Map My World

Vladimir Kobal

Abstract—Robots are used in an unknown environment frequently. Such a case requires robots to orient themselves somehow in that environment. That's where SLAM algorithms are useful. The goal of this project is to implement a working self-oriented robot model in a simulated environment, and it was achieved successfully by means of ROS and RTAB-Map package.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, SLAM.

1 Introduction

It is not always possible to provide a map for a robot. Moreover, most of the possible scenarios for robot usage connected with an unknown or dynamic environment. It is a case when simultaneous localization and mapping is extremely useful. The goal of this project is to practice using SLAM in a simulated environment by the example of RTAB-Map ROS package.

2 Background

There are a lot of SLAM algorithms. We are concentrating on two of them.

FastSLAM uses the Rao-Blackwellized particle filter (MCL for localisation + Low-Dimentional EKF for mapping). There is a Grid-based version of FastSLAM which uses Occupancy Grid Mapping instead of EKF, therefore it is not dependent on landmarks. FastSLAM can generate 2D maps only.

GraphSLAM is capable of generating 3D maps along with 2D ones, it is more accurate and use less computational resources than FastSLAM. The basic idea of the GraphSLAM is to build a graph composed of poses, features and constraints and optimize it over the time. In the project RTAB-Map (an implementation of the GraphSLAM) will be used.

While it might be enough to map the environment in 2D for some cases, 3D mapping allows avoiding complex collisions and using spacious information for motion and path planning. It is especially important for flying robots or mobile robots with manipulators whose motions are not constrained to a 2D plain.

3 Scene and robot configuration

For the customized world a simple office environment was chosen. Building Editor was used to create walls and place windows and doors. A chair model was created using Model Editor. Then the office building was furnished with tables, chairs, bookshelves and other stuff (Fig. 1).

Fig. 1. Office model

For the robot the model from the previous project was used (Fig. 2, 3). Chassis is based on a flat disk. Rear axle holds 2 driven wheels with torque set to 20 Nm. Coordinates of the chassis are shifted to keep the axle at x=0. Front caster friction was decreased down to 0.0. Total mass of the robot is 35.2 kg, dimensions — 0.55 x 0.45 x 0.27 m. RGBD camera and laser sensor are stacked up on the front end of the chassis.

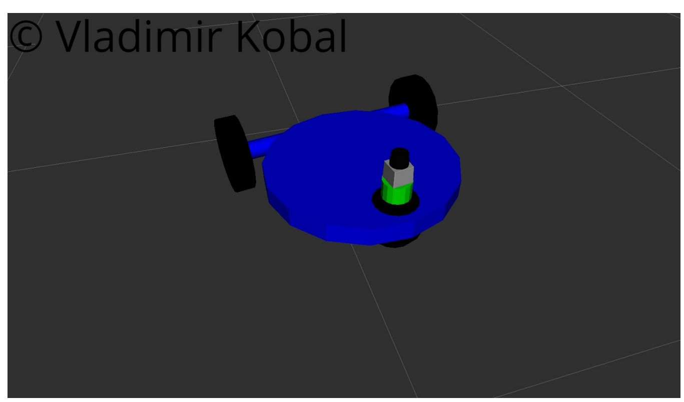

Fig. 2. Robot model

For building the package the structure from the previous project was used. In the robot model `.xacro` file RGB camera was changed to RGBD one. `camera_optical_frame` was added and rotated to conform with RTAB-Map pack-

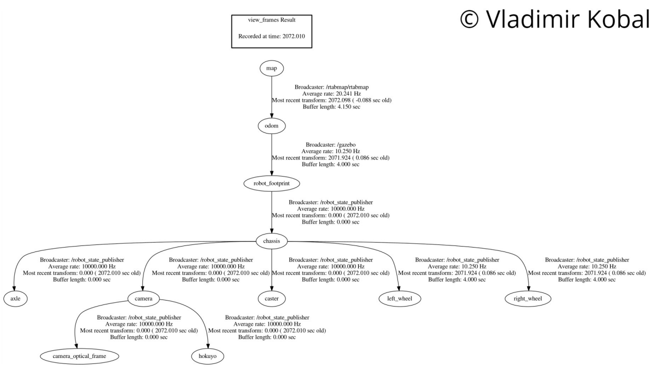

Fig. 3. TF tree

age requirements. Launch files for RTAB-Map package and `teleop` script were added and topics were matched.

