

Realsense2 RTABMAP

Jung, Myoungki

Abstract—A simultaneous localisation and mapping(SLAM) has been a classic complex problem topic in therobotics field. Realtime apperance based mapping (RTABMAP) package shows its approach to tackle down the using the data from vision based sensors. The purpose of this report is to show the application of off the shelf localisation package inin a simulated condition, robotics operating system(ROS), its implementation, tunning parameters and results.

Index Terms—Robot, IEEEtran, Udacity, ROS, RTABMAP, SLAM.

1 INTRODUCTION

W1H , when and where and how, is the main component of a task for any job delegated agents, humans as well as robots. The topic solving the where the agent is simultaneous localisation and mapping. SLAM is an essential tool for discovering never visited area, or continuously changing working area.

2 BACKGROUND

Classic SLAM algorithm has been evolved to many variations. However, Grapsh SLAM was used as its accuracy and capability, also adavantages over other modern SLAM algorithms.

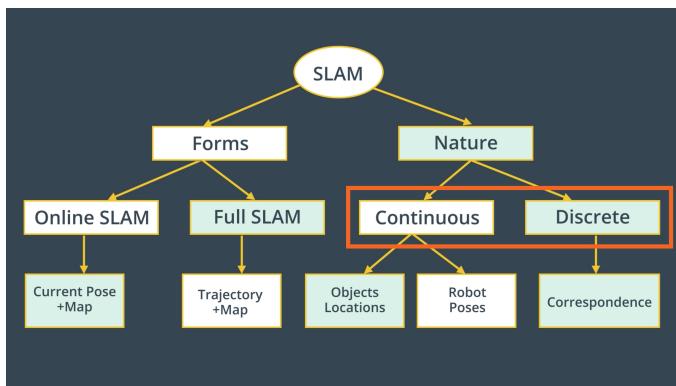


Fig. 1. Category of SLAM

[1]

Figure 1 shows the category of SLAM algorithms.

2.1 Fast SLAM

Fast SLAM is a full and online slam which uses monte carlo localisation(MCL) + Low-Dimensional EKF to solve SLAM problem. Its version 1.0 and 2.0 both also low dimensional extended kalman filter. Version 2.0 is more efficient than the version 1.0 by using different distribution with a low number of particles than the predecessor. Figure 2 shows the relation of fast SLAM category. The Grid based FastSLAM is an extension of FastSLAM using grid maps and this SLAM offers non landmark based SLAM by using MCL,

for sample motion and importance weight, and occupancy grid mapping for map estimation.

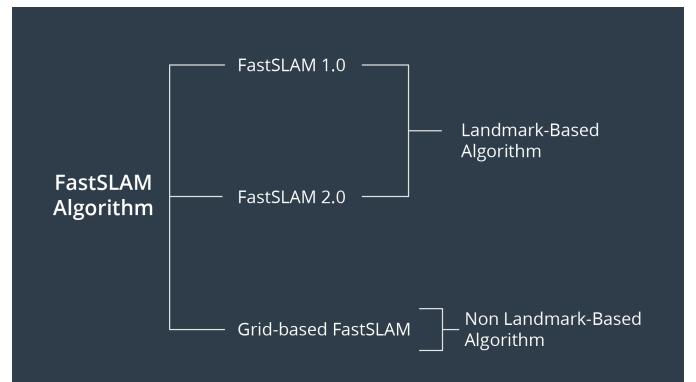


Fig. 2. Category of SLAM

[1]

2.2 Graph SLAM

Graph SLAM can solve a full SLAM, covering the entire path and map. The SLAM is consist of Two Tasks, Front-end and Back-end. Algorithm constructs the graph using the odometry and sensory measurements collected by the robot in the Front-end. he Back-end task optimizes the graph. Maximum Likelihood Estimation(MLE) performs the graph optimization by applying an optimization algorithm such as stocastic gradient descent.

2.3 Real Time Appearance Based Mapping (RTABMAP)

RTABMAP is an open source library implementing loop closure detection with a memory management approach to support online requirements for long-term and large-scale environment mapping [2]. Infolab of RTAB-Map published a cross-platform standalone C++ library and a ROS package. Inputs from scanner data to visual images are processed by RTAB MAP and results a map (2D or 3D) and optionally occupancy grid, point clouds.

3 SCENE AND ROBOT CONFIGURATION

3.1 Robot

The agent model, shown in Figure 3, is a slight modified version of previous localisation model, racerbot, with an heightened camera support on the chassis.

