Map My World Robot

Xueyang Kang

Abstract—In this report, the Real-Time Appearance based simultaneously Localization and Mapping (RTAB SLAM) Ros package is integrated onto a robot in simulator, to map a provided environment, and the customized Gazebo world. And the associated robot and scene configuration work are also clarified. At the end part, the generated 3D map and 2D occupancy map are presented, and the loop closure parameters are analyzed to give reader a clue to the relationship between mapping process and the final result. The comparison between the two worlds and the future work are discussed at last.

Index Terms—RTAB, SLAM, Gazebo, loop closure, 2D Occupancy map.

1 Introduction

Appearance based SLAM is via extraction of features lying in images, then mapped into description in bag of words, to find the similarity in different temporal images. And the detection of revisiting a location can be used by loop closure to correct the visual odometry. But how robust is this appearance based method, and what kind of scenes can be serving as an ideal input to implement RTAB SLAM, and how mapping parameters affect the final result? all theses questions will be solved in this report. In this report, the experimentation was carried on in a Gazebo simulation world with configurable settings, this facilitates the testing convenience of testing algorithm performance, especially, the loop closure, which is a core block of RTAM SLAM, will be explored as well.

2 Background

In reality, to navigate a robot on a ground, both the 2D and 3D maps are required to help the robot to be aware of where it is, and find the interest in 3D world, and interact with the surroundings based on semantic analysis.

Additionally, how to design an efficient, accurate and robust mapping algorithm, to tackle the mapping in physical world is always a challenging and active issue in robotic navigation field. How to perform a mapping in a cluttered environment or an environment with dynamic objects like human-beings, and how to maintain a robust mapping in a large scale scene, these problems are still open issues to bring the robot closer into real world, full of uncertainties.

Through KF, EKF, PF, until the graph based SLAM in recent years , more and more efficient and high-performance algorithms are proposed, graph-based SLAM can either utilize the geometric or appearance feature to establish the association between frames, i.e., LSD-SLAM, a SLAM based on both density and intensity of pixels, ORB-SLAM based on ORB features, and RTAB SLAM [1], which is compatible with different types of 2D and 3D sensors, RTAB Ros package can provide both the 2D and 3D maps. To note is that, this new developed algorithm can achieve a good trade-off between computational time and accuracy in a large scale mapping task, this thanks to its hierarchical memory management [2], and its robust loop closure detection [3], the algorithm can even stitch multiple session mapping into a consistent global map. All these advantages renders it a promising tool for robotic navigation.

3 Model Configuration

The robot model inherits the configuration from the last localization project, with the addition of the 2D laser and replacement of the common RGB camera with RGB-Depth camera. the robot is still differential drive based and the size is unchanged, because the small body size with a rough radius 0.3 meter , facilitates the agility of robot's motion, to be able to pass through some narrow areas. The position of laser is at a height of 0.6m over the ground to avoid the wrong measurement caused by the robot's big wheels. The camera is still mounted in the front of body with a little increase of height to a total 0.4m, this can be helpful for the robot to get a good front view of both the distant scenes and the ground in proximity. Figure 1 demonstrates

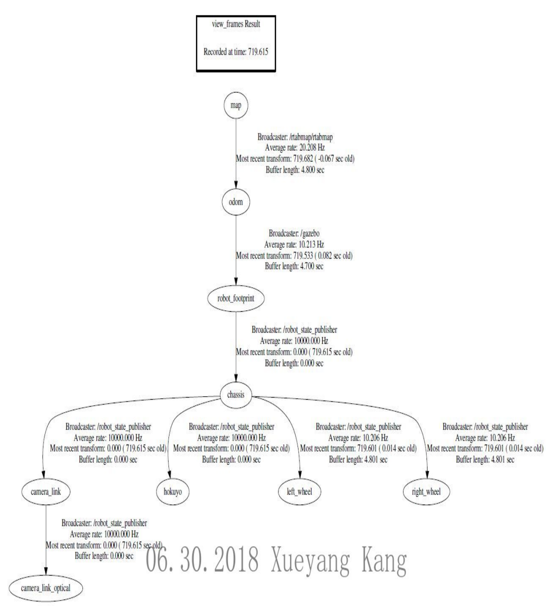


Fig. 1: Frame connections

the link connections of different frames, from "map" to "odom" all the way up to robotic base link, then all the camera frames and the wheel frame are all connected to the "chassis" frame, particularly the "rgbd_optical_frame" is applied to correct the coordinate inconsistency between rviz and gazebo world, so the final images are shown with respect to this frame.

