Encounter Based Multi Robot Simultaneous Localization and Occupancy Grid Mapping

RB Choroszucha[1], C Hyman[2], and A Collier[3]

***Abstract*— In this paper we present and implement the algorithm in Howard's 2006 paper: "Multi-robot simultaneous localization and mapping using particle filters." We also make a small modification to allow for each robot to have its own particle filter and qualitatively demonstrate that it performs better than the current state of the art.**

I. Introduction

The automated exploration of unknown environments has become one of the foremost challenges in mobile robotics. For a robot to explore an environment, it must map the environment and concurrently localize itself within the environment. The framework used to perform this task is known as simultaneous localization and mapping (SLAM) and has been well covered in the literature using a variety of techniques [4], [1].

While SLAM is well known and has a rich history of successes using a single robot, it can often be a slow process due to both constraints on the robot, such as speed and data processing, and lack of redundancy, i.e. robot failure [8], [3]. To address the speed of mapping and to add redundancy, coordinated or multi-robot SLAM (MRSLAM) was created.

MRSLAM is as it sounds, SLAM using multiple exploring robots. This approach allows for the partition of the physical search space using different robots, typically decreasing the time it takes to map an area, and increasing the likelihood of full map coverage in the event of robot failure. This temporal exploration parallelism does, however, come at the cost of added complexity. The added complexity of MRSLAM comes in two major components: coordination of exploration using multiple search agents and merging the maps of these agents [5]. Coordinated exploration consists of planning the groups' search path, most often formulated to cover the most search space in minimum time or to optimize some other mapping cost criteria. The other major component, map merging, is the combination of individual robot's observations and maps into one cohesive global map estimate.

In this paper we use the map merging algorithm presented in Howard's 2006 paper: "Multi-Robot Simultaneous Localization and Mapping using Particle Filters" [6]. This paper is of particular interest as it presents a resource efficient online solution to the MRSLAM map merging problem given unknown initial robot poses.

The algorithm is an encounter-based algorithm. Given that we start with one robot mapping (call this robot 1), when it encounters another robot (robot 2), the relative pose between the two robots is computed and robot 2's odometry and scan data is transmitted in reverse order to robot 1 and integrated to create a single map posterior.

Fig. 1 depicts the process with robot 1's triple $(x_t^1, u_{t-1}^1, z_t^1)$, and robot 2's triple $(x_t^2, u_{t-1}^2, z_t^2)$. When $t = 2$, robot 1 observes robot 2 and determines the relative pose Δ, then the past odometry and scan data is integrated in reverse order.

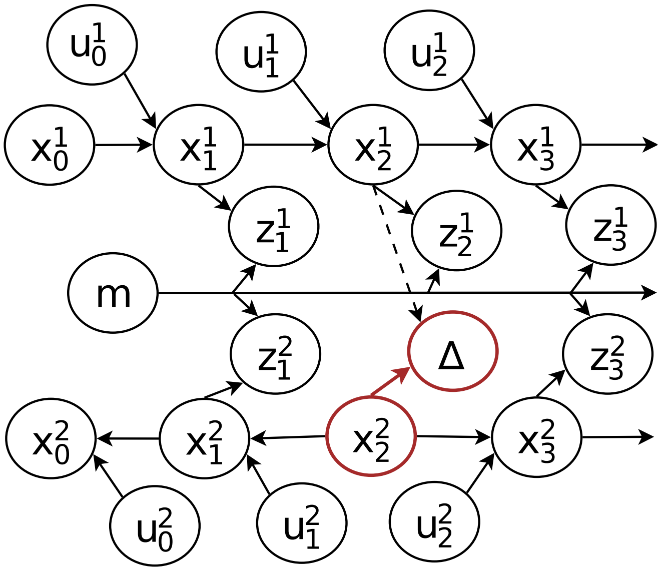


Fig. 1. Depiction of data integration. Taken from [6].

