Map My World Project

Reinaldo Ossuna

Abstract—In this project we implemented the Simultaneous Localizion and mapping (SLAM) with the Real-Time Appearance-Based Mapping (RTAB-Map) to create a 2D occupacy grid and a 3D map of two enviroments created with Gazebo. The used robot is equipped with a RGB-D camera and a Ranger sensor. The result of the 2D and 3D maps in both enviroments was excellent.

Index Terms—Robot, Udacity, SLAM, EKFSLAM, GraphSLAM, RTAB-MAP

1 Introduction

SLAM is the process by which the system performace the simultaneous estimation of the state of the robot equiped with on-board sensors and the construction of the map. This can be called the egg-chicken problem because normaly the robot need the map to performace the localization problem and the pose is needed to performace the mapping.

A solution to the SLAM problem has been seen as a 'Holly grail' for the mobile robotic community as it would provide the means to make a robot truly autonomous.**??**

This project uses RTAB-MAP packages from ROS, wich it's a type of GraphSLAM algorithm.

2 Background

Almong all methods proposed in the literature to solve the SLAM algoriths, we could point out some facts.

2.1 Occupancy grid

Occupancy grid maps address the problem of generating consistent maps from noisy and uncertain measurement data, under the assumption that the robot pose is known.

The standard of the algorithm is to calculate the posterior over maps given $z_{1:t}$ the set of measurements up to time t and the $x_{1:t}$ path of the robot.

$$p(m|z_{1:t}, x_{1:t}) \tag{1}$$

The occupancy mapping algorithm estimates for each grid cell individually the posterior probability of occupancy. It is an adaptation of the binary Bayes filter for static environments.

The algorithm loops through all grid cells i, and updates those that fall into the sensor conde of the measuremets zt. For those where it does, it updates the occupancy value by the function inverse_sensor_model().

Data from multiple sensors can be fused into a single map. By maintaing multiple maps, one for each sensor and extracting the most pessimistic occupancy value when making navigation decisions. This procedure is preferable when differents sensors are sensitive to differents types of obstacles.

2.2 online SLAM

Estimate the posterior over the momentary pose along with the map:

$$p(x_t, m|z_{1:t}, u_{1:t}) \tag{2}$$

We can notice that this algorith involves estimating the *actual pose* and the map.

2.3 Full SLAM

Estimate the posterior over the *entire path* $x_{1:t}$ along with the map.

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) \tag{3}$$

2.4 EKF SLAM

The Extended Kalman Filter (EKF) can be viewed as a variant of a Bayesian Filter. It's provide a recursive estimate of the state of dynamic system.

The EKF uses a prediction and an update step. The predictions step calculates the probability of the current state x_t, while the measurement z_t for the current time step k is not yet available:

$$\text{p(x}_t|z_{t-1}) = \int p(x_t|x_{t-1})p(x_{t_1}|z_{t-1})dx_{t1}$$
(4)

2.5 Fast SLAM

In FastSLAM, alike in EKF SLAM, poses are assumed to behave according to a probabilist law name motion model with an underlying density. FastSLAM decomposes the SLAM problem into one robot localization problem, and a collection of K landmark estimation problems.

2.6 FastSLAM 1.0 x FastSLAM 2.0

Both algorith use the a low dimensional Extended Kalman Filter to estimate the posterior over the map features.

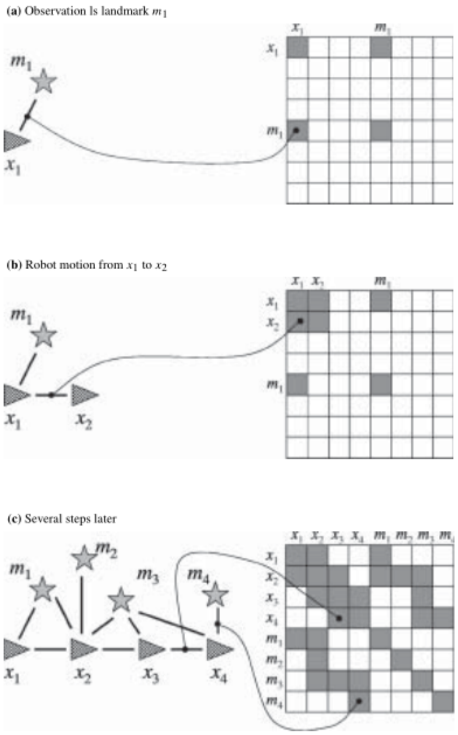
Basically, the only modification in the version 2 is that the proposal distribution should not only rely on the precious estimate of the pose s_{t-1} but also on the actual measurements z_t.**??**

2.7 Grid-based FastSLAM

Grid-based FastSLAM can model the environment using grid maps with out predefining any landmark position. Therefore, we can now solve the SLAM problem in an arbitray environment.