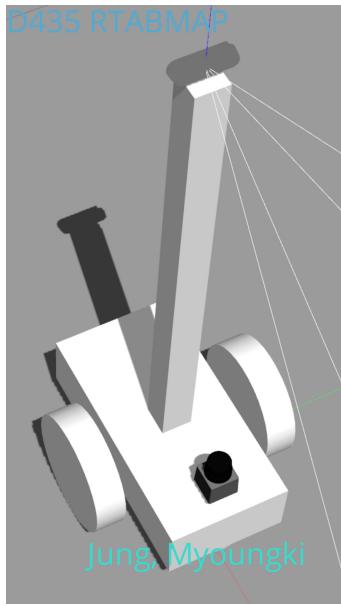


Fig. 3. Agent model

3.2 Sensors

The only visual sensor, realsense D435 plugin was used and hokyuo scanner was not used but not removed. Due to usage of d435, hokyuo's scan data was replaced by the depth_registered/points topic from the intel D435 RGBD camera. The used sensors are shown in Table 1.

TABLE 1
Model Design

Atribute	Value
Agent	0.15 m by 0.10 m
Camera	RealSense D435
Scanner	hokuyo

3.3 Packages

The list below contains the packages used in this project

- 1) Move_base
 - 2) RTABMAP_ros
 - 3) RealSense
 - 4) standalone_nodelet for converting mm to m
and Depth to depth_registered/points
 - 5) teleop package

Launch files, `world.launch` and `mapping.launch` creates the package layout shown in Figure 4. `move_base` and `amcl` also were used for SLAM in ROS. However, `amcl` is not a neccessary package for localisation as `rtabmap` provides localisation mode too.

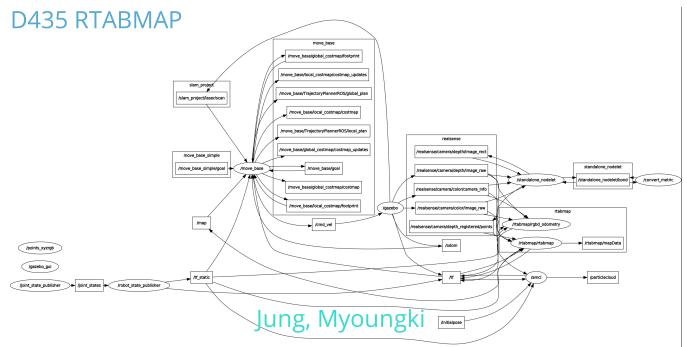


Fig. 4. rqtrosgraph view

3.4 Launch Process

The List 1 shows how to start mapping. The previous data base of rtabmap is erased when mapping starts. The mapping by teleop package is shown in Figure 5. Mapping can be done with rviz by assigning goal pose. However, the move package can maneuver in unexpected ways and corrupt the entire map easily.

```
1 $ rosrun slam_project world.launch
2 $ rosrun slam_project mapping.launch rtabmap_args
   :="--delete_db_on_start" rtabmapviz:=false rviz
   :=true
3 $ rosrun slam_project teleop
```

Listing 1. RTABMAP mapping launch command

The difference from the original mapping parameters from the provided files from udacity is listed on The Listing 2. The occupancy grid map easily duplicated features and corrupted the integrity of the grid map. Therefore the limits for 'LinearSpeedUpdate' and 'AngularSpeedUpdate' was set to remove the corrupting artifacts while moving slower than 0.2 m/s or 0.2 rad/s. I addition, the RGBD camera's 'MaxDepth' was limited to 3.5 m as more than this range is very noisy.

```
1 <param name="RGBD/LinearSpeedUpdate" type="string"
      value="0.1" />
2 <param name="RGBD/AngularSpeedUpdate" type="string"
      value="0.1" />
3 <param name="RGBD/LoopClosureReextractFeatures" type
      = "string" value="true" />
4
5 <param name="RGBD/ProximityBySpace" type="string"
      value="true" />
6
7 <param name="Kp/MaxDepth" type="string" value="3.5"
     />
```

Listing 2. Diff RTABMAP mapping.launch

The listing 1 shows code enabling localisation from the existing map from the mapping process. In localisation mode, the working memory is initialised with the previous map data and does not overwrite on it during robots' navigation. localisation was tested by moving robot randomly and start the localisation mode.

```
1 <param name="Mem/IncrementalMemory" type="string"
      value="false" />
<param name="Mem/InitWMWithAllNodes" type="string"
      value="true" />
```

Listing 3. Diff RTABMAP localisation.launch



Fig. 5. Rtabmap mapping process

```
$ roslaunch slam_project world.launch
$ roslaunch slam_project mapping.launch rtabmapviz:=
false rviz:=true
```

Listing 4. RTABMAP localisation launch command

The shell commands in List 1 starts localisation mode shown on Figure 6.

