

Robotics Software Engineer Nanodegree: Slam Project

Manuel Huertas López

Abstract—Simultaneous localization and mapping, SLAM, is the problem of building a map of an unknown environment while simultaneously localize a robot relative to this map. In this project a 2D and 3D occupancy grid map was created from both a simulated environment provided by the Udacity team and an environment that was created from scratch. A robot with two differential drives and equipped with a Lidar and a RGB-D camera was used in the simulated environment. The library RTAB-Map was selected to solve the slam problem because: it offers a good performance and memory management, a good portfolio of development tools, and a good quality of the documentation.

Index Terms—slam, udacity, rtab-map.

1 INTRODUCTION

THERE are some applications where a robot is provided with a map of its environment and it is able to estimate its pose based on this map, the odometry data and some sensor data like a Lidar or RGB-D camera. Furthermore, in some cases the map is unknown, either because the area is unexplored or because the surrounding changes often and the map may not be updated. In that case the robot must construct a map, the robot's pose is known and the robot is equipped with sensor data to measure its environment. Finally, in the most general case neither the robot's pose nor map is available, this is where SLAM comes in. In SLAM, the robot will use the odometry and the sensor data to build a map of its environment while simultaneously localizing itself relative to this map. The SLAM problem is quite challenging, with noise in the robot pose and measurement, the robot's pose will be uncertain and the construction of the map will be uncertain as well, they are interrelated. This project will use an implementation of SLAM called real-time appearance-based mapping or RTAB-Map. This implementation is based on GraphSLAM. GraphSLAM is a SLAM algorithm that solves the full SLAM problem, this means that the algorithm recovers the entire path and map, instead of just the most recent pose and map.

2 BACKGROUND / FORMULATION

The most fundamental problem in robotics. It is considered as a prerequisite for truly autonomous robot.

2.1 Grid-based FastSLAM

FastSLAM algorithm uses the particle filter approach along with the low-dimensional extended Kalman filter to incrementally estimate the posterior distribution over robot pose along with the positions of some landmarks. FastSLAM decomposes the SLAM problem into a robot localization problem, and a collection of landmark estimation problems that are conditioned on the robot pose estimate. In order to solve the SLAM problem FastSLAM uses a modified particle

filter for estimating the posterior over robot path, for every particle in the filter there are K Kalman filters that estimate the K landmark locations conditioned on the path estimate. An implementation of this idea requires $O(MK)$ time, where M is the number of particles and K the number of landmarks, by using a tree-based data structure the running time can be reduced to $O(M \log K)$. Finally, there are different versions of the algorithm. The one used during the course is called Grid-based FastSLAM which adapts FastSLAM to grid maps with the main advantage of no need for predefining landmarks, that means it will work in arbitrary environments.

$$p(x_{0:t}, m | z_{1:t}, u_{1:t}) = p(x_{0:t} | z_{1:t}, y_{1:t}) p(m | x_{1:t}, z_{1:t}) \quad (1)$$

2.2 GraphSLAM

GraphSLAM uses constraints to represent relationships between robot poses and the environment, and then tries to resolve all of these constraints to create the most likely map given the data.

3 SCENE AND ROBOT CONFIGURATION

4 RESULTS

5 DISCUSSION

6 CONCLUSION / FUTURE WORK

REFERENCES

- [1] <https://eu.udacity.com> Robotics Software Engineer Nanodegree program.
- [2] Adit Deshpande *The 9 Deep Learning Papers You Need To Know About*.
- [3] Alfredo Canziani, Eugenio Culurciello, Adam Paszke *An analysis of deep neural network models for practical applications*.