RoboND Project 3: Map My World Robot

Jaeyoung Lee

Abstract—Simultaneous Localization and Mapping(SLAM) is fundamental to mobile robotics and one of the most challenging issues. Through SLAM process a robot can construct a map of the environment while simultaneously localizing itself relative to this map. There are 2 well known SLAM strategies, FastSLAM and GraphSLAM. In this writeup overall SLAM process and these 2 strategies are covered briefly, and then one of the most famous implementation of GraphSLAM, Real-Time Appearance-Based Mapping(RTABMap) is demonstrated on ROS Platform with figures and parameter values. Important issues related to parameters and environment setups for RTABMap are also mentioned. Finally other options beyond the RTABMap and future testing environments are discussed.

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

1 Introduction

Simultaneous Localization and Mapping is fundamental to mobile robotics such as robot vacuum cleaners and one of the most challenging issues. Basic mapping process produces a map of the environment given the robot's poses using the known trajectory and measurement data. Unfortunately this assumptions can not be applied in real world since robots need to estimate both the map and the poses at the same time in real world environment. Through SLAM process a robot can construct a map of the environment while simultaneously localizing itself relative to this map (unknown map and unknown poses). It is more challenging than localization and mapping process since neither the map nor the robots poses are provided. With noise in the robot's motions and measurements, the map and robot's pose will be uncertain, and the errors in the robot's pose estimates and map will be correlated. The accuracy of the map depends on the accuracy of the localization and vice versa.
In this writeup overall SLAM process and these 2 strategies are covered briefly in Section 2. Additionally one of the most famous implementation of GraphSLAM method, Real-Time Appearance-Based Mapping(RTABMap) is explained. In Section 3 to 5 RTABMap implementation is demonstrated on ROS Platform with figures and parameter values. Important issues related to parameters and environment setups for RTABMap are also mentioned. Finally other options beyond the RTABMap and future testing environments are discussed in Section 6 to 7.

2 Background

The problem of generating a map inside of an unknown environment is called mapping. Mapping with mobile robots is a challenging problem for two main reasons. First the map and poses are both unknown to the robot. Second the hypothesis space is huge.

- Localization := $P(x_{1:t}|u_{1:t}, map, z_{1:t})$
- Mapping := $P(map|z_{1:t}, x_{1:t})$
- SLAM := $P(x_{1:t}, map|z_{1:t}, u_{1:t})$

SLAM Algorithms generally fall into five categories:

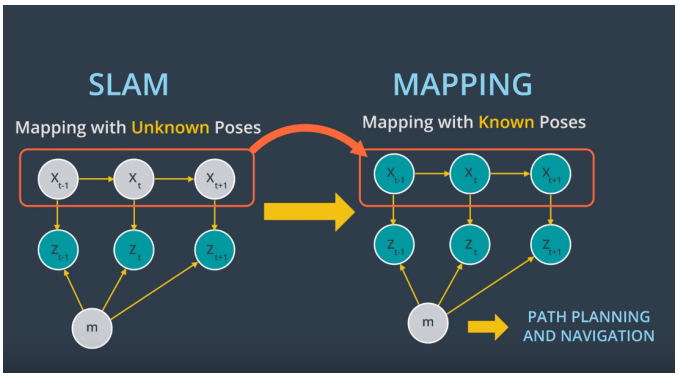

Fig. 1. Relationship between SLAM and Mapping

- Extended Kalman Filter SLAM(EKF)
- Sparse Extended Information Filter(SEIF)
- Extended Information Form(EIF)
- FastSLAM
- GraphSLAM

FastSLAM is the particle filter approach along with the low dimensional Extended Kalman Filter to solve the SLAM problem. This can be adapted to grid based maps.

2.1 Grid based FastSLAM

The FastSLAM algorithm solves the Full SLAM problem with known correspondences.

- Estimating the Trajectory: FastSLAM estimates a posterior over the trajectory using a particle filter approach. This will give an advantage to SLAM to solve the problem of mapping with known poses.
- Estimating the Map: FastSLAM uses a low dimensional Extended Kalman Filter to solve independent features of the map which are modeled with local Gaussian.

The custom approach of representing the posterior with particle filter and Gaussian is known by the Rao-Blackwellized particle filter approach. Grid-based FastSLAM is really an extension to FastSLAM, which adapts FastSLAM to grid

maps. With the grid mapping algorithm the environment can be modeled using grid maps without predefining any landmark position. So by extending the FastSLAM algorithm to occupancy grid maps, the SLAM problem can be solved in an arbitrary environment. To adapt FastSLAM to grid mapping, three different techniques are needed:

- Sampling Motion-$p(x_t|x_{t-1}^{[k]}, u_t)$: Estimates the current pose given the k-th particle previous pose and the current controls u. (MCL)
- Map Estimation-$p(m_t|z_t, x_t^{[k]}, m_{t-1}^{[k]})$: Estimates the current map given the current measurements, the current k-th particle pose, and the previous k-th particle map. (Occupancy Grid Mapping)
- Importance Weight-$p(z_t|x_t^{[k]}, m^{[k]})$: Estimates the current likelihood of the measurement given the current k-th particle pose and the current k-th particle map. (MCL)

