MAP MY WORLD

ISRAILOV SARDOR

Abstract—The purpose of this paper is to present the mapping project with ROS in gazebo/rviz environment. The main objective of the project is to map the environment with a robot using teleop commands. The core package used for mapping is RTAB-Map(Real-Time Appearance-Based Mapping) that used the topics of RGBD camera as well as the laser scan to make 3D map of the environment with 2D occupancy grid mapping.

Index Terms—Robot, IEEEtran, RTAB, SLAM, Udacity, ROS, Gazebo, Rviz, Localization, Mapping.

1 Introduction

MAPPING is the process of processing the information from sensors in a meaningful way to reconstruct the 2D occupancy grid or the 3D structure of the environment.

2 Background

2.1 SLAM

Simultaneous localization and mapping is the process of mapping the environment and localising the robot inside of it. There are many variations of SLAM, notably:

1) **GRAPH SLAM** is the method of SLAM that uses sparse information matrices produced by generating a factor graph of observation inter-dependencies (two observations are related if they contain data about the same landmark)
2) **ONLINE SLAM** is the method of estimating the map and the current pose given all the measurements and controls of a robot.
3) **FULL SLAM** is the method of estimating the map and all the poses given all the measurements and controls of a robot.
4) **EKF SLAM** the variation of the slam that used extended Kalman filter. The algorithm works under the assumption that the noise is Gaussian, which makes it sensible to the noise variations.
5) **Fast-SLAM** is the method of SLAM, that used low-dimensional EKF to solve independent features of the map that are modeled with local GAUSSIAN. It estimates the posterior over the trajectory using a particle filter approach. This gives an advantage to solve SLAM with the known poses.

Briefly describe Kalman filters. Explain how they work and why they are used for localization. Additionally, discuss the drawbacks of linear Kalman filters and how Extended Kalman Filters (EKFs) help resolve some of these issues.

2D occupancy grid is used to represent the map of the environment as an evenly spaced field of binary values: occupied-1 free-0 or unknown.

2.2 RTAB-Mapping

Rtab-map is a variation of graph SLAM. It is an approach based on a global loop closure detector with real-time constraints. This package can be used to generate a 3D point clouds of the environment and/or to create a 2D occupancy grid map for navigation. Rtab-map is optimised for large-scale and long-term SLAM by using multiple strategies to know the loop-closure images. The rtab-map used the global loop closure and bag-of-words technique.

loop-closure is a detection of relations between a newly-seen image and images seen before.

bag of words is a technique used in vision-based mapping. It uses the visual features as comparison of different images.

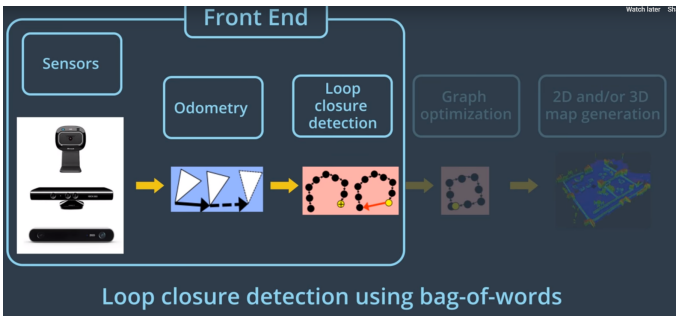

Fig. 1. graphical description of RTAB-map (credits to Udacity)

Rtab-map has a memory optimisation algorithm which uses short-term, long-term and working memories.