4 Results

The model successfully creates a map of an environment and positions itself in the map. The occupancy grid and 3D maps obtained for the Kitchen and Dining world are shown in Fig. 4 and 5. Corresponding maps for the Office world are shown in Fig. 6 and 7.

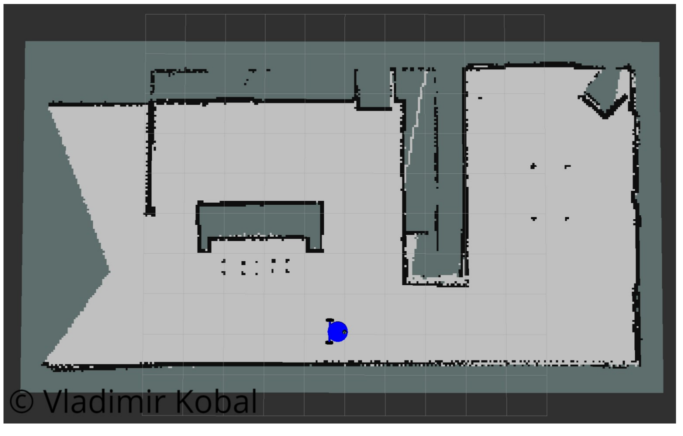

Fig. 4. Kitchen and Dining 2D map

Fig. 5. Kitchen and Dining 3D map

Fig. 6. Office 2D map

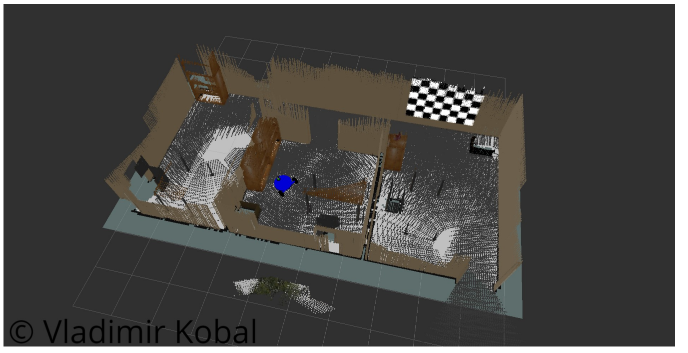

Fig. 7. Office 3D map

5 Discussion

The robot made 3 passes across the provided map (Fig. 8). 148 global loop closures where detected. However, after the first pass the maps were already quite good. For the Office world only one pass was made but the obtained maps are acceptable. So the algorithm was good for both scenarios even without a scrupulous parameter tuning.

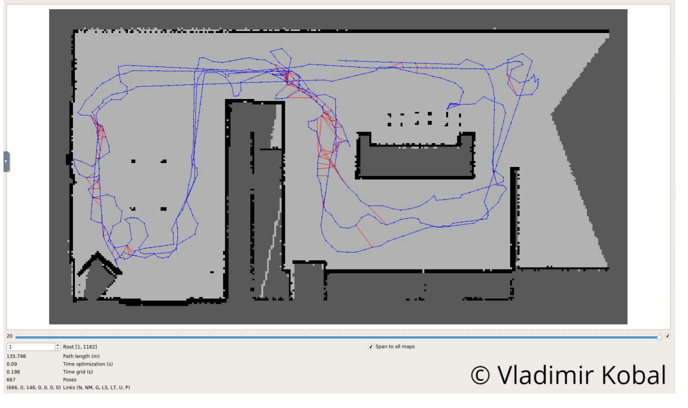

Fig. 8. Graph with global loop closures

6 Future work

RTAB-Map has a huge number of parameters. If it is needed to compose more accurate maps or use less resources it is possible to achieve by tuning the parameters.

It is also worth trying out the algorithm in different types of environment to study its strong and weak sides, because usefulness of SLAM can not be underestimated. It is very helpful, and even essential in overwhelming majority of mobile robot applications. A list of just a small number of them: rescue missions, exploration, self-driving, mining, housekeeping, agriculture, warehouse transportation, pipe system maintaining, etc. There is an infinity number of tasks where mobile robots can be applied, and at the same time they feature dynamic or unknown environment where the robot needs to orient himself at first to be able to complete the task.