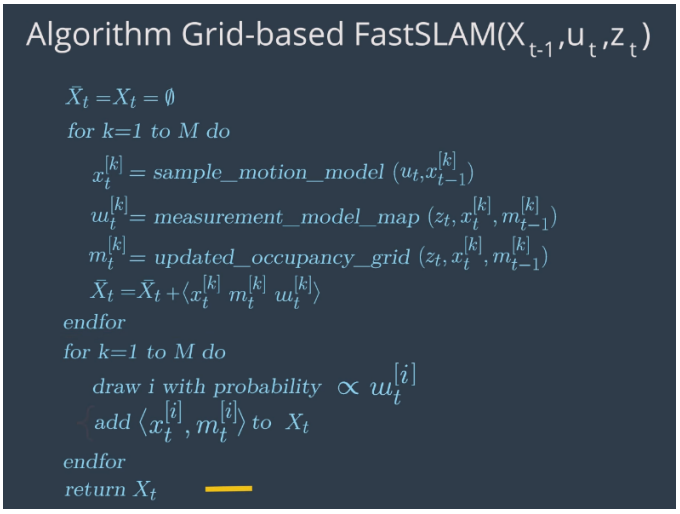

Fig. 2. Grid-based FastSLAM

2.2 GraphSLAM

GraphSLAM uses graphs to represent the robot's poses and the environment. The brief process of GraphSLAM is as fallows:

- Construct Graph (Add Nodes and Edges)
- Define Constraint on the Edges
- Solve the System
- Linearize (if Nonlinear) using Taylor's expansion.
- Optimization using Iterative Methods (Gradient Descent etc..)

Fig 3 shows a example graph.

2.3 Comparison / Contrast

GraphSLAM improved accuracy over FastSLAM. FastSLAM uses particles to estimate the robot's most likely pose. It means that it's possible that there is not a particle in the most likely location. Specially in large environments, the chances are converging to zero. Even though it find likely location with moderate possibility, its error not suitable for practical application. All particle based approaches have the same issue. On the other hand GraphSLAM solves the full slam problem, it can work with all of the pose at once to find optimal solution. This improvement are related to the progress in terms of computation. Currently since high performance system including embedded APU emerge at appropriate price. even complex optimization problems can be solved in real time.

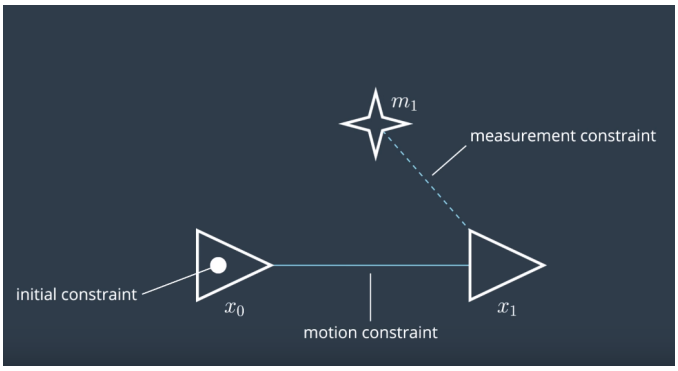

Fig. 3. Graphs for GraphSLAM

2.4 RTABMap

Real-Time Appearance-Based Mapping (RTABMap) is one of the most widely used GraphSLAM method. And its author's implementation is added to the ROS eco-system and it is still well managed.
Appearance-Based means that it uses visual sensor data such as RGB images. In the Appearance-Based Method a process called Loop Closure is used to determine whether the robot has seen a location before. As robot travels to the new areas in its environment the map is expanded and the number of images that each new image must be compared to increases. Because of the increasing number of images complexity of the loop closure increases linearly. RTABMap is optimized for large-scale and long-term SLAM by using multiple strategies for real-time computation: loop closure detection method using bag of words to determine the difference distance between two images [1] and specialized memory management [2].

2.4.1 Bag of Words

RTABMap uses Speed-Up Robust Feature(SURF) [3] for visual instance detection. Since SURF feature extraction from images is performed fast enough real-time lots of real time applications use it for detection and identification. Basic distance is determined the number of common features between two images with appropriate threshold. Which two features are common or not is determined using Bag of Words techniques. Bag of Words is commonly used in vision-based approaches. Comparing feature descriptors directly is time consuming. So limited-size vocabulary is used for faster comparison. This vocabulary is where similar features(synonyms) are clustered together. One of feature detected on images is mapped to one through the process called quantization. When all features in an image are quantized the image is not Bag of Words.

2.4.2 RTABMap Memory Management

Fig 4 shows overall process of RTABMap. Fig 5 shows memory management system.

