Cooperative Multi-robot Map Merging Using Fast-SLAM

N. Ergin Özkucur and H. Levent Akın

Boğaziçi University, Department of Computer Engineering,
Artificial Intelligence Laboratory, 34342 Istanbul, Turkey
{nezih.ozkucur,akin}@boun.edu.tr

Abstract. Multi-robot map merging is an essential task for cooperative robot navigation. In the realistic case, the robots do not know the initial positions of the others and this adds extra challenges to the problem. Some approaches search transformation parameters using the local maps and some approaches assume the robots will observe each other and use robot to robot observations. This work extends a previous work which is based on EKF-SLAM to the Fast-SLAM algorithm. The robots can observe each other and non-unique landmarks using visual sensors and merge maps by propagating uncertainty. Another contribution is the calibration of noise parameters with supervised data using the Evolutionary Strategies method. The developed algorithms are tested in both simulated and real robot experiments and the improvements and applicability of the developed methods are shown with the results.

1 Introduction

One of the problems in robot navigation is the simultaneous localization and mapping problem (SLAM). The problem addresses a robot generating a map of an unknown environment and localizing itself in this map. The RoboCup @Home and Rescue [1] leagues already require exploration and map building for the high level planning tasks. On the other hand, localization in the soccer leagues like Standard Platform and Middle Size gets harder due to decreasing number of unique landmarks, so the problem moves towards the SLAM problem. Since the environment is unknown, landmark observations do not include identity information. This ambiguity adds an extra challenge and is called the *data association problem*. In multi-robot systems, the cooperative map building task introduces additional challenges. The robots should transform and merge their own maps and resolve ambiguities. The problem attracts researchers because the solution provides more autonomy to robots and allows them to operate in more realistic application domains.

The single-robot SLAM problem is a more or less solved problem. In the EKF-SLAM algorithm [2,3], the landmark positions and the robot pose (position and orientation) form an augmented state where the belief state is represented as a Gaussian distribution and is updated using Extended Kalman Filter (EKF) [2]. Another well-known solution method called Fast-SLAM [4,5] uses the fact that given the robot pose, the landmark positions are independent. The belief state in this case is a set of particles where each particle is a robot pose and the associated map hypothesis. In both algorithms, the uncertainty in the odometry and the observations is assumed to have a Gaussian distribution [6]. The performance of the methods highly depends on the parameters of the actual noise distributions, which are generally unknown. In [7], the problem is solved with the least square approach and in [8], the optimal parameters are found with Expectation Maximization. In this paper we propose employing Evolutionary Strategies to search for the optimal parameter set, using the robot's ground truth position information.

In multi-robot systems, current solution methods can be categorized with respect to their assumptions. The most simplistic assumption is that the robots know their initial positions. In this case, the single-robot solutions can easily be extended for the cooperative case [9]. In more realistic cases, the robots do not know or observe the positions of others. In [10,11,12,13], the transformation parameters are searched with different heuristics using only the local maps of the robots. In [14,15], the idea is about localizing the robots within the maps of the other robots and finding the transformation hypothesis. In another assumption category, the robots can observe each other and are expected to meet at some point. In this case, the problem reduces to finding suitable transformation parameters using the robot to robot observations. In [9], the robots record their observation history and provide it to the other robots when they meet, so that the other robots can use the history to build the map of unexplored areas using Fast-SLAM algorithm. In [16], similar to the previous method, the transformation parameters are found with robot to robot observations, and instead of using observation history, current map estimations are merged.

This work is an extension of the map merging algorithm based on EKF-SLAM in [16] to the Fast-SLAM algorithm. The map merging is performed when robots meet using the robot to robot observations. However, uncertainty propagation in Fast-SLAM differs from the EKF-SLAM case because each particle has its own map estimation instead of single map estimation. In [9], this problem is addressed differently by recording the observation history. In this paper, we exploit the Markovian assumption of state representation and merge the most recent map estimations.

The rest of this paper is organized as follows. In Section 2, our methodology and assumptions are detailed. In Section 3, the experiment setups in both simulation and real world are explained and results are discussed. Finally Section 4 summarizes and concludes our work and points out some possible future work.

2 Proposed Approach

2.1 Map-Merging

For the simplicity of illustration of the methods, in the multi-robot case, we assume that there are two robots, exploring some part of the environment and eventually meet at some point where they can observe each other. When they meet, they inform each other and share knowledge to merge the map of the other robot with their own-map. In the map-merging case, we only focus on the instant where a robot receives the estimation of the other robot and since our method is distributed, we will consider the situation from the point of view of only one robot.

