

# World Mapping Using ROS RTAB-Map

Ussama Naal

**Abstract**—This work. investigates the use of one implementation of a Graph-based SLAM known as RTAB-Map, in order to create a 3D mapping of a given site. Within a simulated environment used Gazebo and ROS, we deploy a two wheeled robot equipped with a laser scanner and an RGB-D camera and runs a ros package that implements the RTAB-Map algorithm. We test the configured robot under two different sites and demonstrate obtained results.

**Index Terms**—world mapping, ROS, slam, GraphSLAM, RTAB-Map.

## 1 INTRODUCTION

Successful robot navigation requires having an accurate representation of the world that robot traverses through. When a detailed map of the world is not readily available to the robot or current map is not up to date, in the case, the robot could rely on its sensors to survey the site and build a good approximation of the site. Moreover, since the set of the sensors that the robot rely on could be subject to noise and inaccuracies which leads to drift in the robot position, the robot can at the same time utilize the map being created in order to find the actual location of the robot within the map. This process is known as Simultaneous Localization and Mapping (SLAM). In next section we outline the problem and available solutions. In section 3, we review our robot configuration. In section 4, we review achieved results when running the RTAB-Map algorithm against two different environments. We discuss these results in section 5. Finally, in section 6, we provide conclusions and hint at future work.

## 2 BACKGROUND / FORMULATION

SLAM deals with the case when the robot has only access to its sensors data and knowledge of its own movement. Using available data the robots tries to estimate the map and its position within the estimated map. The main challenge with estimating the map lies within the count of state variable to describe the map. Another challenge is when dealing with a representation that contain repetitive patterns causing uncertainty when estimating the robot pose. Known SLAM-based approaches can be categorized into five categories: Extended Kalman Filter SLAM (EKF) [1], Sparse Extended Filter Information SLAM (SEIF) [2], Extended Information Form (EIF) [3], FastSLAM [4] and GraphSLAM [5].

In this work we employ the RTAB-Map which is GraphSLAM-based approach. In GraphSLAM, a graph is constructed where nodes represent estimated robot poses

and estimated locations of landmarks or identifiable features on the map at a given point in time. Edges on the other hand represent constraints that connect between robot poses and the detected features on the map. There are two types of constraints: motion constraint and measurement constraint. Motion constraints represent the distance between robot poses at different times, while measurement constraints represent the distance between a given pose and the distance to a nearby landmark or feature measured by the robot at that pose. Following figure shows this relationship.

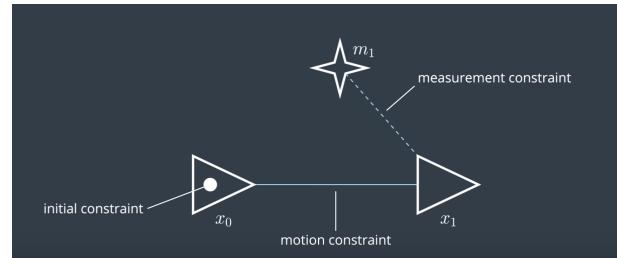


Fig. 1. Constraints Graph.

The robot continues to build the graph as it moves and surveys its environment adding new nodes and constraints, however, due to noise and inaccuracies of estimated poses and measurements the constraints may not map properly. This results in an over-determined system which converts the problem into an optimization problem. To solve the system, graph data is organized into an information matrix and an information vector where cells represent the linearized constraint around an estimate between two given nodes. Both data structures get passed to a numerical solver that returns most probable configuration of the graph.

In GraphSLAM it is essential to detect loop closures when surveying the environment, i.e. identify if the robot has observed the same place again. Detecting loop closures requires comparing current acquired scan of the environ-

ment against all previous scans. This is a very daunting task since complexity increases linearly as the robot acquires new scans. The RTAB-Map performs a set of techniques to the original GraphSLAM to keep the process running near real-time. These techniques can be summarized in: Local/Global Loop Closure Detection, Visual Bag-of-Words, Memory Management.

### 3 ROBOT CONFIGURATION

We used the robot configuration used in the previous project [6] with one exception. In this robot we replace previous RGB camera with a depth camera also known as an RGB-D camera.

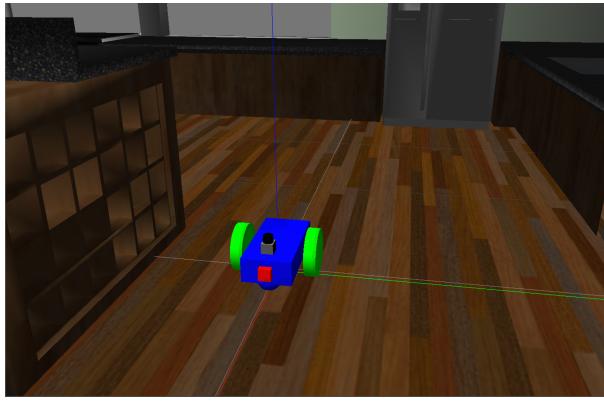


Fig. 2. Robot in the kitchen environment.