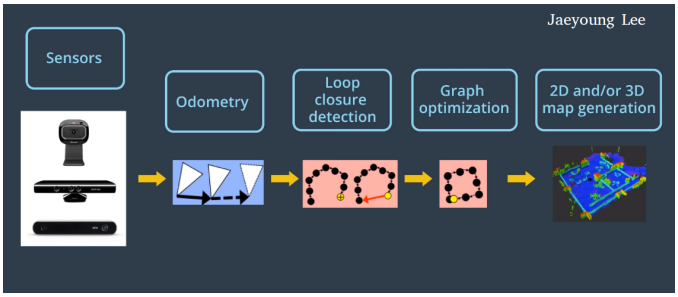

Fig. 4. RTABMap Process

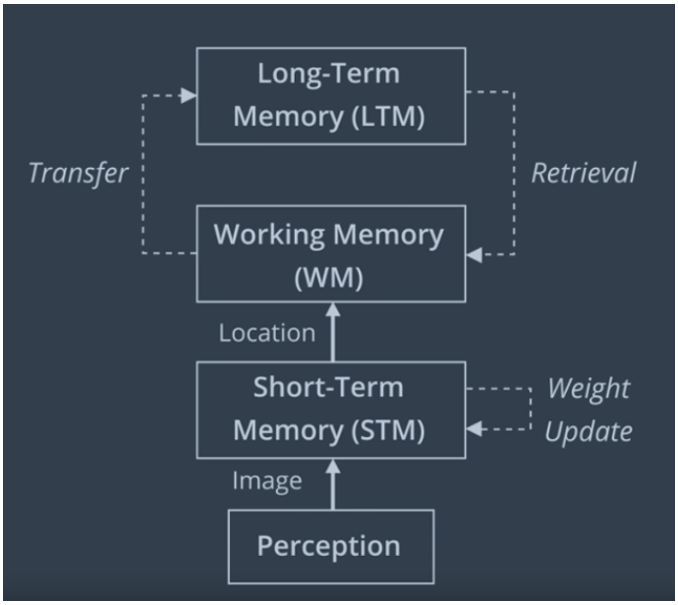

Fig. 5. RTABMap Memory Management

2.4.3 RTABMap vs Other GraphSLAM

Although landmark constraints are used for other GraphSLAM method like 2D GraphSLAM, RTABMap does not use them. Only odometry and loop closure constraints are considered. RTABMap supports appearance based method with no metric distance information. In this setup RTABMap can use a single monocular camera to detect loop closure. To use metric distance information RGBD cameras or stereo vision cameras are required.

3 Simulations

In this section all simulation environment setup except for ROS robot model and scene are described. Local desktop PC with high performance NVIDIA 1080TI is used for the simulation.

3.1 Packages Used

The list below shows the packages used in the project.

- gazebo_ros
- move_base
- rtabmap_ros

3.2 Parameters for RTABMap SLAM

All supplied parameters are used as default for the experiments, except for Vis/MinInliers. This is mentioned in the later sections.

- Rtabmap/DetectionRate := 1
- Reg/Force3DoF := true
- Kp/DetectorStrategy := 0(SURF)
- Kp/MaxFeatures := 400
- SURF/HessianThreshold := 100
- Reg/Strategy := 0 (Visual Appearance)
- Vis/MinInliers := 15 (50 for customized world model)
- Mem/NotLinkedNodesKept := false

4 Scene and Robot Configuration

4.1 Kitchen Dining World (Supplied kitchen_dining.world)

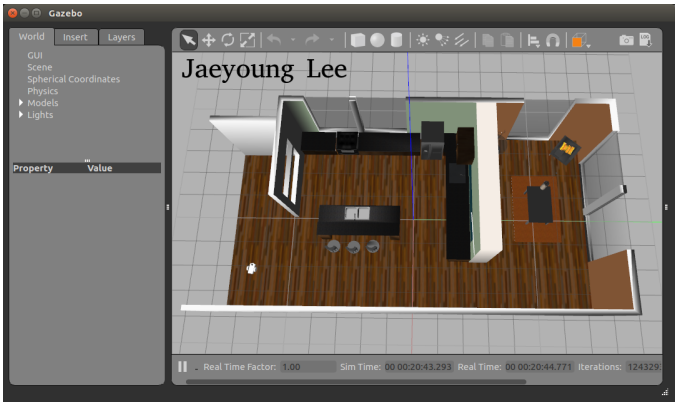

Fig. 6. Gazebo

4.2 Customized World (customized.world)

Fig 7 shows the customized gazebo world model. The base model is C̈afeM̈odel provided from the http://gazebosim.org/models. Tree, Cafe Table, Table, Box, BookShelf object are added to the scene. Whole these small objects are scattered randomly. These objects are needed for easier loop closure detection. This is mentioned in detail in the next chapter.
Fig 8 shows the robot used in the project. Fig 9 shows all frame information: 2 wheels, 1 laser scanner and 1 rgbd camera. camera_rgbd_optical_frame is the RGBD camera image plane.

5 Results

5.1 Kitchen Dining World (Supplied kitchen_dining.world)

Fig 10, Fig 11 show the 2D, 3D maps generated after mapping on customized world model. 314 nodes and 16 detected loop closures are generated in the experiment (Fig 13).

