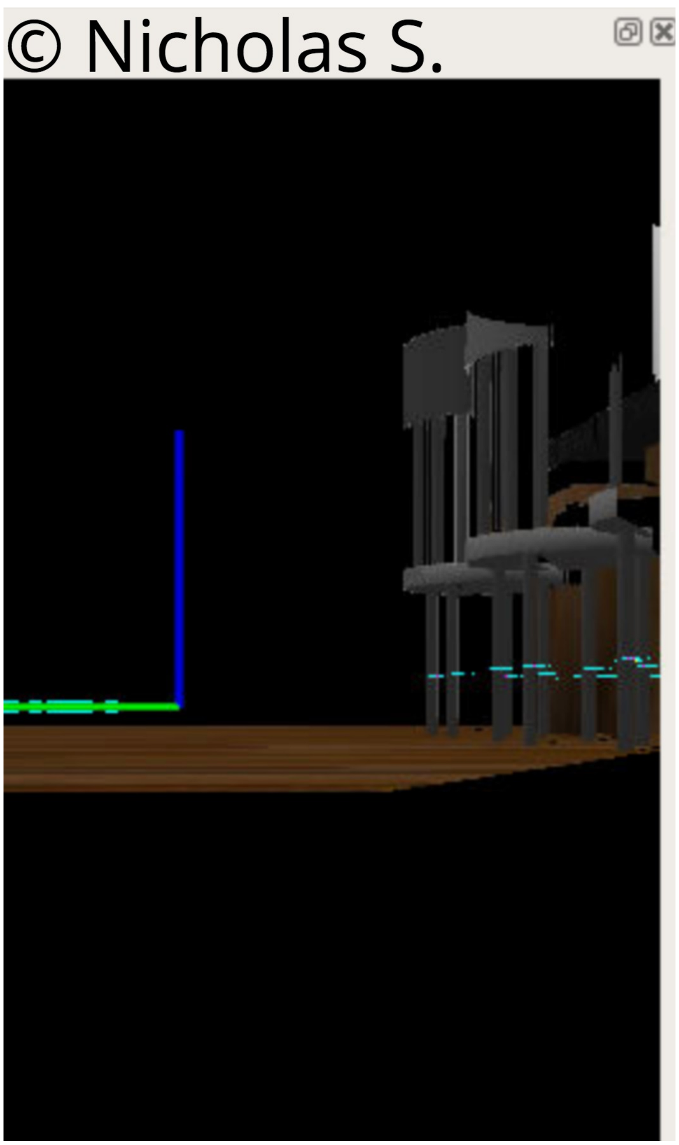

Fig. 5. Sample picture

is important to understand that these kinds of challenges are not unique for simulation environments. Real world problems can have a same problem of determining whether an object that has already been seen is a new one or the already seen one. Simple solutions to making the mapping process more smooth are: low speeds, avoiding collisions, linear movements and minimized amount of rotations. These suggestions will improve the quality of the map produced in an unknown environment both with mapping and SLAM.

4.2 Technical Comparison

While in a lot of cases the real world environment would be similar to the first one: kitchen_dining; there is also a sufficient probability that the details in the real environment will have common mapping problems that were experienced in mappin my_weird_expo.

5 Discussion

The problems of mapping as well as SLAM are very dependent on the type of the environment the robot is expected to find itself in. For this reason it is important to give a lot of thought to the possible environments the robot would encounter and choose the right algorithm to perform mapping/SLAM with the highest efficiency. The observations from simulation were very useful to translate into real world problems and how they would affect mapping. While the mapping techniques described above are able to accomplish their tasks, more complex mapping might require new algorithms and ways to deal with uncertainties.

5.1 Topics

- Which mapping performed better? Provided map performed better
- Why it performed better? (opinion) Provided map was much more suitable for the mapping process
- How would you approach the SLAM using Bag of Words problem? To test SLAM using Bag of Words I would route the rtdatabase with map information to the localization node to update robot's understanding of its position and orientation in-between each map update.
- In what types of scenario can this be performed? While RTAB-Map can be used to create maps of real world places (these could be unaccessible as well like Lunar/Martian maps), SLAM should be used in a more contained environments which do not have a predefined map.
- Where would you use SLAM and Mapping in an industry domain? Delivery, security or transportation robots that are designed to only be in certain places (company headquarters, shopping malls etc.) would all benefit from moderate usage of SLAM and Bag of Words algorithms.

6 Conclusion / Future work

The mapping project showed the practical side of the mapping/SLAM process and how complicated it can be. The expected result was reached, however, the world created by myself showed the extra challenges that mapping has to deal with. To smoothen out these I would investigate into the RTAB-Map package parameters and how they can be used to improve real loop closures and make mapping less corruptible.

6.1 Modifications for Improvement

Examples:

- Improved Mapping: Various tune ups can be made to improve the mapping inside the environment that did not perform as good.
- SLAM implementation: creating a package that would use RTAB-Map for simultaneous Localization and Mapping is another challenge for this project, which, however, might be too complex computationally for a simulation.
- Hardware Deployment: The project settings can be changed to be deployed on an actual robot. Several nodes would need to be updated to correctly control and receive info from real hardware.

6.2 Hardware Deployment

1) What would need to be done? Package needs to be modified depending on hardware specifics to correctly pipeline sensory data as well as velocity commands and odometer readings.
2) Computation time/resource considerations? Deployment on a real robot would be less computationally expensive than running in a simulation. Therefore, I expect the state of the art Nvidia TX2 embedded system to easily accomplish this serious task.

References

[1] L. Lamport, *LATEX: a document preparation system: user's guide and reference manual*. Addison-wesley, 1994.