

Project “Map My World”

Dmitry Gavrilenko

Abstract – This article describes the set up procedure of Real-Time Appearance-Based Mapping (RTAB MAP) algorithm in Robotic Operating System (ROS) environment. The algorithm reads RGB-D camera, odometry and LIDAR sensor data. It simultaneously generates a map of a static environment with a robot motion trajectory inside it. RTAB MAP calculates both 2D and 3D maps. The algorithm detects loop closures in the robot trajectory in order to improve mapping accuracy.

Index – SLAM, RTAB MAP, ROS, RGB-D, LIDAR, mapping

1 Introduction

In order to do long-term path planning in a static environment, the mobile robot needs to know its map. If the map is unknown or outdated, the robot can generate or update it with the help of a Simultaneous Localization and Mapping (SLAM) method. SLAM fuses robot sensor data and odometry in order to build the map and track the robot's location with respect to it.

2 Background

SLAM is an inherent software module of the robot, which task is to explore some unknown environment. SLAM methods vary by the map formats they generate, what sensors they use, their resistance to noise and outliers in sensor input and data associations. Usually, SLAM methods are computationally intensive. Therefore, real-time online SLAM, which updates the map incrementally at the rate of the robot motion, is the most challenging problem.

Two most popular map formats are occupancy grids and GraphSLAM.

In the occupancy grid, a map consists of cells. Each cell holds a probability or log likelihood value of whether it is passable or not. The map size grows exponentially with the number of dimensions it represents. However, there are some optimization structures to keep the size small, such as oct-trees.

In GraphSLAM, the map consists of nodes, holding feature attributes with their relative coordinates; and connections, holding transformations between the nodes. Each node represents some robot spatial location. If a robot during its motion encounters the location it visited before, GraphSLAM algorithm adds a loop to the graph and activates the loop closure routine, which performs a global error minimization of relative transformations, connecting the looped nodes. The global error minimization procedure distributes the error noise residuals across the loop in order to set their sum of squares minimum.

[3] gives an introduction to SLAM based on Extended Kalman Filter (EKF).

Real-Time Appearance-Based Mapping (RTAB MAP, see [1] and [2]) library implements a SLAM algorithm based on RGB-D camera and LIDAR data. It provides a Robotic Operating System (ROS) package, which simplifies its integration with simulated and real robots and their sensors. RTAB MAP generates an occupancy grid 2D Map and a GraphSLAM 3D point-cloud map of the environment.

This paper describes the set up procedure of RTAB MAP library for the robot, simulated in a Gazebo environment under Robotic Operating System.

3 Scene and Robot Configuration

The simulated robot, building the map, is shown in Fig 1. It is equipped with the LIDAR sensor and a forward-facing RGB-D camera.

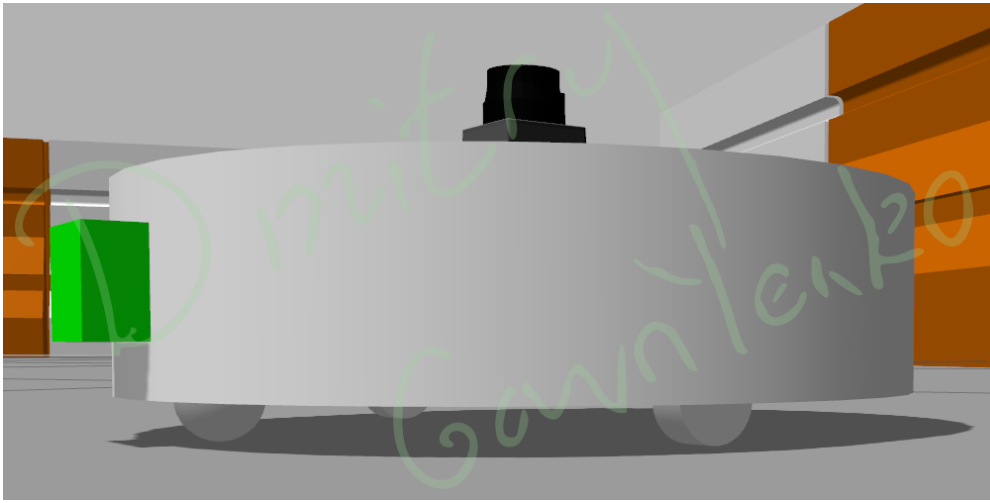


Fig. 1. Simulated Robot

The transformation tree of the robot is shown in Fig 2. It consists of the following nodes:

- **robot_footprint** is the root of the transformation tree. Its position is tracked by RTAB MAP package as it builds the map
- **chassis** is the cylindrical body of the robot, which all the other nodes are attached to
- **left_wheel** and **right_wheel** are attached to the differential drive controller that actuates the robot motions and rotations
- **caster** is the forward wheel of the robot that supports the robot stability and allows it to move in any direction
- **hokuyo** is the LIDAR of the robot, which is installed in its top to provide 360 degree view
- **camera_depth_frame** and **camera_optical_link** are two nodes that represent the same RGB-D camera. In the simulated environment, Gazebo and RTAB_MAP cameras have different reference frame orientations. To overcome this incompatibility issue, two nodes for two different reference frames were introduced.

