Indoor 3D Mapping Using RGB-D Camera

Nagarjun Vinukonda
Robotics Engineering
Worcester Polytechnic Institute
nvinukonda@wpi.edu

Vrushabh Desai
Robotics Engineering
Worcester Polytechnic Institute
vmdesai@wpi.edu

Kavit Nilesh Shah
Robotics Engineering
Worcester Polytechnic Institute
kshah@wpi.eduy@wpi.edu

***Abstract*—In this paper, we attempt to implement the Real-Time Appearance-Based Mapping (RTAB-Map) through generating 2D occupancy maps as well as 3D point cloud of the environment as per the requirement. For this, we plan to use an RGB-D camera such as Microsoft Kinect to perform RGB-D SLAM in indoor environment. This method of creating 2D occupancy maps or 3D point clouds of the environment could as well be implemented for performing SLAM on indoor robots such as a cleaning robot integrated with an RGB-D camera.**

***Index Terms*—SLAM, point cloud, RGB-D, Kinect**

I. Introduction

There are different ways available for localizing the robot in an unknown environment, using visual odometry or SLAM(Simultaneous Localization and Mapping) with sensors like Li-DAR, IMU, etc. But, in this paper we are working on generating robust 2D and 3D maps for an unknown environment using an RGB-D camera as shown in Fig. 1. Our approach is based on using monocular Kinect Camera for extracting features in real time and combining this data with depth features.

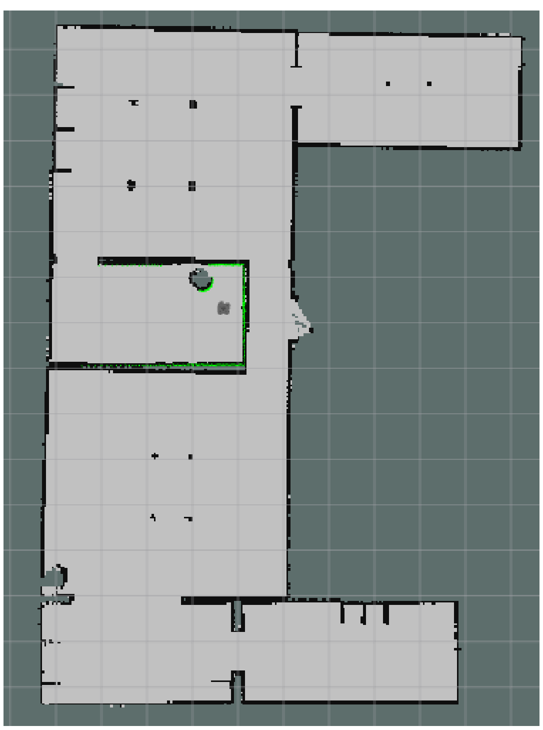

Fig. 1: 2D grid map

One of the most famous methods for sensor data registration and modeling is RANSAC(Random Sample Consensus) algorithm. In some vision-based SLAM algorithms, RANSAC is an efficient solution for image matching procedure and it establishes the data association among different views. There are two methods available for image matching: The dense and sparse way. The dense way uses the complete image for parameter estimation while the sparse away uses SIFT, SURF or any other feature detection algorithms to detect features and these features are matched by using the distance between feature vectors, where sometimes the matching pairs are ambiguous. To remove these mismatching errors and efficiently achieve a correct matching result algorithms such as M-estimation, LMedS (Least Median of Squares), or RANSAC are used. Among the following methods RAN-SAC was estimated as efficient algorithm [1].

In this paper we are implementing ICP(Iterative Closest Point) for 3-D point cloud Matching. The way it works is that it tries to minimize the distance between two point clouds as error metric. Thus it calculates the net translation and rotation (the net transformation matrix) between two point clouds. ICP is (in general) only locally optimal i.e. can get stuck in local minima.

Loop closure(or cycle closure) is the last step in SLAM which is necessary for obtaining a consistent solution. Loop closure is when one of the frames matches with one of the initial frames. It is important in SLAM as it helps to remove the accumulated drift from the data. In order the save the computation of checking each new frame with all of the previous frames, a better alternative is to check every nth new frame [2].

Over last decades most of studies are concentrated towards 3D SLAM, since there is no longer available systems to use 2D SLAM for mapping in environments [3].

II. Literature-Review

RTAB mapping is based on an incremental appearance-based loop closure detection. RTAB map is usually combination of graph optimization and loop closure detection algorithm. [4] used RTAB SLAM on real-world data sets (eg: KITTI dataset) and made a comparative analysis outlining limitations and strengths of Lidar based and visual based SLAM approaches used in practical applications of Autonomous Industries. [5] made a comparative experimental study between RTAB and Kintinuous, in which it is concluded that RTAB is more preferred. In this paper, kintinuous is more dependent on depth data which makes it less reliable while using RGBD data, where RTAB uses RGBD data more effectively.