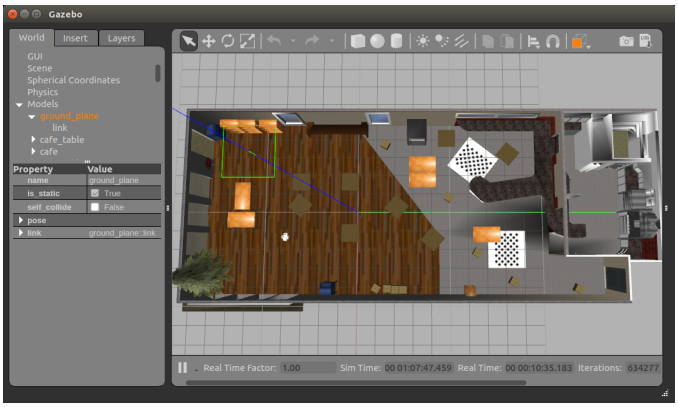

Fig. 7. Gazebo for Customized Cafe Model

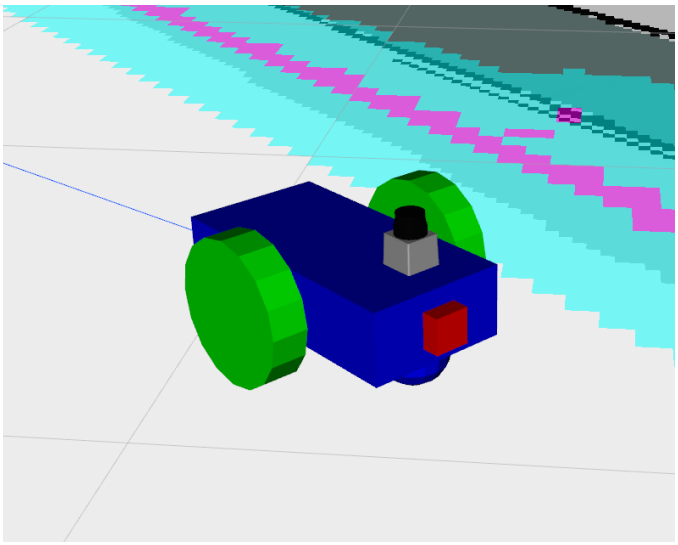

Fig. 8. Robot

5.2 Customized World (customized.world)

Since the floor pattern in the Cafe model are too compact and repetitive, there are too many repetitive SURF features. Because of these repetitive SURF features image matching using Bag of Word may not distinguish wekk two rgbd images taken at the different position. It means that exact loop closure detection is harder than the supplied ̈kitchen_dining.world ̈model. Fig 14 shows the example of the loop closure failure. This issue will be discussed in more detail and depth in Section 6. For easier loop closure detection, small objects with random pattern are added as marks for SURF features. And the RTABMap parameter Vis/MinInliers was set up to 50.
Fig 15, Fig 16 show the 2D, 3D maps generated after mapping on customized world model.

6 Discussion

Main issue in RTABMap is loop closure detection. Bag of Words strategy is a traditional method and it works well in the limited environment, where there are lots of distinct SURF features for example a garden with lots of stones and plants or a messy room with lots of stuffs and wall paintings. But the opposite situation is shown in the previous section. Human-made environment such as tiled floors and wallpapers with repetitive patterns can confuse loop closure detectors because that patterns have identical SURF features and distribution in the multiple position. Under this condition false positive loop closure can be detected and this failure ruined all SLAM results. Fig 14 shows the example of the loop closure failure. Once loop closure fails, subsequent processes also all fails. It is a critical flaw of RTABMap
To prevent this failure parameter fine-tuning such as increasing the inliner threshold and other SURF related setup carefully. As more fundamental solution deep learning methods are also emerging [4]. Over all the issues in computer vision and robotics traditional Bag of Words based methods are replaced with deep learning solutions very quickly. Deep learning solutions can detect or distinguish even simple objects with a few features such as simple plane boxes, which are very difficult for the traditional one. If it is applied to this loop closure detection process, it will show high accuracy by filtering repetitive patterns and focusing the non-repetitive features such as the corners of tables and unique corner points.
One limitation of deep learning solutions is that it needs high performance computation. If more powerful mobile system with high performance GPU are released in the future, deep learning approach will be applied to SLAM problem more actively.

7 Conclusion / Future work

One of the important future work is implementation of deep learning based loop closure detection. Typical CNN architecture and lots of data based on simulation without noise setup are needed. Several CNN models and hyper parameter setups should be tested for better solution.
Besides developing new one, there are still a lot of test for experiments in the real hardware environment.

7.1 Harsher Simulation Test

Before the hardware deployment, lots of simulation test under the various conditions should be done since testing on real hardware environment is harder to set up and more time-consuming. And ROS support this types of test through many parameters in ROSMSG.

7.1.1 Odometry noise simulation

Odometry measurement data in real environment is so noisy unlike gazebo generated one. If appropriate noises are setup for the Odometry rostopic, it makes the experiment hard to solve but it can reflect real world.

7.1.2 RGB and Depth image noise simulation

Real RGB images are a bit blurred comparing to the 3d simulation generated images. And depth noise in real environment is severely noisy regardless of its spec resolution. Lots of pixels have unknown value in depth images and their error variance is too high even if they have values. By adding large noise and removing random pixel values, harsher test can be done before real world test.