2.8 GraphSLAM

A Graph-based SLAM approach construct a simplified estimation problem by abstracting the raw sensor measurements. These raw measurements are replaced by the edges in the graph which can be seen as 'virtual measurements'. We can apply several algorithm to reduce the dimensionality of the optimization problem. [1]


(a) Observation ls landmark m_1

(b) Robot motion from x_1 to x_2

(c) Several steps later

Fig. 1. Illustration of the acquisition of the information matrix in GraphSLAM. The left diagram shows the dependence graph [1]

2.9 RTAB-MAP

Real-Time Appearance-Based Mapping is a RGB-D, Stereo and Lidar Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. The loop closure detector uses a bag-of-words approach to determinate how likely a new image comes from a previous location or a new location. When a loop closure hypothesis is accepted, a new constraint is added to the map's graph, then a graph optimizer minimizes the errors in the map. A memory management approach is used to limit the number of locations used for loop closure detection and graph optimization, so that real-time constraints on large-scale environnements are always respected. RTAB-Map can be used alone with a handheld Kinect, a stereo camera or a 3D lidar for 6DoF mapping, or on a robot equipped with a laser rangefinder for 3DoF mapping.

3 Scene robot configuration

3.1 Robot Configuration

The robot model was based in the robot from the previous project. The RGB camera was replaced with a RGB-D camera. The LaserScan could be removed since we can create a scan topic using the depth point cloud from the RGB-D camera. See Fig2

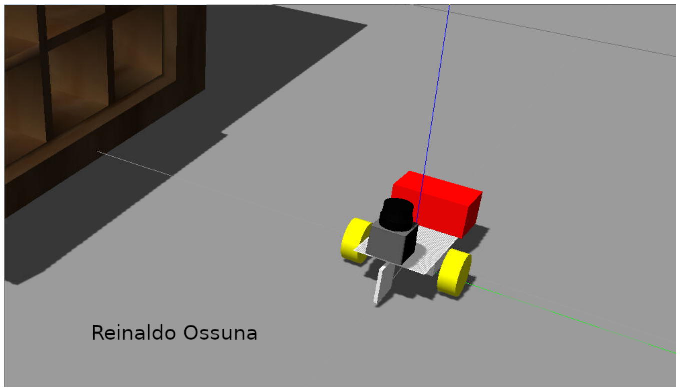


Fig. 2. Robot with RGB-D Camera

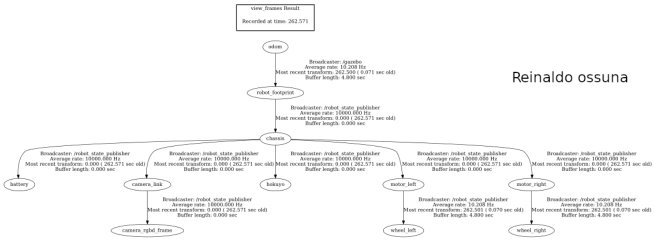

view_frames Result
Recorded at time: 262.571
odom
Broadcaster: /gazebo
Average rate: 10.208 Hz
Most recent transform: 262.500 (0.071 sec old)
Buffer length: 4.800 sec
robot_footprint
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
chassis
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
battery
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
camera_link
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
hokuyo
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
motor_left
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
motor_right
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (262.571 sec old)
Buffer length: 0.000 sec
camera_rgbd_frame
Broadcaster: /robot_state_publisher
Average rate: 10.208 Hz
Most recent transform: 262.501 (0.070 sec old)
Buffer length: 4.800 sec
wheel_left
Broadcaster: /robot_state_publisher
Average rate: 10.208 Hz
Most recent transform: 262.501 (0.070 sec old)
Buffer length: 4.800 sec
wheel_right
Reinaldo ossuna

Fig. 3. Transform tree

3.2 Scene Configuration

A new scene was builted using the Gazebo. We placed diferents objects in the scene to facilitate the loops closures. See Fig4.