[6] made a comparison study on ROS based visual SLAM methods in indoor environment which are ORB-SLAM, RTAB-Map, stereo ZedFu and monocular DPPTAM. The main conclusions from there experiments are there are no a monocular SLAM algorithms which make an estimation of localization. Hence, it is required to use ground truth for verification of camera displacement absolute value or size of map objects. They also made conclusion that Stereo cameras or RGB-D sensors gives builds map, a localization in absolute values and provides good results while in usage. If we require to build a map using high depth it is more preferable to use stereo camera.

III. Environment and Robot Configurations

Robot model that we used for our project is turtlebot3 which has 360 LiDAR sensor, RGB-D camera. The differential drive plugin is configured in the navigation launch file to publish control commands to the /cmd vel topic and odometry messages to the /odom topic. The camera plugin is configured to publish raw RGB images to /camera/rgb/image raw and raw depth images to /camera/depth/image raw. The laser plugin is configured to publish messages of type LaserScan to the /scan topic. Graphical view of the ROS topics and nodes is shown in Fig.2. and the robot model is in Fig.3.

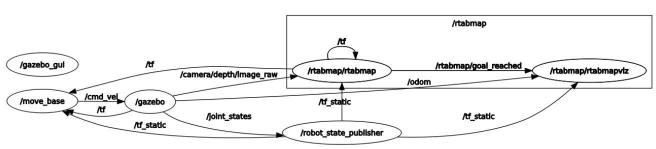

Fig. 2: RQT graph of the all the topics

Fig. 3: Turtlebot3 ros package

The ROS package RTAB-Map is deployed on the robot to perform SLAM of indoor environment. The environment is a is a default environment of house named turtlebot3 house as shown in Fig. 4.

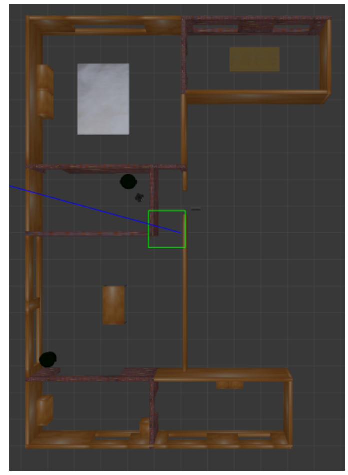

Fig. 4: House Environment

IV. Methodology

This project aims to construct a 3D map using RGB-D camera(Kinect). The sensor data i.e. extracted features from camera are extracted two different ways: sparse images using RGB and point-cloud using Depth feature of camera. Later, image matching is done for sparse images using RANSAC algorithm and point cloud data matching is done by ICP (Iterative closest point) algorithm. Further, global optimization takes place for mapped images in order to remove discrete noisy cloud of points with weak information on orientations and as ICP is locally optimum. The loop closure is used to help robot understand it has navigated through the map and then uses that data to generate map. Also, this property will make sure the feature mapping stops after images are extracted and mapped after a loop. The data yielded after mapping and optimization is visualized in 3d point cloud map which will be further visualized in occupancy grid. For general overview the procedure in detailed in a flowchart as shown in Fig. 5.

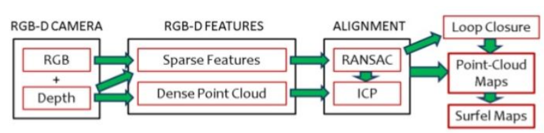

Fig. 5: Flowchart for 3D mapping

A. GMapping

One of the most common and widely used SLAM algorithm is the GMapping algorithm. This algorithm is to create the map of the environment and also estimate the robot's pose inside the map. The GMapping algorithm uses Particle Filters (PF) for localizing in the environment. This is unlike the 'karto' algorithm which uses Kalman filters. These particle filters help to give a probability distribution model of the robot's position in the map based on the control inputs. These particle filters are implemented along with motion models(odometry readings) as well as the sensor models. All the three components together on detecting loop-closure, optimize the error in the map while driving it to zero. The following figure is our rqt_graph for gmapping as shown in Fig. 6

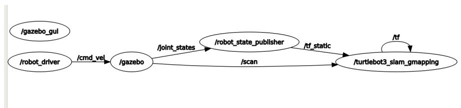

Fig. 6: RQT Graph of our gmap

B. RTAB-MAP

Another one of the most popular SLAM algorithms is the RTAB-Mapping which is Real-Time Appearance-Based Mapping. The main reason for its popularity can be partly due to its flexibility to be applied to many different camera systems such as RGB-D, Stereo, Lidar, etc. It is a graph based slam, meaning that the ech new detected feature becomes a new node being added to the graph with edges being the transformation between two or more nearby nodes. When loop-closure is detected, it adds new constraints to the system which optimizes the errors in the graph. Also it uses smart memory management system to limit the number of loop-closure points.