We make sure to update the gazebo files such that sensors data are published to the channels.

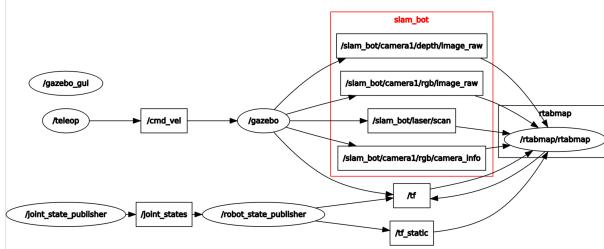


Fig. 3. Robot Graph.

### 4 RESULTS

In this section we review results obtained by deploying the robot with the same algorithm and configuration to two different environments. First environment is the kitchen environment and then a personal one.

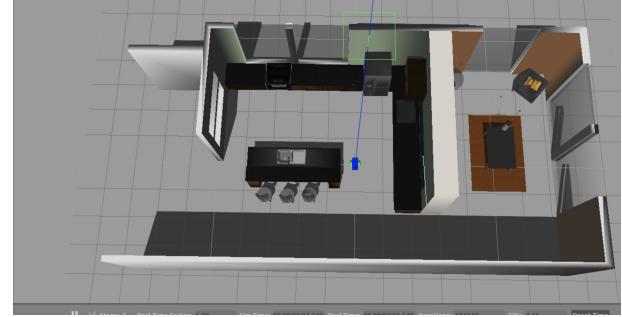


Fig. 4. Kitchen Environment.

#### 4.1 Kitchen Environment

The following image shows the configuration of this environment within the simulator:

Images in figure 5 and 6 show the path took by the robot and the detected features and how these features were used to correspondence between two different frames.

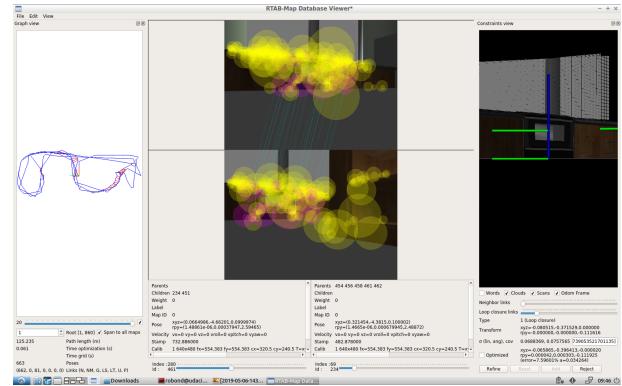


Fig. 5. Kitchen Environment: Graph View and Feature Correspondence 1.

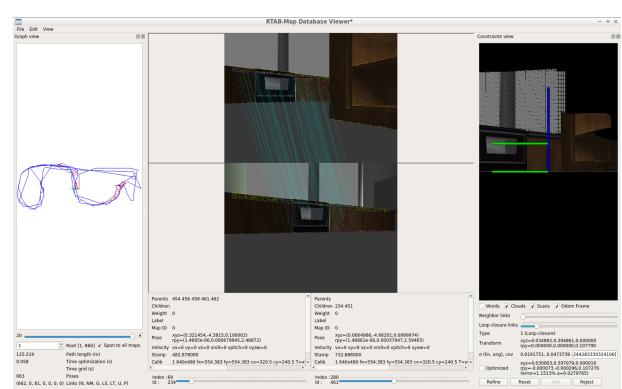


Fig. 6. Kitchen Environment: Graph View and Feature Correspondence 2.

Finally, Figure 7 shows 3D reconstructed point cloud of the kitchen environment. We can clearly see that RTAB-Map creates a good representation of the environment.

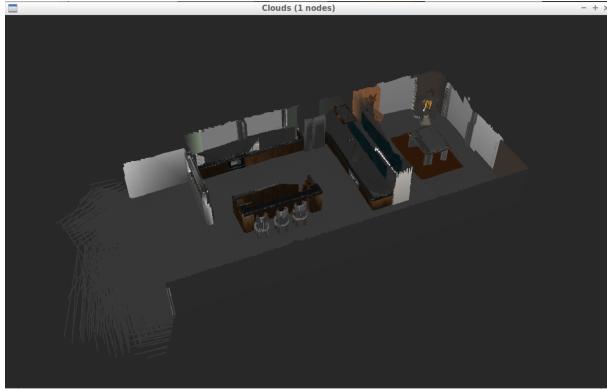


Fig. 7. Kitchen Environment: Point cloud generated through RTAB-Map.