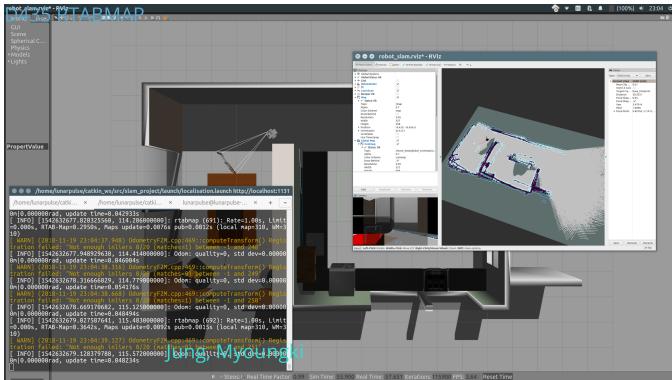


Fig. 6. Rtabmap localisation process

4 RESULTS

4.1 Benchmark scene - Kitchen and Dining

The benchmark world kitchen and dining imposed some difficulty in creation of occupancy grid mapping. The turning velocity had to be very slow. With the provided teleop package, a satisfactory level of details in mapping could be achieved. The result of mapping process in kitchen and dining world in rtabmap database viewer is shown on Figure 7. The number of loop closure is 46 and has total 309 local map. The features in the scene, chairs, tables, lamp, sofa, and etc. can be recognisable from the occupancy grid map. The map view shows clear single lined walls of the building. If it was normal kinect, instead of realsense_gazebo_plugin, it could be easier to achieve cleaner map and occupancy grid map. The realsense_gazebo_plugin used for this project has flaws and instability, resulting jumps in odometry, excessively slow in complex scene. The challenge of mapping was how patiently move and planning before moving the agent. Localisation was successful, Figure 6,

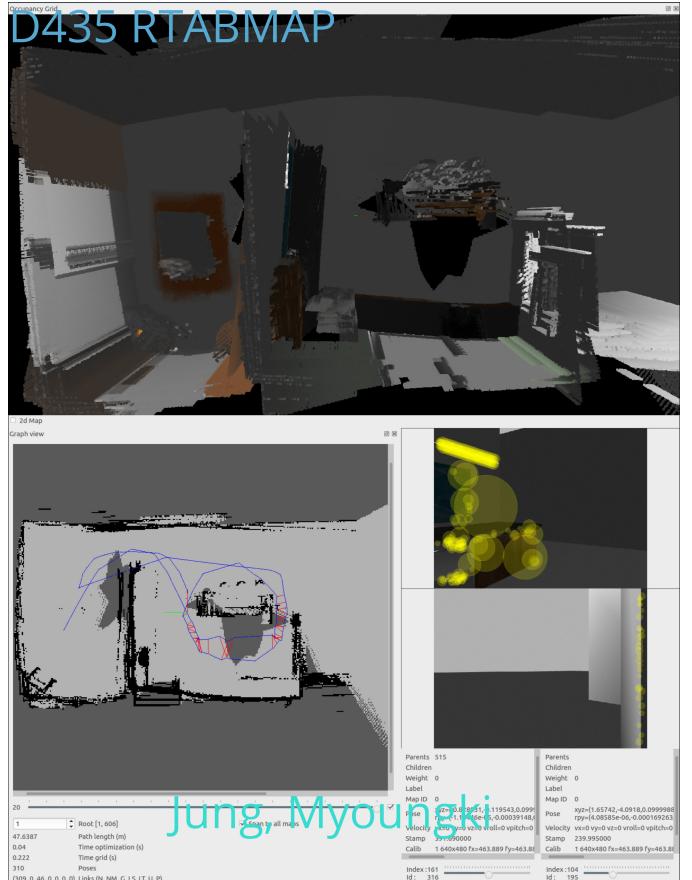


Fig. 7. Kitchen and dining rtabmap data base view

even with kidnapped robot scenario and this shows the strength of rtabmap.

4.2 My model - gazebo

/world/gazebo.world contains second custom world file for this project, gazbo.world. The gazbo.world has rich visual features, due to the trees, and makes it easier to SLAM compared to the benchmark kitchen and dining world. However, processing of tree branches and leaves slows down the overall ROS system occasionally. The result of mapping process in this scene was saved in . Figure 8 shows the rtabmap database view of the mapping result in gazebo world. The details of objects in the scene are clearly shown, SUV, walking person, car, rag doll, gazebo, fountain and the exact number of trees. Mapping in this world was done by rviz and performed SLAM instead of mapping. The agent was able to map the new environment and localise itself inside it using the obtained map. The number of loop closure is 7 and it was intentionally avoided to have too many loop closures by not visiting the same place many times to avoid the creation of duplicated features in occupancy grid map.