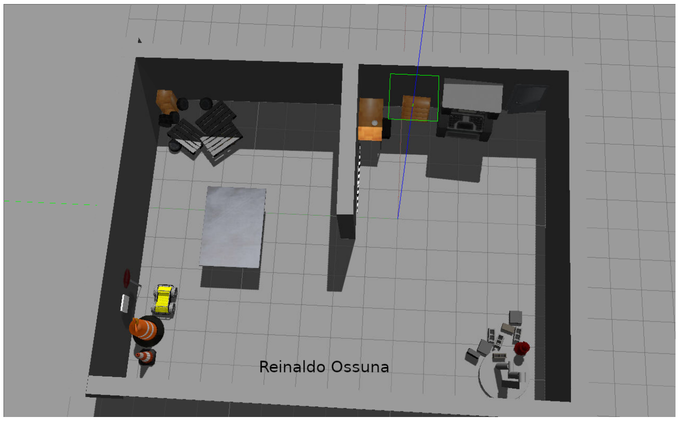


Fig. 4. Room world

4 Results

After navigating around the worlds, the 3D and 2D occupacy grid map was generated for each environment.

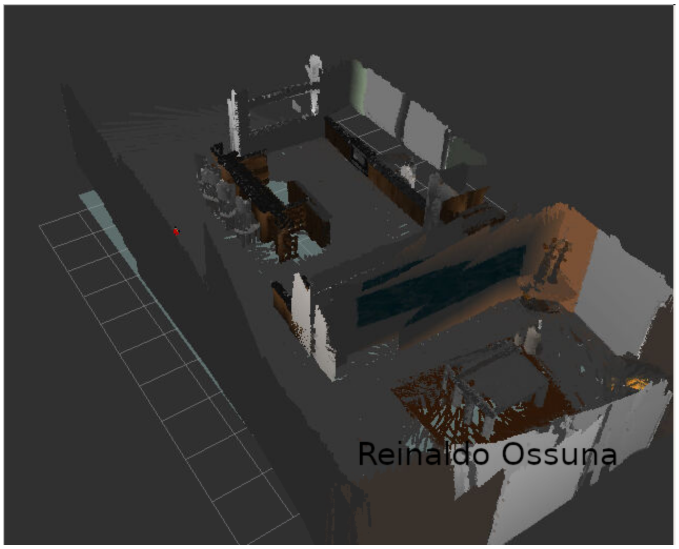
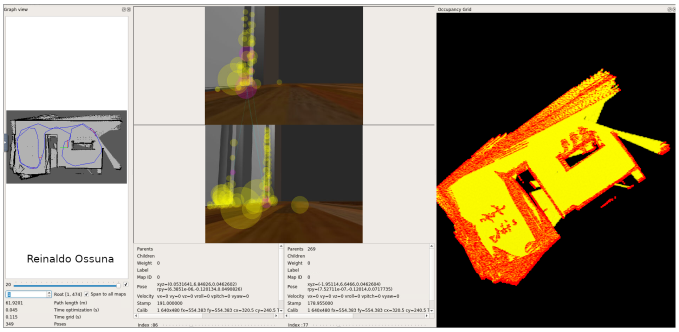


Fig. 5. 3D Map from the Kitchen

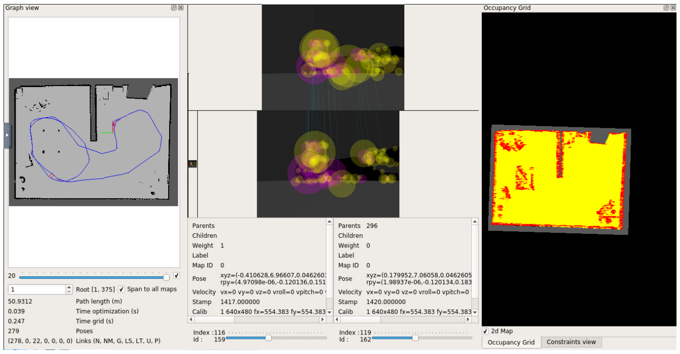


```
20
1            Root [1, 474]  ✓ Span to all maps
61.9201      Path length (m)
0.045        Time optimization (s)
0.115        Time grid (s)
349          Poses

Parents
Children
Weight    0
Label
Map ID    0
Pose      xyz=(0.0531641,6.84826,0.0462602)
          rpy=(6.3851e-06,-0.120134,0.0490826)
Velocity  vx=0 vy=0 vz=0 vroll=0 vpitch=0 vyaw=0
Stamp     191.000000
Calib     1 640x480 fx=554.383 fy=554.383 cx=320.5 cy=240.5 T
Index :86

Parents   269
Children
Weight    0
Label
Map ID    0
Pose      xyz=(-1.95114,6.6466,0.0462604)
          rpy=(7.52711e-07,-0.12014,0.0717735)
Velocity  vx=0 vy=0 vz=0 vroll=0 vpitch=0 vyaw=0
Stamp     178.955000
Calib     1 640x480 fx=554.383 fy=554.383 cx=320.5 cy=240.5 T
Index :77
```