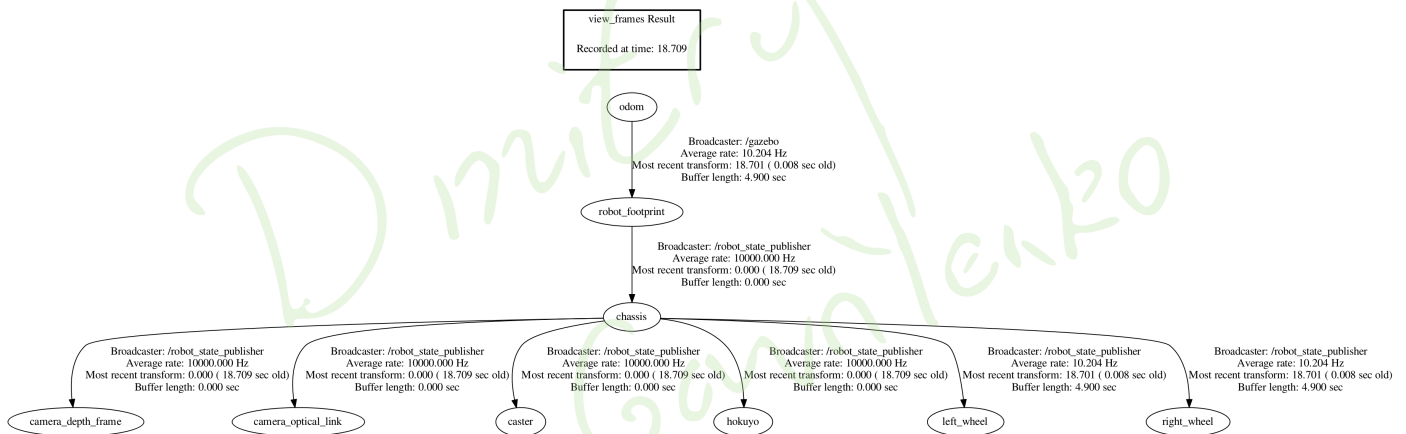


Fig. 2. Transformation Tree of Simulated Robot

slam_project package, simulating RTAB MAP SLAM in Gazebo environment, consists of the following folders:

- **launch/** contains ROS launch scripts for Gazebo, mapping, localization, robot control and rviz, as requested in the project assignment
- **meshes/** contains a mesh file for the LIDAR sensor
- **scripts/** contains a Python script, controlling the robot motion via a keyboard, similar to turtlesim teleop
- **urdf/** contains files, describing the simulated robot structure
- **worlds/** contains two Gazebo worlds: kitchen_dining.world, provided by the assignment, and my.world, designed from inception within the scope of this project

RTAB MAP parameter values and references to the robot published topics are defined in **launch/mapping.launch** file, which generates rtabmap.db in `~/.`ros folder. Localization script **launch/localization.launch** has a similar structure and settings, but does not remove or update existing rtabmap.db file.

The following settings were updated in **launch/mapping.launch** to enable correct RTAB MAP work:

1. `rgb_topic` subscribes to the colored camera image topic, published by `urdf/my_bot.gazebo` robot
2. `depth_topic` subscribes to the depth layer of RGB-D camera, published by `urdf/my_bot.gazebo`
3. `camera_info` subscribes to the camera intrinsic parameters, published by `urdf/my_bot.gazebo` robot
4. `rtab_map` package reads `/scan` topic, published by `urdf/my_bot.gazebo` hokuyo LIDAR sensor

The other parameters were left with their default values:

- Use visual bag of words to perform the loop closure (Kp/Strategy=0) instead of Iterative Closest Point procedure
- Use SURF descriptor for visual features (Kp/DetectorStrategy=0)
- Do not update the map if the robot is not moving (Mom/NotLinkedNodesKept=false)
- Update nodes once per second if the robot moves (Rtabmap/DetectionRate=1)

world/my.world consists of the default Gazebo objects, representing the outer environment (a few buildings, the car and some other stuff). Since rtab_map uses computer vision methods to detect loop closures, each Gazebo mesh is used only once in the map. This prevents false loop closures. Another alternative to overcome this problem would be to increase the number of inliers (Vis/MinInliers) or use ICP method to close the graph loops.

4 Results

Fig 3 and 4 illustrate 3D mapping results of **kitchen_dining.world** with three loop closures. 2D map is visualized in rviz view in Fig 4.