3 Configurations

3.1 Scene configuration

The first scene configuration used the built-in scene in gazebo **kitchen-dining** The second scene configuration was built in gazebo built-in editor and using insertion of objects **robocop.world**

3.2 robot configuration

The robot configuration was similar to udacity_bot developed during the second project. The Depth_camera link was added and the support for hokuyo sensor. Moreover, some

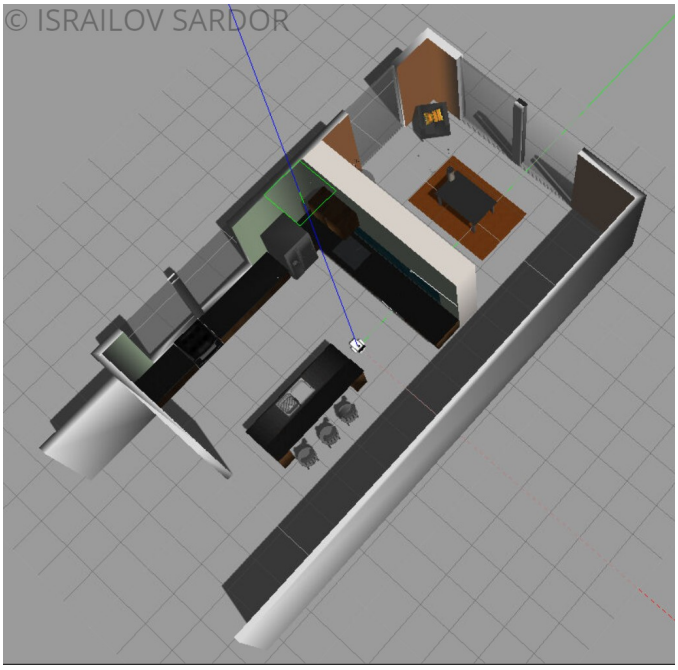

Fig. 2. kitchen_dinning world

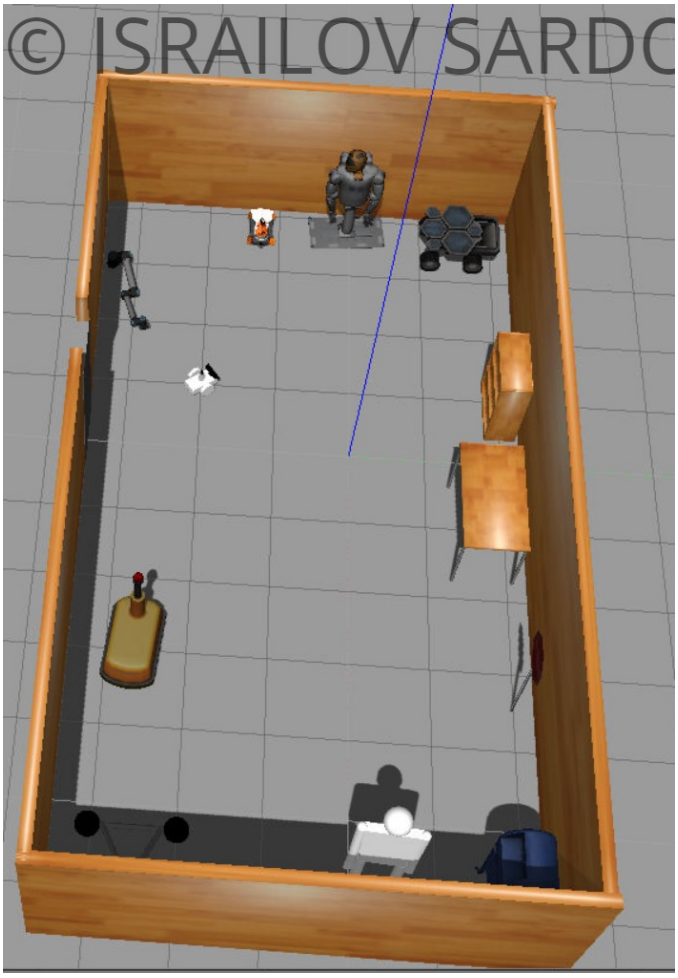

Fig. 3. robocop world in gazebo

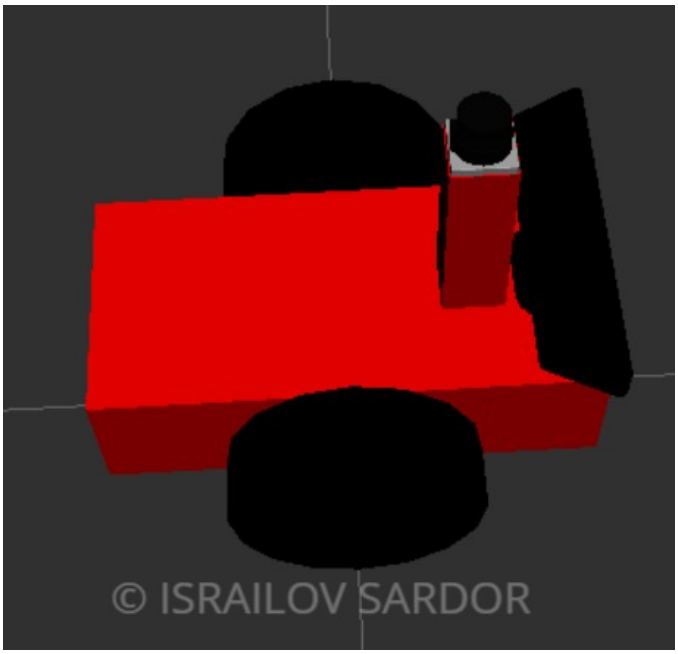

Fig. 4. robot configuration

parameters were tuned to make the robot movement more fluid. The robot was moved with the help of keyboard and teleop_keyboard node that was originally part of the turtlebot simulator.

3.3 Architecture of the package

Overall, the project included 4 launch files:

- **world.launch** to launch the gazebo with the robot configuration and a particular environment.
- **mapping.launch** was the the nodes used to map the environment and visualise the mapping.
- **teleop.launch** was the launch file in order to manipulate the robot and to publish on teleop_cmd_vel that was subscribed by gazebo afterwards.
- **rviz.launch** used to launch the rviz and observe the mapped data as well as topics, maps etc.
- **localisation.launch** used to localise the robot when the environment is already mapped. for this used it is necessary to implement it in the same session with the saved database.

Tools for debugging(most often used)

- **rqt_graph** to launch the gazebo with the robot configuration and a particular environment.
- **rosrun tf view_frames** the nodes that creates the .pdf file that shows all the frames of the configuration.
- **roswtf** is the tool to see if there are some errors or some subscriptions unconnected.
- **rqt_image_view** the tool used to visualise the images of rgb, depth etc.
- **rqt_console** is the tool to see in more details the warning and errors and to see the suggestions to correct them.
- **rostopic/rosnode list** is the tool to see the list of the running topics, nodes.