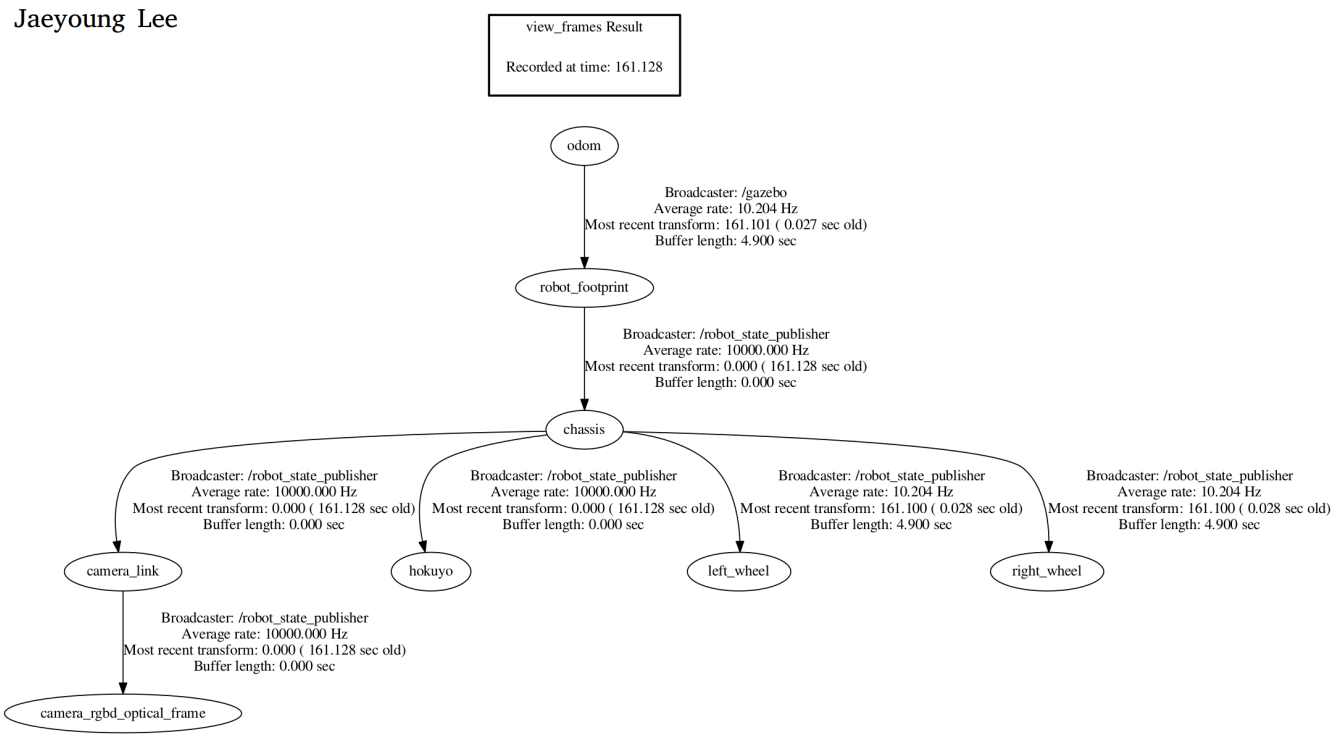

Fig. 9. Robot Frame Architectures

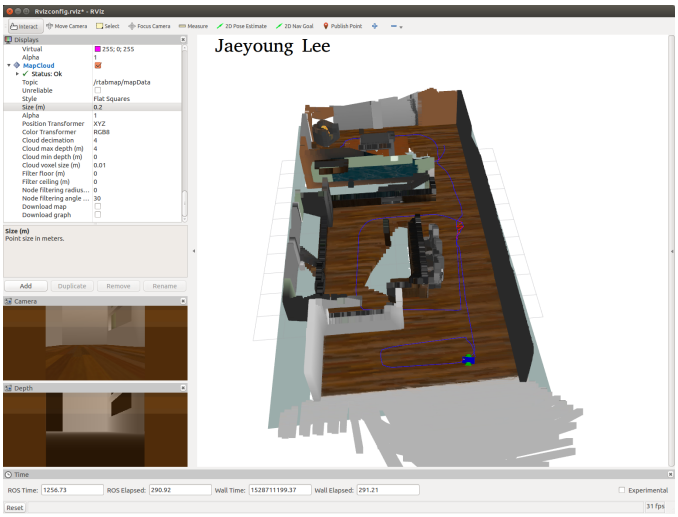

Fig. 10. 3D Map

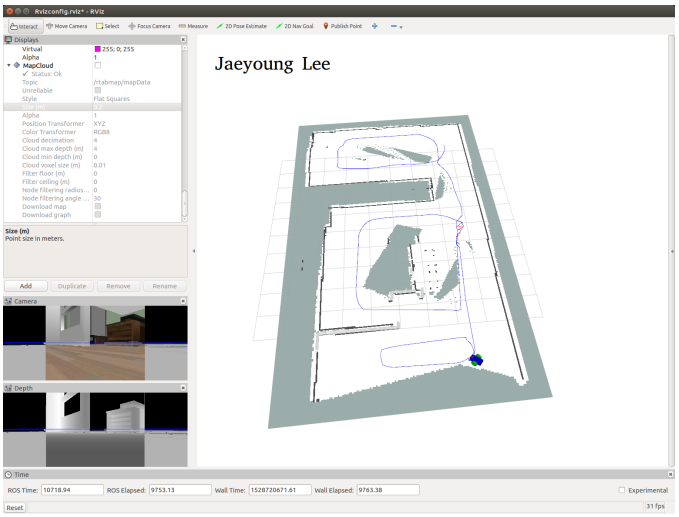

Fig. 11. 2D Map

7.2 Hardware Deployment

7.2.1 Memory Limitation

Mobile embedded system like TX2 has smaller memory than desktop PC. Since one of RTABMap's main contribution is memory management, lots of parameter tuning is a important task for real world test.

7.2.2 Power Limitation

Limited battery power is always critical limitation in mobile robot. Long wire connection can interfere robot movement and make error large. For large scale SLAM, robot should have large size battery. And it is also related to computation performance, since non optimal calculation means waste of power.

REFERENCES

[1] M. Labbe and F. Michaud, "Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013.

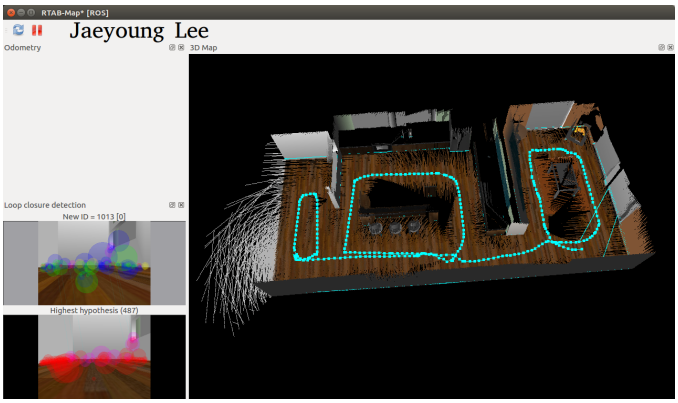

Fig. 12. RTABMapViz

[2] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2661–2666, Sept 2014.
[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008.
[4] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, pp. 6565–6574, July 2017.

