Multi-Robot SLAM

Esraa Magdy

Abstract—Multi-Robot interaction is an important problem in the field of service robots or emergency assistance where multiple robots need to share specific information among them, the shared information could vary from sharing a map for the environment to sharing information regarding an object inside the environment. This project will be concerned with the case where multiple robots are navigating through an environment, building local maps, each robot from its point of view, and these local maps needs to merged together to get a global map of the environment.

Index Terms—Robot, IEEEtran, LaTeX, SLAM.

1 BACKROUND

MAPPING large areas require more than one robot to span the whole place. The problem of multiple robots cooperating together to build a shared map is defined as Cooperative SLAM or Multi-Robot SLAM. The Cooperative SLAM problem has added more complexity to the usual SLAM problem. As, in this case not only that each robot will perform SLAM on its own. But the robots need to communicate with each others to get the joint map. This problem could be tackled using multiple approaches: One approach is to represent each local map as a metric map and represents the global map as a topological graph with each node being one local map.The advantage of using a topological map is that it is computationally less expensive than the metric map. Another problem is the data sharing, which amount of the data should be communicated between the robots? And should the communication be done at discrete intervals or continuous time communication. The following two figures illustrate the difference between topological and metric map. To simplify the problem, In this project,The metric map was used for the local and the global map, the data was shared at discrete time intervals and only the global map is shared between the robots.

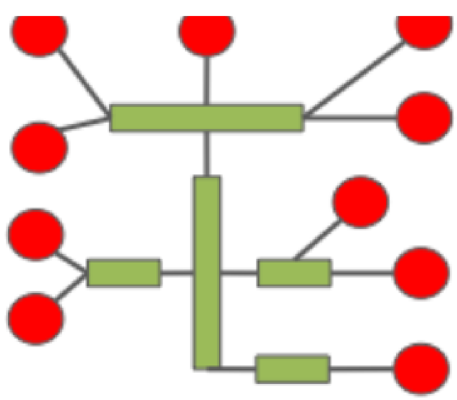

Fig. 1. Topological map.

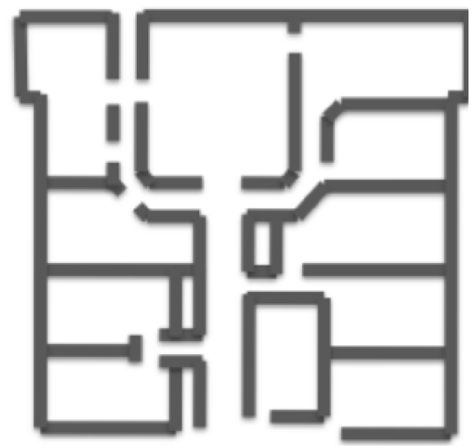

Fig. 2. Metric map.

2 SLAM PROBLEM

In the localization problem the map was given to the robot so the robot used the sensor data, the control actions and the given map to estimate its pose using either Kalman filter or Particle filter. However, In the mapping problem, the robot's pose was known and the map of the environment was the only unknown information using occupancy grid mapping algorithm which divides the entire space into a finite number of grids and the Binary Byes filter the robot was able to retrieve the map of the environment successfully. In SLAM problem neither the robot's pose nor the map is known so the robot has to solve the both problems simultaneously.

2.1 Particle Filter

Particles are virtual elements that resembles a robot. Each particle has a position orientation and represents a guess of where your robot might be. Monte-Carlo localization algorithm uses the robot's sensor measurements and particles to keep track of the robot position. First, the particles are spread uniformly and randomly throughout the map. Every particle has a weight which is an indication of how well the particle represents the robot states. After several iterations of MCL and re-sampling the particles, particles will converge and better estimate the robot pose.

2.2 Fast SLAM

Fast SLAM is a probabilistic algorithm that solves the full SLAM problem with known correspondences. it uses particles to estimate a posterior over the robot path, each particle holds a possible trajectory and a map where each feature in the map is represented by a Gaussian distribution. The main disadvantage of Fast-SLAM is that it assumes known landmarks positions which makes it unable to model arbitrary environment. Another instance of Fast-SLAM algorithm is Grid-based FastSLAM that adapts the FastSLAM to grid maps which extends the application of FastSLAM to work in environment where the landmarks positions are unknown. Like FastSLAM, In Grid-based FastSLAM each particle holds a guess of the robot trajectory and maintains its own map. The map will be updated by solving mapping with known poses problem using occupancy grid map algorithm.[1]

2.3 Grid-Based FastSLAM Algorithm

The Grid-based FastSLAM Algorithm could be reduced to three main building blocks demonstrated below: 1-Sampling Motion where the current pose is estimated given the k-th particle previous pose and the current controls u.