Fig. 6. Mapping from the Kitchen



```
20
1                         Root [1, 375]  ✓ Span to all maps
50.9312                   Path length (m)
0.039                     Time optimization (s)
0.247                     Time grid (s)
279                       Poses
(278, 0, 22, 0, 0, 0, 0)  Links (N, NM, G, LS, LT, U, P)

Parents
Children
Weight    1
Label
Map ID    0
Pose      xyz=(-0.410628,6.96607,0.046260
          rpy=(4.97098e-06,-0.120136,0.151
Velocity  vx=0 vy=0 vz=0 vroll=0 vpitch=0
Stamp     1417.000000
Calib     1 640x480 fx=554.383 fy=554.383
Index :116
Id :   159

Parents   296
Children
Weight    0
Label
Map ID    0
Pose      xyz=(0.179952,7.06058,0.0462605
          rpy=(1.98937e-06,-0.120134,0.183
Velocity  vx=0 vy=0 vz=0 vroll=0 vpitch=0
Stamp     1420.000000
Calib     1 640x480 fx=554.383 fy=554.383
Index :119
Id :   162

✓ 2d Map
```

Fig. 7. Mapping from the Room

5 Discussion

In the provided map there's many features, thereby a good 2d occupacy grid could be generated with two passes in the world.

In the first try in the created map the robot didn't find many features and mapping was not working. We add more object throught the scene to achieve a sucessful mapping.

In the created scene, there's a Armbot, which it's keep swinging, the SLAM was not able to map this part.

6 Future work

In the future, deploy the model in a real world and perfome similar task performed here. Try different algorithms to the SLAM problem. Like Hector SLAM, ORB-SLAM, something similar to this article [2].

References

[1] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[2] Vehicular Technology Society and Institute of Electrical and Electronics Engineers, "2017 14th Workshop on Positioning, Navigation and Communications (WPNC) : 25-26 Oct. 2017.," 2017.

[3] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[4] T. Zhang, Z. J. Chong, B. Qin, J. G. Fu, S. Pendleton, and M. H. Ang, "Sensor fusion for localization, mapping and navigation in an indoor environment," *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2014 - 7th HNICEM 2014 Joint with 6th International Symposium on Computational Intelligence and Intelligent In*, no. November, pp. 1–6, 2014.

[5] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph SLAM," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1254–1261, 2012.

[6] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[7] S. Thrun, W. Burgard, and D. Fox, "Rob550: Probabilistic Robotics," pp. 1999–2000, 1999.

[8] Z. Kurt-Yavuz and S. Yavuz, "A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms," *INES 2012 - IEEE 16th International Conference on Intelligent Engineering Systems, Proceedings*, pp. 37–43, 2012.

[9] V. Celan, I. Stancic, and J. Music, "Cleaning up smart cities-Localization of semi-autonomous floor scrubber," *2016 International Multidisciplinary Conference on Computer and Energy Science, SpliTech 2016*, 2016.

[10] M. Calonder, "EKF SLAM vs. FastSLAM {A Comparison}," *Cvlab-Report-2010-001*, pp. 1–5, 2006.

[11] T. Bailey and H. F. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[12] W. Burgard, C. Stachniss, K. Arras, and M. Bennewitz, "Introduction to Mobile Robotics SLAM : Simultaneous Localization and Mapping What is SLAM ?," no. June, 2010.

[13] J.-A. Fernandez-Madrigal and J. L. Blanco Claraco, "Simultaneous localization and mapping for mobile robots : introduction and methods," p. 483, 2013.

[14] J. J. Cadena, C and Carlone, L and Carrillo, H and Latif, Y and Scaramuzza, D and Neira, J and Reid, I and Leonard, J J and Leonard}, C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[15] S. Hong, "A Comparison of SLAM Algorithms Based on a Graph of Re- lations," p. 2012, 2012.

[16] K. Berns and E. von Puttkamer, "Simultaneous localization and mapping (SLAM)," *Autonomous Land Vehicles*, no. September, pp. 146–172, 2010.