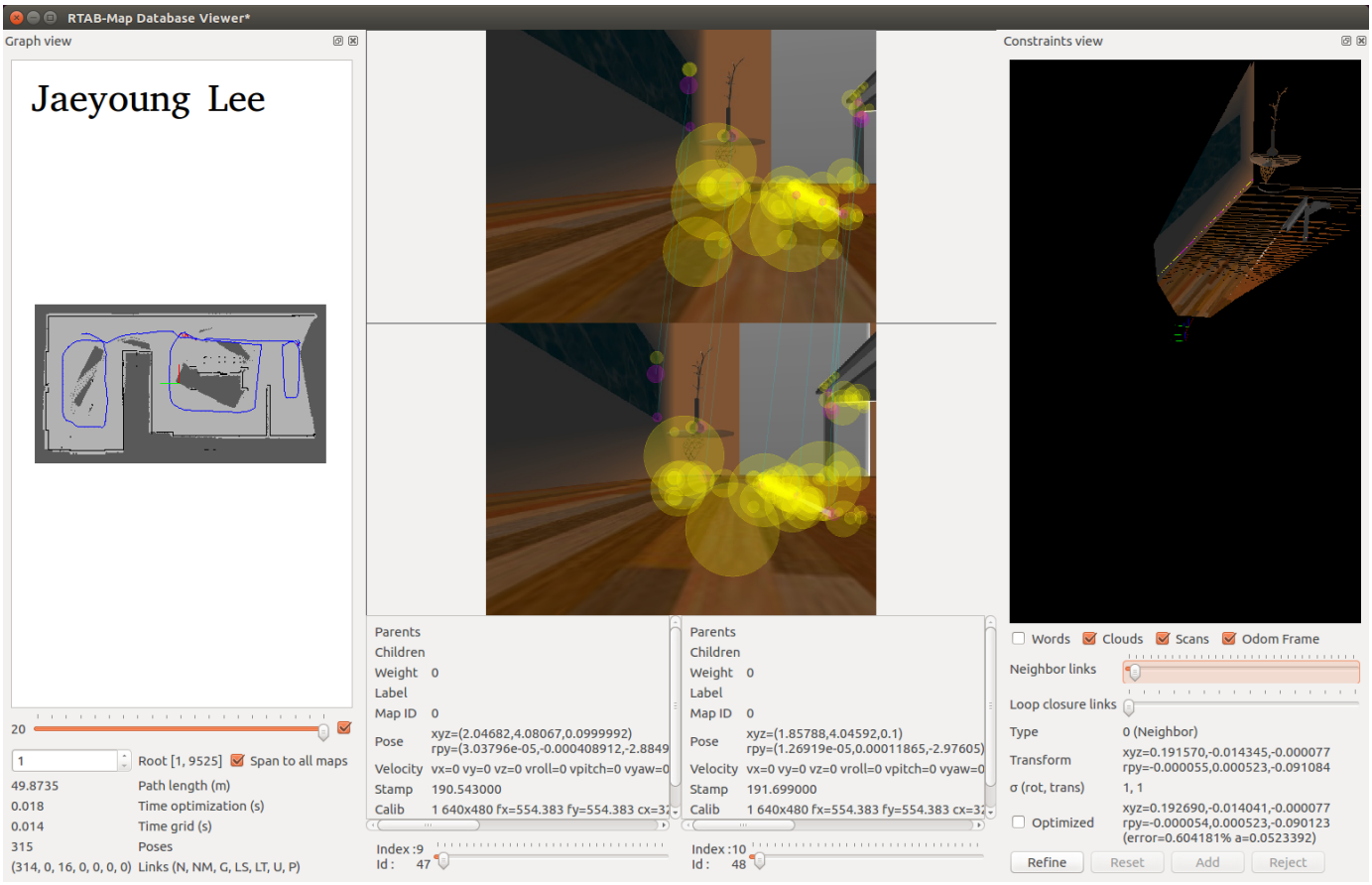

Fig. 13. DataBase Viewer

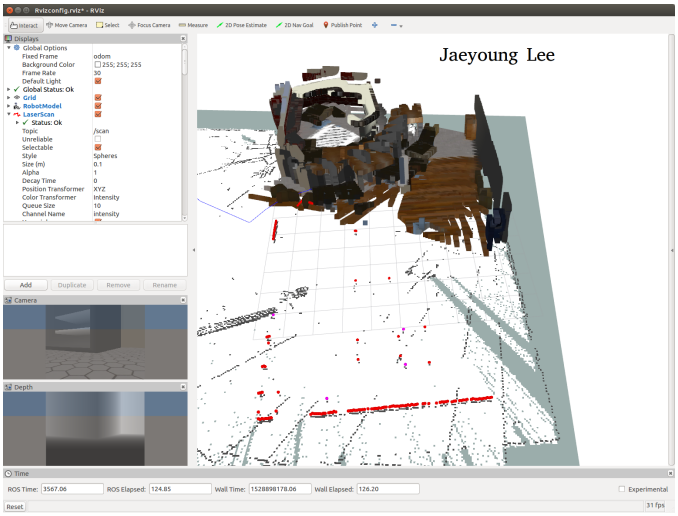

Fig. 14. Loop closure detection failure (False positive)