4 World Creation

The provided gazebo world includes some basic furnitures in a kitchen and dining room, while the customized world is built upon a gas station with sun light and ground plane, to create an outdoor environment with enough features. The three customized side walls plus gas station stop are joined together to form a closed space on the ground. Then to make some distinctive features along the corridor and the side walls, some objects with regular shape are distributed in the whole space. The folders in customized package is stacked up in a following oder.

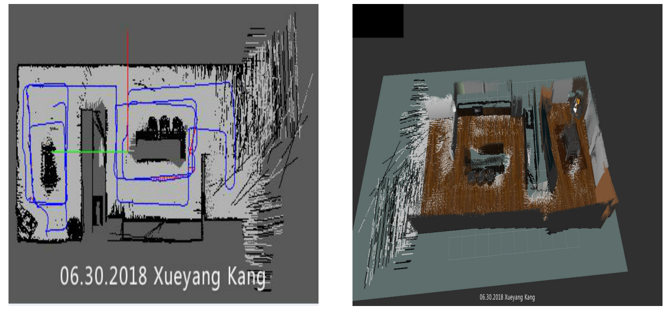
TABLE 1: layout list

Model	Layout
Grey wall	At the fringe to form two sides
Construction cone	In middle of a corridor
Construction barrel	Connecting to brick to form one side barrier
Mail box	At a corridor end, near Grey wall.
Cafe table	In the corner
Fire hydrant	Near gas filling machine
Jersey barrier	In the other corner
Postbox	Near Grey wall
Book shelf	Along the Grey wall
Dumpster	Along the Grey wall

1) "config", configuration of the rviz
2) "launch",including launch files for mapping localization, and loading of the robot model, tele-operate node to command the robot movement, rviz launch file, file to load the gazebo world.
3) scripts, Teleoperate node file and the bash script running a batch of commands in terminal for launching nodes.
4) src, source file for publishing the goal position in autonomous mode.
5) urdf, robot plugin loaded in gazebo, and model description file.
6) worlds, containing the gazebo description file for provided and customized world.

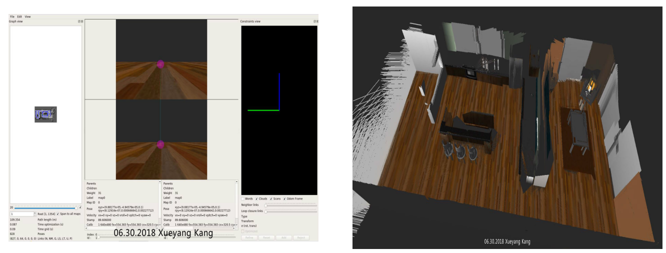
5 Results

Here the final 2D occupancy grid map and the 3D point cloud map for both worlds are displayed. At Left of figure 2a is the 2D occupancy grid map for default world, here the robot passed through the kitchen three times before entering into the dining room, each traversing loop closure is with different radius to map the characteristics of surroundings as complete as possible. The generated 3D map via RTAB SLAM is Figure 2b, the structure is recognizable, in particular the local map in kitchen is consistent, while the map of dining room is not so well,with some obvious mismatch existing. This is due to the blanket placed in the center ground of the room, and during mapping sometimes one wheel of the robot was on the blanket while the other one still on the ground, the uneven surface induces the odometry errors in the map.



(a) 2D occupancy map (b) 3D point cloud map

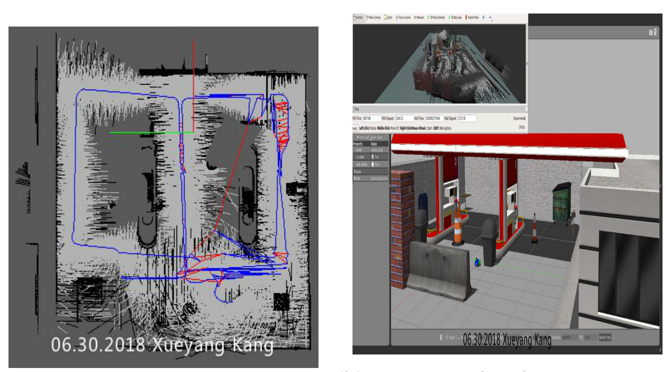
Fig. 2: 2D and 3D map for provided world



(a) Loop closure in database (b) Optimized 3D point cloud

Fig. 3: Database-viewer & optimized map for provided world

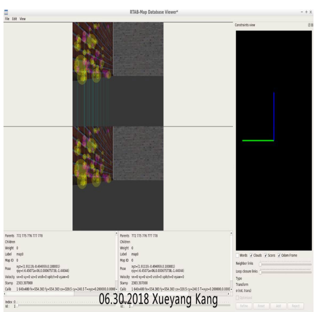
Figure 3a shows the 2D graph and constraints number in the database viewer, the number of detected loop closure is 827, this is a enough number of geometric constraints, and can be applied to optimize the 3D map. On the right is the optimized result, from the top view, the structure in both rooms are quite clear, even the mismatch in the dining room is corrected.