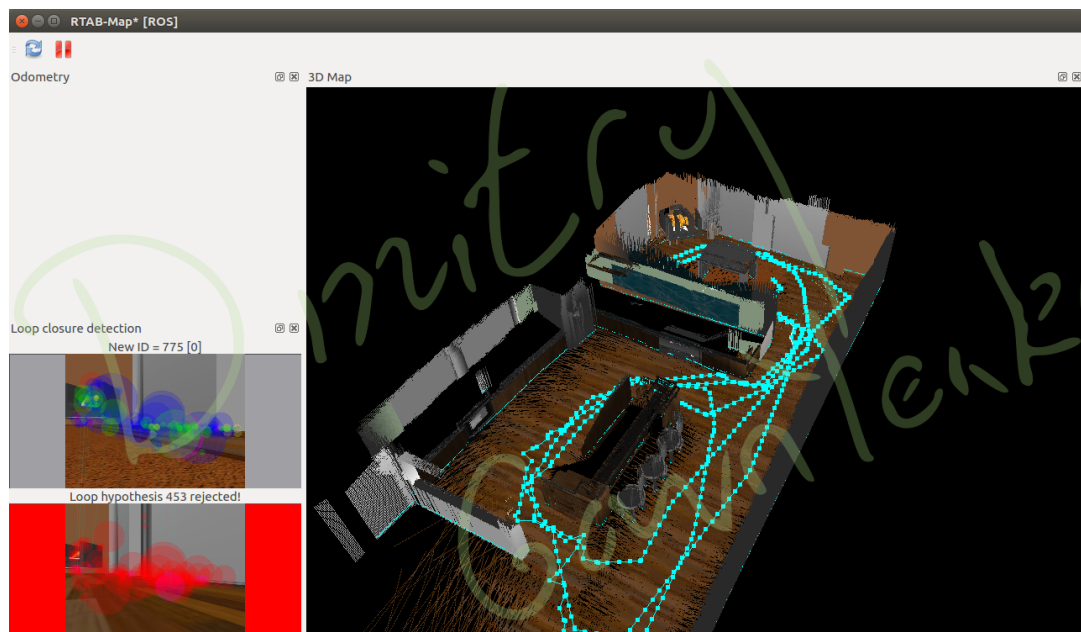


Fig. 3. RTAB MAP mapping results of kitchen_dining.world

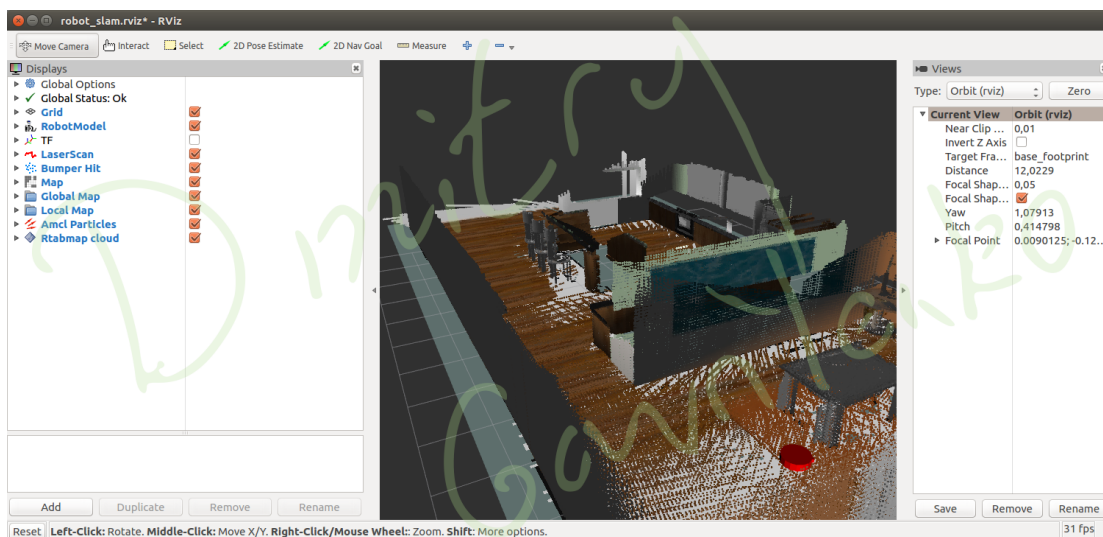


Fig. 4. Mapping results of kitchen_dining.world visualized in rviz

Fig 5 and 6 illustrate 3D mapping results of **my.world**. 2D map is also visualized in rviz view in Fig 6.