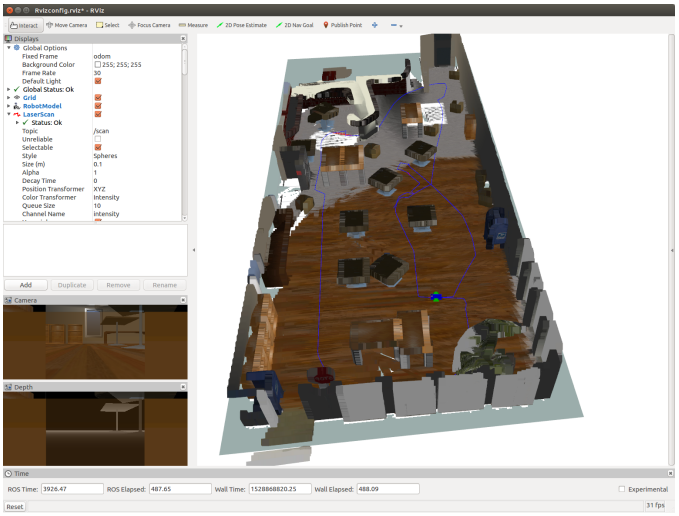

Fig. 15. 3D Map for Customized Cafe Model

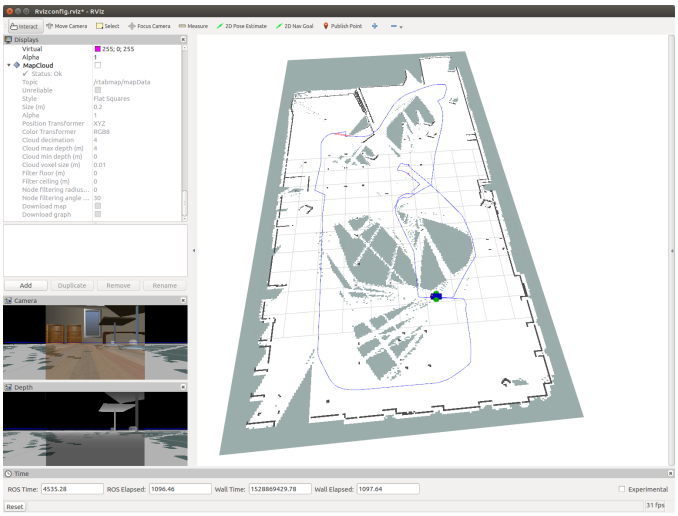

Fig. 16. 2D Map for Customized Cafe Model