The paper is structured as follows: In §II some background about MRSLAM is provided, the contribution of [6] is discussed and a modified scheme is presented. In §III the algorithm of [6] is discussed, and in §IV experimental results are provided for a simulated example and the Albert B data set [7].

II. Background

A. State of the Art

In the case where all relative robot poses are known, merging maps is a trivial problem using small modifications to existing SLAM techniques [8]. In the general MRSLAM problem, however, robots may start with unknown absolute and relative poses, and therefore merging of maps requires the discovery of relative relationships between different robot

[1] RB Choroszucha is with the School of Naval Architecture and Marine Engineering, University of Michigan. E-mail: riboch@umich.edu

[2] C Hyman is with the School of Electrical Engineering and Computer Science, University of Michigan. E-mail: hymanc@umich.edu

[3] A Collier is with the School of Electrical Engineering and Computer Science, University of Michigan. E-mail: amcollie@umich.edu

trajectories to build a single map. This is often a costly process, which in general can be solved for robots sharing a search space by estimating each robot's relative pose given a partial map, but this leads to exponential complexity with respect to the number of exploring robots[5]. Nevertheless, several practical algorithms exist to circumvent this naive and inefficient approach, including coarse topological matching and stitching techniques borrowed from computer vision [2].

B. *Extension to Original Work[6]*

¡Independent weights per pose: Richard¿

III. The Algorithm

A. *Overview*

Howards multi-robot SLAM algorithm generates a single map and pose posterior much in the same way an occupancy grid map SLAM algorithm based on a Rao-Blackwellized particle filter (RBPF) would, but with additions to accommodate multiple robots. Mapping begins with a set consisting of a single robot with known pose and an occupancy grid map of the environment is built as this robot traverses and measures it. The ultimate goal of the algorithm is to simultaneously compute for time t, the full SLAM posterior containing all robot pose trajectories $x^i_{1:t}$ and the global map $m_{1:t}$, given only a single known initial pose and sets of measurements

$$p(x^1_{1:t}, x^2_{1:t}, ..., x^M_{1:t}, m_{1:t} | x^1_0, z^1_{1:t}, u^1_{0:t}, \Delta^2_s, z^2_{1:t}, u^2_{0:t}, \Delta^3_s, ..., \Delta^M_s, z^M_{1:t}, u^M_{0:t})$$

The odometry and measurement data of all other robots is stored as it is collected, as it cannot contribute to the global map without a known relative pose linking it to the frame of the first robot. As the first robot encounters additional robots via a mutual pose observation, the newly observed robot is added to the set of mapping robots and its actions and the map is sequentially conditioned by its actions and measurements. From the point of observation, all future actions and measurements of the new robot are used to condition the map posterior, and likewise all of its previously stored actions and measurements are played back in reverse order from that point as a virtual robot. This information is then passed to a pair of new particle filters for the pose of these robots that contribute to the same global map. As further encounters occur between any robot, real or virtual, in the mapping set and a previously unseen robot, the previously unseen robot is added to the set of mapping robots and its measured relative pose is again used to establish a reference point with which its actions and measurements can condition the map posterior. Further mutual observations of two robots already in the mapping set are ignored for the sake of simplicity. This encounter-add process is continued recursively for all robots until all stored and future odometry and measurements are exhausted or all possible mapping robots have made an encounter with the mapping set. Beyond this point, the algorithm behaves as a normal RBPF with a stacked state containing pose posteriors of all robots in addition to the map.

This algorithm relies on a number of assumptions that are requisite for it to effectively solve the MRSLAM problem. Firstly, for all explored regions to count towards the map, each mapping robot must have encountered a robot that is an element of the mapping set in order to make a fully connected graph of robot poses, and therefore maximally complete map. Additionally, each robot pose trajectory is independent of all other trajectories such that motion or observation from one robot does not affect another outside of encounter events [6].