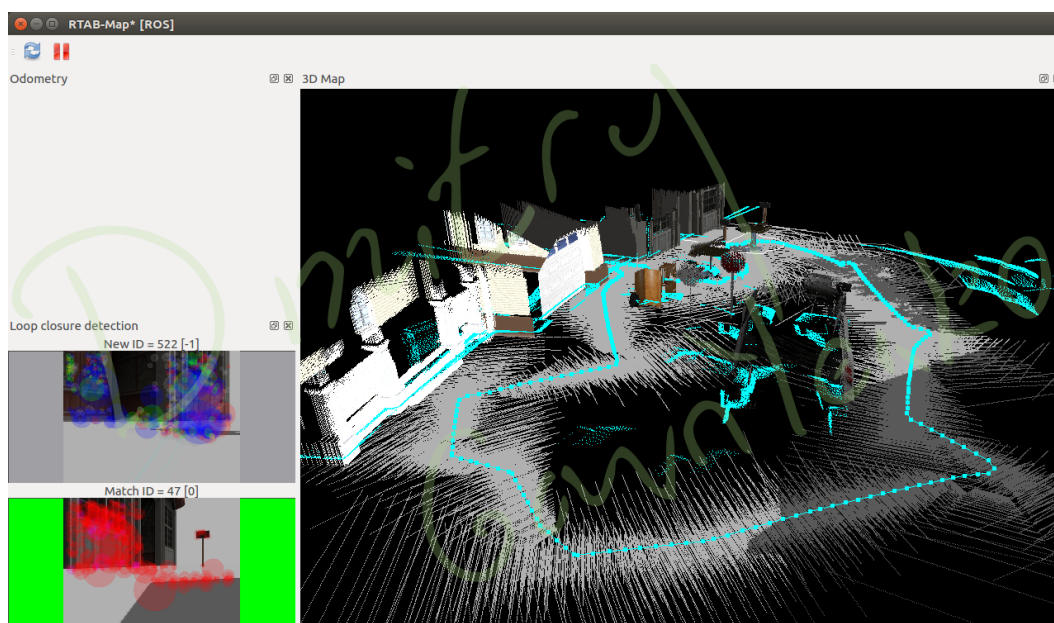


Fig. 5. RTAB MAP mapping results of my.world

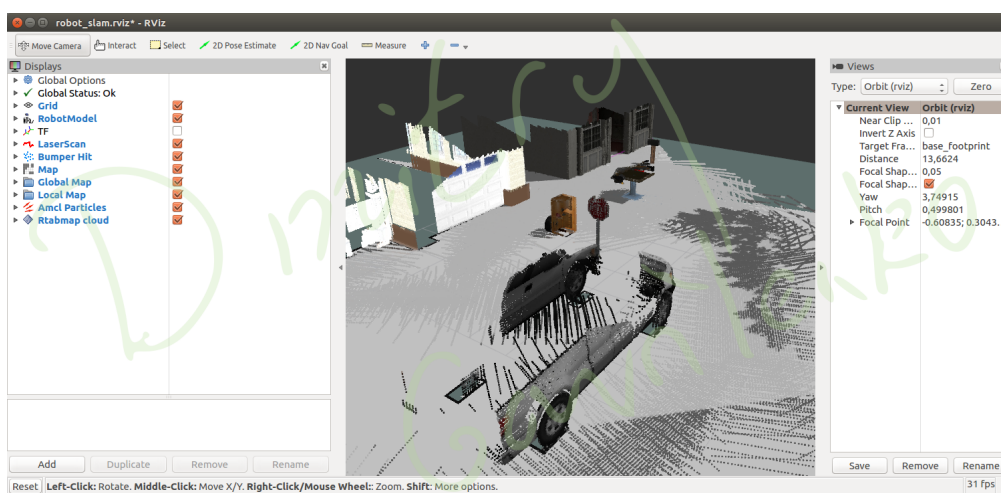


Fig. 6. RTAB MAP mapping results of my.world in rviz

5 Discussion

slam_project package has successfully mapped kitchen_dining.world and my.world environments. The quality of kitchen_dining.world mapping is higher due to the following reasons:

- More loop closures. The robot path of kitchen_dining.world consists of three loop closures, while the robot path of my.world consists of only one. The more loop closures, the more accurate the map is.
- Smaller area with richer visual features. The environment in kitchen_dining.world consists of a lot of distinctive visual features for SURF algorithm to detect and track. While my.world represents an open space, which has lower level of detail, where RTAB MAP has less number of inliers to track.

6 Future Work

The robot my_bot, assembled in slam_project, represents a home appliance vacuum cleaner. Mapping algorithm allows the robot to plan the cleaning procedure more efficiently by memorizing the areas in the apartment that have already been cleaned.

3D map produced by RTAB_MAP algorithm could also be useful in a surveillance or reckoning quadcopter path planner, as it follows the 3D path.

References

- [1] <http://introlab.github.io/rtabmap/>
- [2] M. Labbé and F. Michaud, *RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation*, 2018
- [3] Soren Riisgaard and Morten Rufus Blas, *SLAM for Dummies*