$$p(x_t|x_{t-1}, x_t)$$

2-Map Estimation where the current map is estimated given the current measurements, the current k-th particle pose, and the previous k-th particle map.

$$p(m_t|z_t, x_t^k)$$

3-Importance Weight which estimates the current likelihood of the measurement given the current k-th particle pose and the current k-th particle map.

$$p(z_t|x_t^k, m^k)$$

As seen from the above, Grid-based FastSLAM is capable of solving both the full SLAM and Online SLAM problems. FastSLAM estimates the full robot path, and hence it solves the Full SLAM problem. On the other hand, each particle in FastSLAM estimates instantaneous poses, and thus FastSLAM also solves the Online SLAM problem.

2.4 Multi Robot SLAM by Merging Occupancy Grid Maps

The following approach to multi-robot SLAM is based on merging local occupancy grid maps into a global map. This technique is simple as it does not rely on mutual encounters or relative position measurements but on finding and aligning overlapping regions between the local maps. Formally, the problem consist of finding a rotation and a translation between two occupancy grid maps m1 and m2 such that the overlap between m1 and the transformed m2 is maximized. The rotation and the translation between m1 and m2 could be parameterized by three variables grouped together in the following matrix.

$$M = \begin{bmatrix} cos(\theta) & -sin(\theta) & tx \\ sin(\theta) & sin(\theta) & ty \\ 0 & 0 & 1 \end{bmatrix}$$

To solve for the three parameters multiple optimization techniques could be employed, the simplest method is to optimize an heuristic function defining the quality of the alignment in terms of overlap and coherence. The following figure illustrates the merging problem.

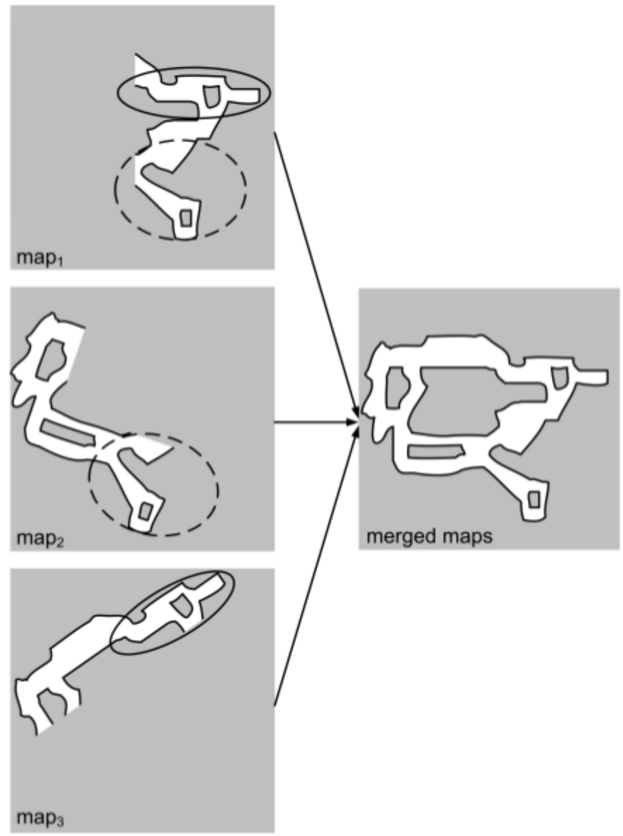

Fig. 3. Merging maps example.

3 Simulation and Results

3.1 Robot Description

For this project Two mobile robots were used, each equipped with a laser scanner, IMU(inertia measurement unit) and an odometry sensor. In the first part of the simulation, a small world is used where one robot was able to perform SLAM and then navigate through the environment using Adaptive Monte Carlo Localization algorithm (AMCL) for localization and A* Algorithm for motion planning. In the second part, A larger environment was created where two robots started at different locations in the environment, each robot performed the SLAM problem and sent its local map then the two local maps were merged together into one coherent global map.

3.2 Mapping With One Robot

The Figures below show the Robot World, the generated map using Gmapping and the navigation hierarchy.