IV. Experimental Validation

$$\begin{bmatrix} x_t \\ y_t \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \theta_{t-1} \end{bmatrix} + \begin{bmatrix} dx_t \cos(\theta_t + d\theta_t) \\ dy_t \sin(\theta_t + d\theta_t) \\ \theta_{t-1} + d\theta_t \end{bmatrix} \tag{1}$$

V. Conclusion

In this paper we have presented Howards algorithm from [6] for the integration of data from several independent robots, and demonstrated a successful implementation. Further, we have modified the algorithm to use a particle filter for each robot and demonstrated that it can qualitatively

References

[1] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[2] Andreas Birk and Stefano Carpin. Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, 94(7):1384–1397, 2006.

[3] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386, 2005.

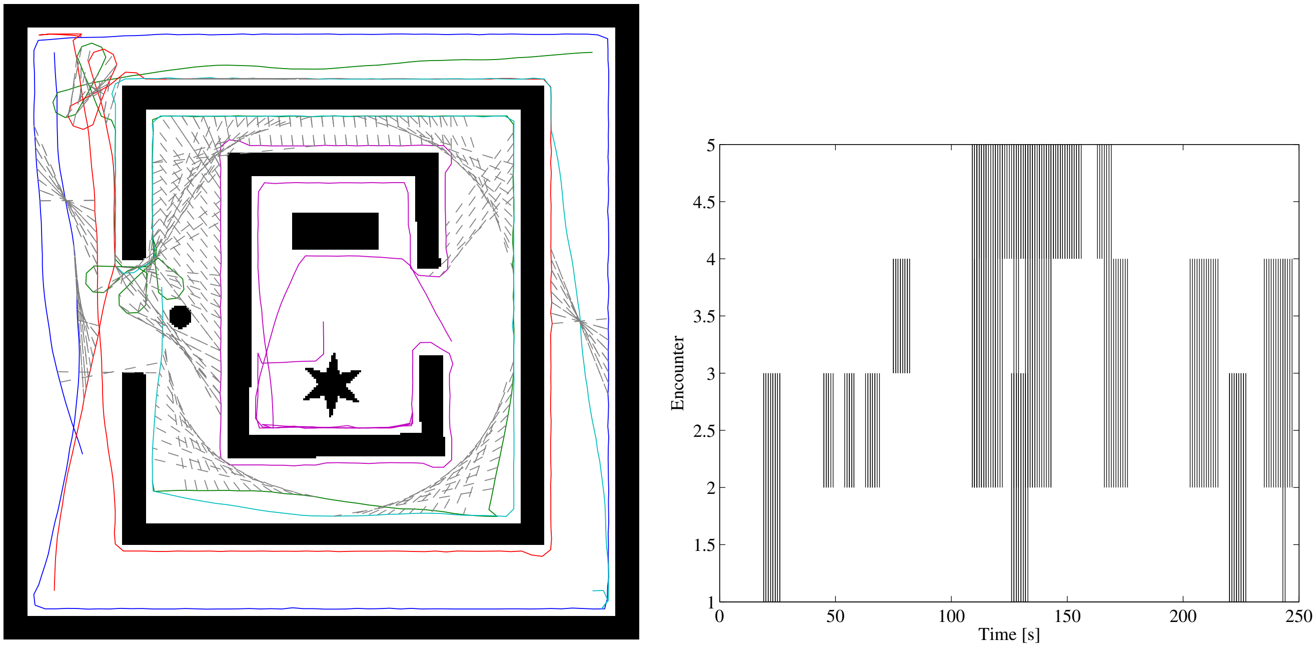
[4] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[5] Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Dirk Schulz, and Benjamin Stewart. Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339, 2006.

[6] Andrew Howard. Multi-robot simultaneous localization and mapping using particle filters. *The International Journal of Robotics Research*, 25(12):1243–1256, 2006.

[7] Andrew Howard and Nicholas Roy. The robotics data set repository (radish), 2003.

[8] Sebastian Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5):335–363, 2001.



(a) The test geometry, robot paths, and encounters (gray).

(b) Robot encounters at a given time.

Fig. 2.