## 4.2 Personal Environment

Using available objects in Gazebo, we construct personal environment that simulate site that has many wrecks and ambulance vehicle and few other objects and trees. Figure 8, displays the configuration used in the personal environment within the simulator.



Fig. 8. Personal Environment.

Same in previous sub-section, We show the two following images show the path took by the robot and the detected features and how these features where used to correspondence between two different frames:

Lastly, next image shows 3D reconstructed point cloud of the personal environment. We can observe partial reconstruction of the scene but the algorithm was successful at closing loops, which caused discontinuities for the 3D constructed scene.



Fig. 9. Personal Environment: Graph View and Feature Correspondence 1.

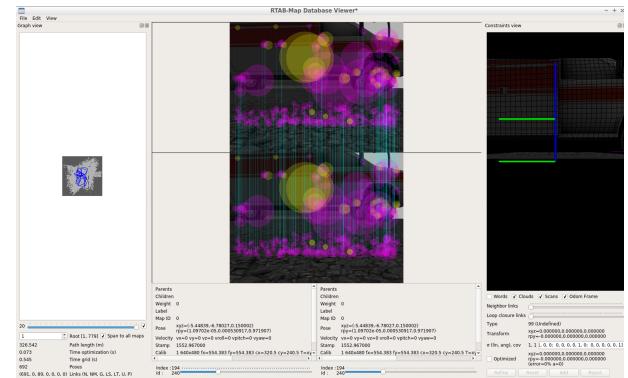


Fig. 10. Personal Environment: Graph View and Feature Correspondence 2.

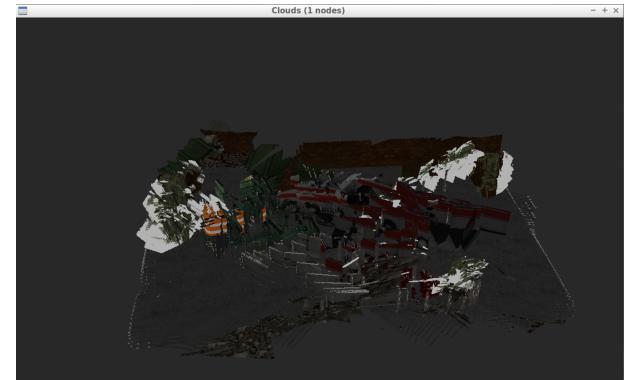


Fig. 11. Personal Environment: Point cloud generated through RTAB-Map.

## 5 DISCUSSION

Evidently, the algorithm managed to successfully map the kitchen environment but was as successful for the personal environment. The failure to map the personal can be mostly attributed to the lack of sufficient distinguishable features in the environment that the algorithm could rely on. Additionally, due to the size of the personal environment, we navigated the site twice only. This is in order to keep the size of the rtabmap database file within manageable limits.

## 6 CONCLUSION / FUTURE WORK

As we saw, the rtabmap algorithm can successfully create a good representation of a survey site, given the site has good amount of structure.

For future, we would investigate that fine tuning some of the parameters available in the ROS RTAB-Map package help to successfully map the constructed personal environment.

Another thing to configure and run the system on hardware that an RGB-D camera and use it to create a 3D mapping of single object or a whole environment such my personal room.

Also, we would like to test the algorithm performance on creating a 3D representation of a living object.

## REFERENCES

- [1] wikipedia, “EKF SLAM.” [https://en.wikipedia.org/wiki/EKF\\\_SLAM](https://en.wikipedia.org/wiki/EKF_SLAM), 2019. [Online; accessed 12-May-2019].
- [2] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. F. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *I. J. Robotic Res.*, vol. 23, pp. 693–716, 07 2004.
- [3] P. S. Maybeck, *Stochastic models, estimation, and control*, vol. 3. Academic press, 1982.
- [4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “Fastslam: A factored solution to the simultaneous localization and mapping problem,” in *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598. AAAI, 2002.
- [5] S. Thrun and M. Montemerlo, “The graph slam algorithm with applications to large-scale mapping of urban structures,” *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.
- [6] U. Naal, “ROS AMCL Robot Navigation.” [https://github.com/Samahu/RoboND-WhereAmI-Project/blob/master/Udacity\\_Where\\_Am\\_I\\_Project\\_Report.pdf](https://github.com/Samahu/RoboND-WhereAmI-Project/blob/master/Udacity_Where_Am_I_Project_Report.pdf), 2019. [Online; accessed 12-May-2019].