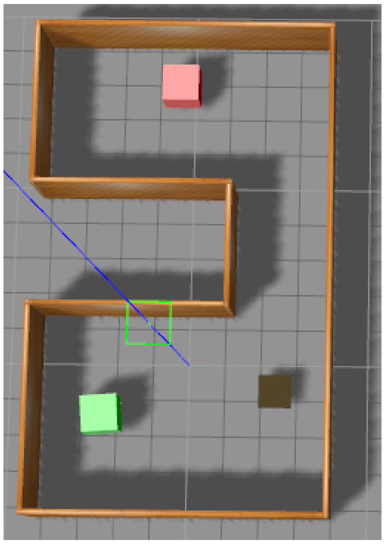

Fig. 4. One Robot World

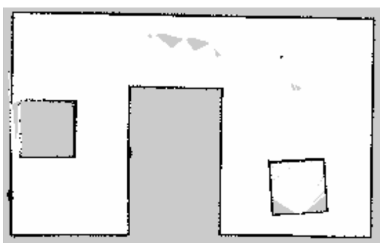

Fig. 5. Generated Map

3.3 Mapping With Multiple Robots

For Multi-robot SLAM, the world is shown below followed by the generated local maps versus the global map and finally the global map after the robots spend some time navigating through the environment. The package for this work could be found here : https://github.com/Esraa-Magdy-Mahmoud/coslam

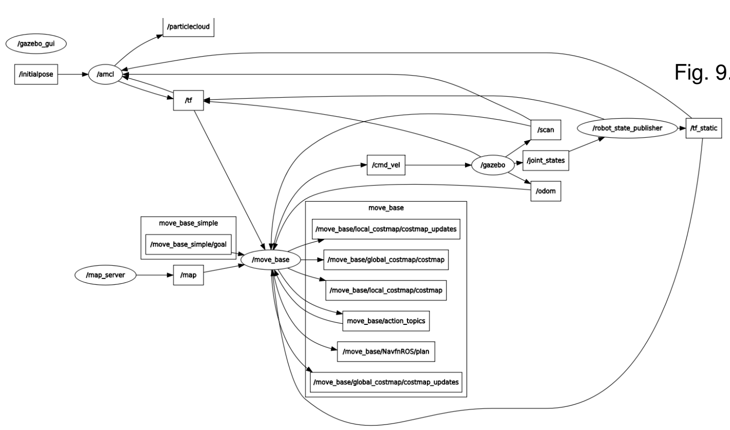

Fig. 6. Navigation Hierarchy

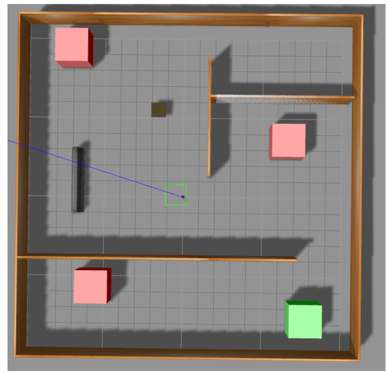

Fig. 7. Multi Robot World

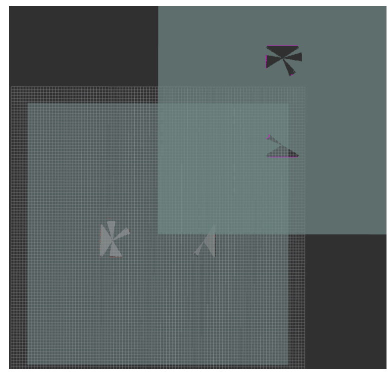

Fig. 8. Generated Maps

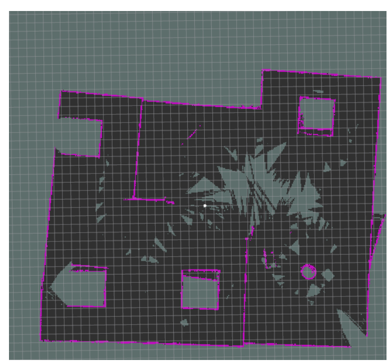

Fig. 9. Merged Map

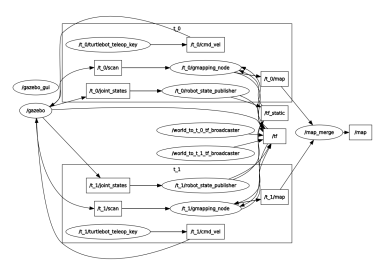

Fig. 10. Multi-Robot

4 Future Improvements

The quality of the generated map could be further improved, other methods for map merging might be implemented to compare the accuracy of merging. In addition, Methods for multiple robots navigation will be investigated.

References

[1] S. Thrun, M. Montemerlo, D. Koller and B. Wegbreit, ”FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem”.

[2] S. Thrun and M. Montemerlo, ”The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures”, The International Journal of Robotics Research, vol. 25, no. 5-6, pp. 403-429, 2006.

[3] Simon Bultmann, Latitia Matignon, Olivier Simonin. Multi-Robot Navigation and Cooperative Mapping in a Circular Topology. [Research Report] INSA Lyon. 2017. ¡hal-01526089¿