(a) 2D occupancy map (b) 3D point cloud mapping process

Fig. 4: 2D and 3D map for customized world

Figure 4a is the 2D occupancy grid map corresponding to customized world, this outdoor environment is more complex, with three vertical corridors and two horizontal corridors, as mapped in 2D graph, the gird map is precise in general. The right part is the 3D mapping process in gas station world, the 3D point cloud is only a partial of the structure, because here only the 2D closed area is visited, otherwise two much time will be spent on mapping such a big environment, in experimentation, the robot is moving at a linear speed at 0.3ms, angular speed at 0.3 rads, this relatively low motion speed can make sure the SLAM being updated timely with completion of the matching process.



(a) Loop closure in database

(b) Optimized 3D point cloud map

Fig. 5: Database viewer and optimized map for customized world

Around 1000 loop closures are found in customized world, this is due to a longer traversing trajectory compared with the mapping in provided world. Figure 5b is the final optimized 3D point cloud, the mapping area is consistent and the closure 3D space with distributed objects is recognizable/

6 Discussion & Future Work

6.1 Discussion

The robot movement is with enough flexibility, but the camera view is limited instead of a 360-degree view, and the robot needs a long time to observe a complete 3D component from multiple viewpoints.
The parameters regarding the linked graph node update and the threshold of feature detection, and inlier of loop closure association are vital for mapping task, the parameter tuning work is verified by the generated database loaded into the RTAB database viewer, more loop closures occur, meaning more pose constrains are induced, and the post-optimization can smooth over the entire point cloud. In addition to the algorithm relevant tweaking work, the robot is moving at a relatively low linear and angular speed, to leave enough time for map update, in particular, the possibility of mismatch happens frequently at turning, the feature extraction method adopted in RTAB SLAM may not be invariant to rotation, so the big motion between adjacent frames will result in a failure of association, and it should be avoided. During mapping the rviz is used to visualize the incremental built-up process, if some unexplored area exists or mismatch occurs, the trajectory of robot is changed accordingly to resolve these problems. The whole procedure repeats until the enough loop closures are detected and the map is complete and consistent along each dimensionality intuitively.
From the results, the mapping in the indoor structure is better than the process done in the outdoor gas station, although the inconsistency exists in partial area of map given the provided world, this is due to the uneven ground as mentioned before.
Because the path in customized world is longer, and the world contains more corridors and more complex and irregular structures, the ratio of objects with distinctive features to the total 3D space is quite small, so the real-time performance of SLAM in customized world is unsatisfactory, but the post-optimization can process the point cloud to generate a more consistent map, in sacrifice of computing time.

6.2 Future Work

RTAB-MAP is a powerful tool to be used for large-scale mapping, this can be helpful in the inspection and security filed. Like the robot integrates the SLAM functionality, can maintain a long-term map and use multi-session mapping to take advantage of history mapping efficiently. In logistic branch, some big warehouse needs to be inspected to keep safety of the inventory, these kind of application involves a mapping in a scene with slow change over long time span, the robot having RTAB based SLAM can operate in such a scenario without too much maintenance work needed from human-being, therefore, the cost can be reduced a lot compare to labor-work. The 2D occupancy map can help the robot to move freely in free area, and the 3D map combined with higher level processing enables the robot to do classification and detection task, like alerting the unauthorized entry of strangers, the robot can also contribute to the goods management, but this means the whole navigation system including mapping, localization, path planning, recognition, controlling should be set up properly to provide more intelligence to robot itself.

References

[1] M. Labbé and F. Michaud, "Long-term online multi-session graph-based splam with memory management," *Autonomous Robots*, vol. 42, no. 6, pp. 1133–1150, 2018.

[2] M. Labbé and F. Michaud, "Memory management for real-time appearance-based loop closure detection," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1271–1276, IEEE, 2011.

[3] 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.