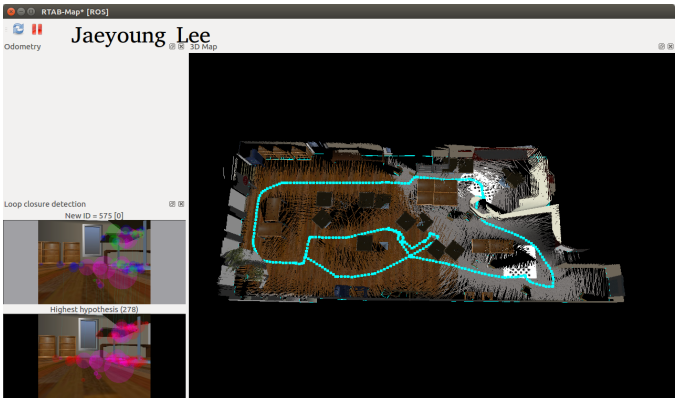

Fig. 17. RTABMapViz for Customized Cafe Model

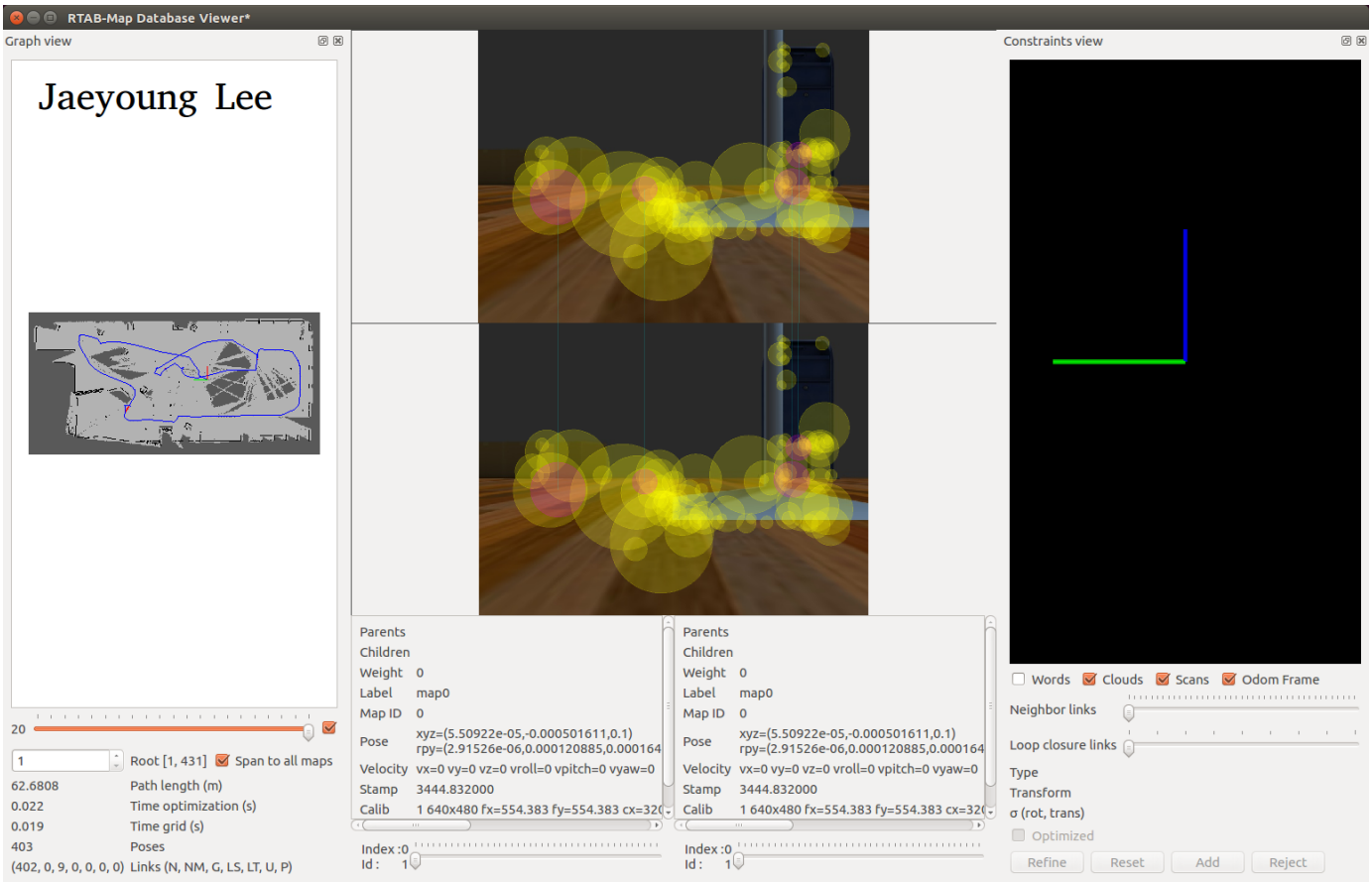

Fig. 18. DataBase Viewer for Customized Cafe Model