The message from the other robot includes:

- M_{other}: the map estimation of the other robot. Each entity $m_{other,i}$ contains the position $p_{other,i}$ and a 2x2 covariance matrix $\Sigma_{other,i}$ of the ith landmark.
- p_{other}: the pose of the other robot. Note that p_{other} is the other robots pose and $p_{other,i}$ is the position of the ith landmark of other robot.
- $z_{other,self} = \{l_{other,self}, \theta_{other,self}\}$ is the observation of the other robot to self robot.

In the map estimation M_{other}, the correlation between the landmarks are omitted except the 2x2 covariances of landmark positions. Extracting this information from the EKF-SLAM is trivial, however in the Fast-SLAM, we take the weighted mean of the map estimation of particles with the importance weight.

When the robot receives the message from the other robot, it first calculates the transformation matrix between the coordinate frames:

$$T = \begin{bmatrix} \cos(tr_\theta) & -\sin(tr_\theta) & tr_x \\ \sin(tr_\theta) & \cos(tr_\theta) & tr_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

which has the translation parameters $\{tr_x, tr_y\}$ and the rotation parameter tr_θ. Figure 1 illustrates the geometric configuration of coordinate frames and parameters in the information sharing step. Calculating the transformation parameters is an analytic geometry problem so it is skipped for simplicity. With the transformation matrix, each entity in the incoming map is transformed with the following equations:

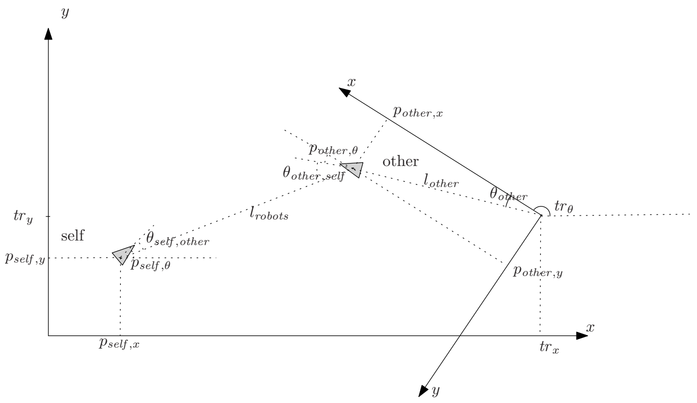


Fig. 1. The configuration and parameters when robots observe each other

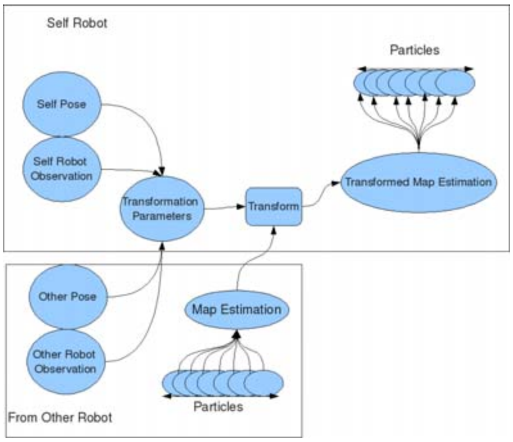


Fig. 2. Diagram of multi-robot map merging algorithm with the Fast-SLAM method

$$\begin{bmatrix} p_{other,x,i\prime} \\ p_{other,y,i\prime} \\ 1 \end{bmatrix} = T \begin{bmatrix} p_{other,x,i} \\ p_{other,y,i} \\ 1 \end{bmatrix} \tag{2}$$

$$\Sigma_{other,i\prime} = T^T \Sigma_{other,i} T \tag{3}$$

After transforming, the incoming map is merged into the map of each particle. Figure 2 gives the summary of the transformation and the merging operation in the Fast-SLAM algorithm. If N is the number of particles and M is the number of landmarks, the complexity of the merging step is $O(NM^2)$ with a naive implementation.