C. Coding Libraries

Point Cloud Library (PCL) and will be visualised using the ROS Visualization Tool (RViz). PCL is an open-source library which is used to process tasks in 2D and 3D and contains algorithms for feature estimation, model fitting, and data segmentation. RViz is a 3D visualization tool used for displaying sensory and live state data from Robot Operating System (ROS). Kinect Camera is connected using C++ libraries in ROS.

V. Results

In order to navigate the robot around the environment we used custom teleop keyboard script. In order to have a better loop closure detection, the robot was navigated through the whole environment in multiple direction. Through navigation we have the following results of RTAB-map as shown in Fig. 7 and Fig. 8. In the graph the blue dots represent our turtle bot navigation path.

VI. Conclusion

As a result of the project one thing which we observed was int the computational efficiency. Gmapping being 2D occupancy based SLAM algorithm is computationally light as compared to RTAB mapping which is graph based. Also another striking difference was in loop closure detection. RTAB-MAP has better loop closure detection as compared to particle filter based GMapping Fig. 9

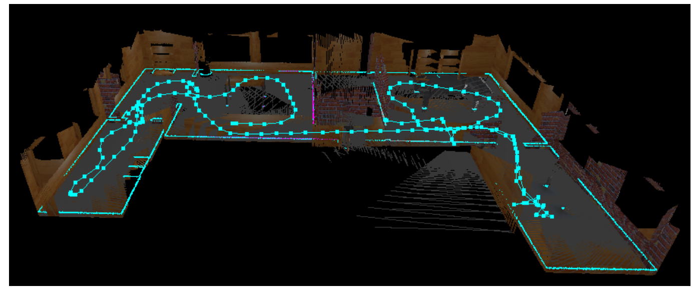

Fig. 7: RTAB map results

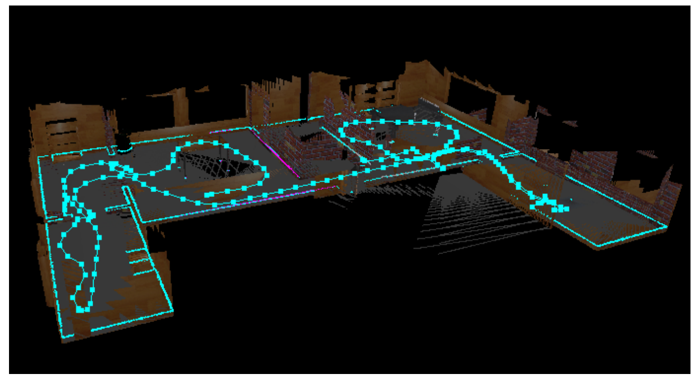

Fig. 8: RTAB map results

	Inputs							Online Outputs			
	Camera				Lidar		Odom	Pose	Occupancy		Point Cloud
	Stereo	RGB-D	Multi	IMU	2D	3D			2D	3D	
GMapping					✓		✓	✓	✓		
TinySLAM					✓		✓	✓	✓		
Hector SLAM					✓			✓	✓		
ETHZASL-ICP					✓	✓	✓	✓	✓		Dense
Karto SLAM					✓		✓	✓	✓		
Lago SLAM					✓		✓	✓	✓		
Cartographer					✓	✓	✓	✓	✓		Dense
BLAM						✓		✓			Dense
SegMatch						✓					Dense
VINS-Mono				✓				✓			
ORB-SLAM2	✓	✓									
S-PTAM	✓							✓			Sparse
DVO-SLAM		✓						✓			
RGBiD-SLAM		✓									
MCPTAM	✓		✓					✓			Sparse
RGBDSLAMv2		✓					✓	✓		✓	Dense
RTAB-Map	✓	✓	✓		✓	✓	✓	✓	✓	✓	Dense

Fig. 9: RTAB map

VII. Future Work

An interesting future work would be to explore the RTABMap package's visualization section in more details. The obstacle detection feature can be deployed in order to extracts obstacles and the ground from your point cloud. With this information in hand, these obstacles can be avoided when executing a desired path. Another potential area would be Wifi signal strength mapping.

References

[1] S. Jia, K. Wang, X. Li, and T. Xu, "A novel improved probability-guided ransac algorithm for robot 3d map building," *Journal of Sensors*, vol. 2016, 2016.

[2] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," *IEEE transactions on robotics*, vol. 30, no. 1, pp. 177–187, 2013.

[3] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous robots*, vol. 4, no. 4, pp. 333–349, 1997.

[4] M. Labbé 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*, vol. 36, no. 2, pp. 416–446, 2019.

[5] N. Altuntaş, E. Uslu, F. Çakmak, M. F. Amasyalı, and S. Yavuz, "Comparison of 3-dimensional slam systems: Rtab-map vs. kintinuous," in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 99–103.

[6] I. Z. Ibragimov and I. M. Afanasyev, "Comparison of ros-based visual slam methods in homogeneous indoor environment," in *2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*. IEEE, 2017, pp. 1–6.