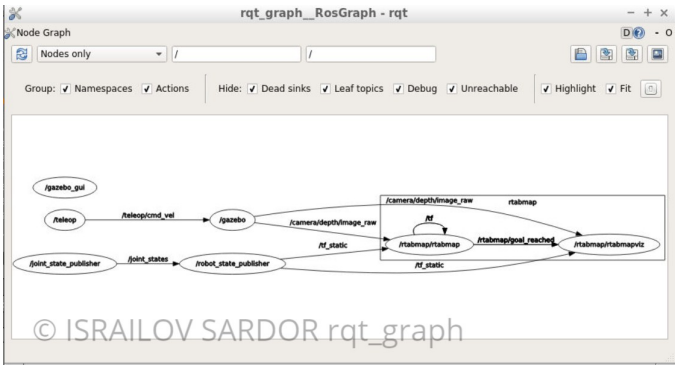

Fig. 5. rqt graph

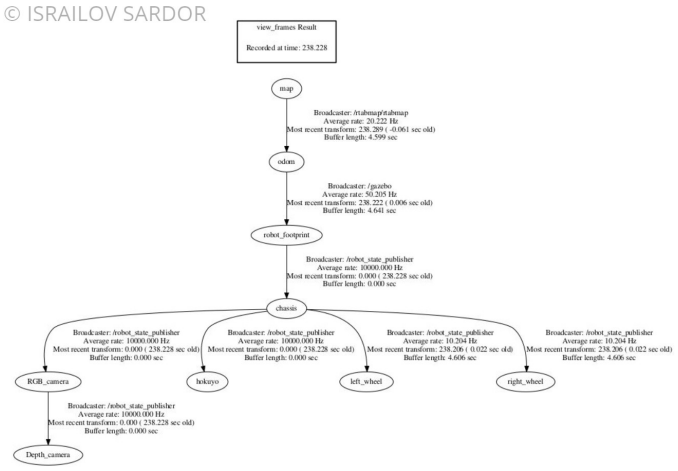

Fig. 6. frame connectivity observation with TF package

- **rostopic info topic_name** is the tool to know the publishers/subscribers of a particular topic
- **rostopic echo topic_name** was the tools to use to observe the messages send via the topic. It was used for teleop node as well to understand the warning of the type(scan is empty etc.)

4 Results

Two environments were mapped and visualised: You can

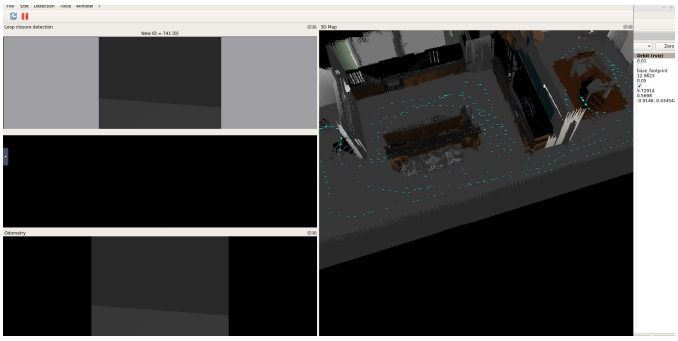

Fig. 7. turtle-bot burger

as well see the black lines in rviz that correspond to the occupancy grid mapping.

Fig. 8. robocop.world mapped

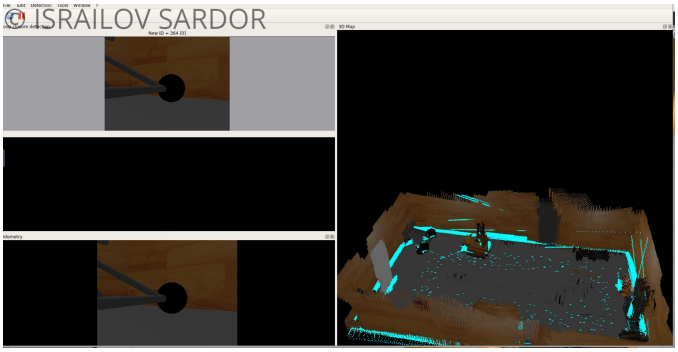

Fig. 9. visualisation of features with rtabmapviz

5 Discussion and Conclusion

In this project the robot was moved with the keyboard to map the environment with one of the most advanced mapping packages in ROS (rtab-map). Several strategies were used and parameters were tuned: **SURF- FAST/Brief** Before implementation in real life, it is necessary to adjust the main parameters as minimum visual inliers, maximum visual words per image(bag-of-words).

In general, the richer the scene in details, the more precise the map we can obtain.

6 Future Work

It is possible to generate the /scan topic without using hokuyo sensor and converting the depth topic published by the kinect camera with the usage of the following package **depthimage_to_laserscan** that would make the robot cheaper as it would possibly eliminate the LiDAR. But this solution is not always reliable in many applications because of noise and precision.

The rtabmap can be combined with amcl and other ros packages to autonomously navigate and explore the environment, creating maps for buildings, cities etc.

7 REFERENCES

1) Retrieved september 12, 2019, from http://wiki.ros.org
2) Retrieved september 12, 2019, from https://udacity.com