In both scenarios, loop closures were observed in rviz. The map data shown in rviz was updated and this shows map data of rtab was updated prior to rviz. The database files are bigger than normal created by kinect sensors, as the size of scan data from d435 is much larger than the one by hokuyo and this larger scan data makes SLAM database bigger.

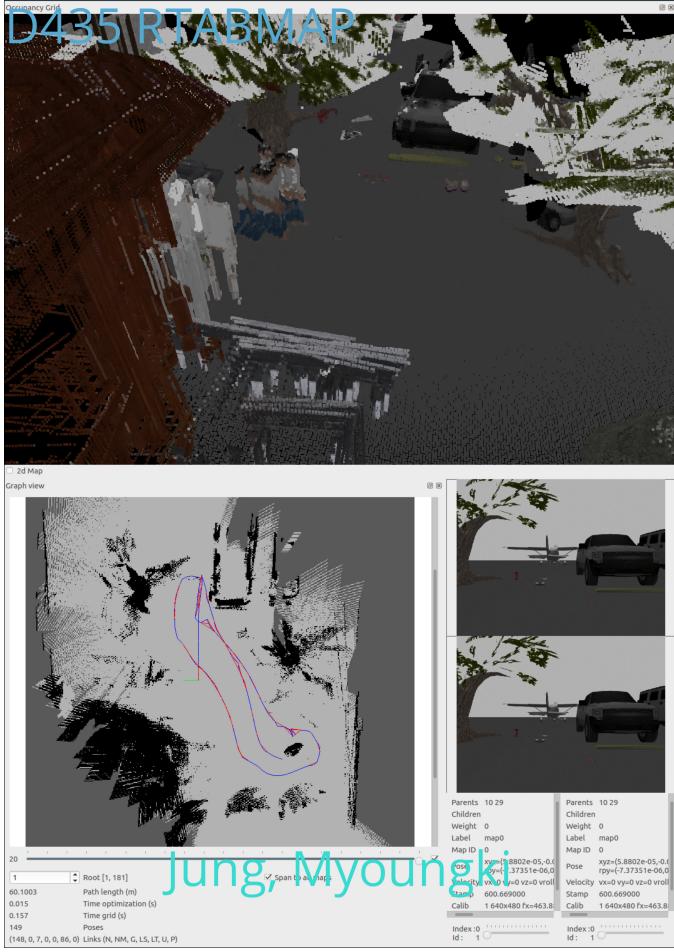


Fig. 8. gazebo rtabmap data base view

5 DISCUSSION

The tuning process of the SLAM was non-trivial task and requires understanding of many parameters, the incomplete and unstable `realsense_gazebo_plugin` added more difficulty. The implication of loop closure for healing every problem in mapping should be discarded. The mapping must be carefully done but not expecting loop closure function to fix poor mapping process. The loop closure should be only used to recover from disrupted map due to the mapping errors, odometry drift, IMU drifts, sensor noises. In other words, speedy mapping required by fast autonomous vehicles, drones, uav, ugv, and autonomous flying cars, is far from the current technology and this has been solved for the required technology.

6 CONCLUSION / FUTURE WORK

The RTABMAP tends to create a displaced occupancy grid map and persists the wrong artifacts for long time. Once this happens, an agent need to map from the beginning again. There might be parameters prevent this happening or allow less occurrences, the documentation on RTABMAP is not sufficient to find it easy. RTABMAP should take this to be accounted and make the parameter tuning more straight forward for the users' precious time and effort. RGBD still

contains much errors in output point clouds. And this propagates to the final SLAM result. RTABMAP is just as good as collision avoidance level accuracy not for fine maneuver purposes.

Drones can construct 3D maps of the rescue mission in dangerous area, such disaster zone, underground working area, radio activity influenced area. However, the zone specific challenges exists, visibility for visual odometry, richness of textures also affects the choice of slam, RTABMAP can only be used in optimal visual condition filled with rich textured objects. This can be still used with a good environment conditioning, sufficient ambient lights and clear atmosphere but not for heavy industrial, dusty and monotonous area. From these limitation of visual based mapping, home service unit can be suited best. Another one noticeable is that RTABMAP uses SURF or SIFT, visual feature matching algorithms. With the recent progress in machine learning, especially neural networks, those image recognition methods can be replaced with well trained CNN or voxnet, 3d volumetric convolutional neural network [3] for specific environment to improve accuracy and range of the feature recognition. In addition, deep reinforcement learning can be a good candidate for new SLAM approach.

REFERENCES

- [1] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, pp. 1999–2000, 2002.
- [2] M. Labb   and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, 2018.
- [3] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," 2015.