Nearest neighbor method is used to find duplicate landmarks when merging the map of the other robot with a single particle's map. If a landmark is a new landmark, it simply is added to the map. If the landmark is also known by itself, the other robot's estimation is considered as an evidence and the resulting state is calculated as:

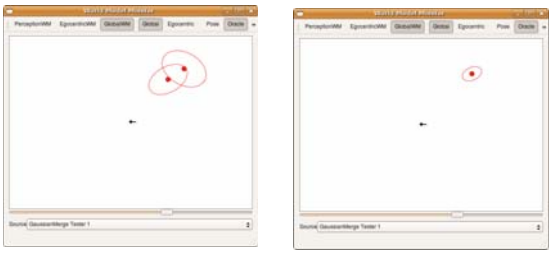
$$\Sigma_{merged} = \Sigma_{self} - \Sigma_{self} \left[\Sigma_{self} + \Sigma_{other\prime}\right]^{-1} \Sigma_{self} \tag{4}$$

$$p_{merged} = p_{self} + \Sigma_{self} \left[\Sigma_{self} + \Sigma_{other\prime}\right]^{-1} (p_{other\prime} - p_{self}) \tag{5}$$

The merging of two Gaussians is displayed in Figure 3. Note that the uncertainty decreases in the resulting Gaussian.

2.2 Parameter Calibration of the Kalman Filter

The parameters of the Kalman Filter are the initial uncertainty, odometry reading noise and the observation noise. In the SLAM application, the state vector size is $3 + 2L$ where three for robot pose $\{p_x, p_y, p_\theta\}$ and two for each landmark $m_i = \{p_x^i, p_y^i\}$ in the map M with size L. The initial uncertainty has a covariance matrix P with size $(3 + 2L) \times (3 + 2L)$. An observation is represented as



(a) Two Gaussian (b) Merged Gaussian

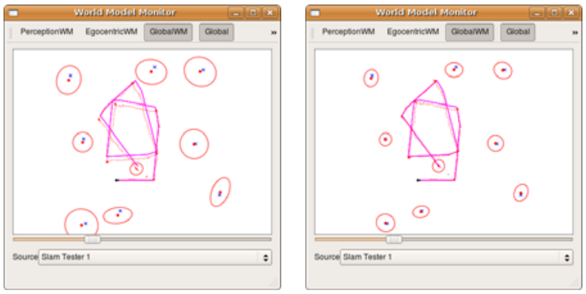
Fig. 3. Merging two Gaussian

the distance and orientation $z_i = \{l_i, \theta_i\}$ and therefore the observation noise covariance matrix Q has size 2×2. The odometry reading is the displacement of the robot in two dimensions and the change in the orientation $u = \{\Delta_x, \Delta_y, \Delta_\theta\}$ so the process noise covariance matrix R has size 3×3. These three covariance matrices form a very large parameter set, however we can reduce the number of parameters using some basic assumptions. The noise in all dimensions are assumed to be independent, so the correlation values become all zero. In addition, the noise on the x and y axes are assumed to be same. Finally, when we add the process noise or the landmark location, the parameter set becomes:

$$Q = \begin{pmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{pmatrix}, R = \begin{pmatrix} \omega_3 & 0 & 0 & 0 & 0 & \dots \\ 0 & \omega_3 & 0 & 0 & 0 & \dots \\ 0 & 0 & \omega_4 & 0 & 0 & \dots \\ 0 & 0 & 0 & \omega_5 & 0 & \dots \\ 0 & 0 & 0 & 0 & \omega_5 & \dots \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{pmatrix}, P = \begin{pmatrix} \omega_6 & 0 & 0 & 0 & \dots \\ 0 & \omega_6 & 0 & 0 & \dots \\ 0 & 0 & \omega_7 & 0 & \dots \\ 0 & 0 & 0 & \omega_8 & \dots \\ . & . & . & . & . \\ . & . & . & . & . \end{pmatrix} \quad (6)$$

where the vector $\omega = \{\omega_1, \omega_2, ..., \omega_8\}$ forms our actual parameter set to be estimated using Evolutionary Strategies [17]. A chromosome represents a possible combination of these values. An episode is executed with the parameter set of an individual to calculate the fitness value, which is the mean distance between the robot pose estimation and the actual (ground truth) pose. If the error grows very large, the episode is terminated to avoid unnecessary continuation and a small fitness value is returned with respect to the episode. As new generations are formed, they result with smaller errors on the robot position estimation. Note that only the ground truth position information of the robot is used as the supervised data.

In the Fast-SLAM method, the particles are sampled with a Gaussian distribution using the odometry readings. The importance weight calculations with the observations are also made with Gaussian likelihood function. For these reasons, the parameter set we estimated can be explicitly used for the Fast-SLAM algorithm. The only extra parameter is the number of particles which affects



(a) Hand Tuned (b) Calibrated

Fig. 4. Path comparison of the hand tuned and calibrated parameters in a single episode

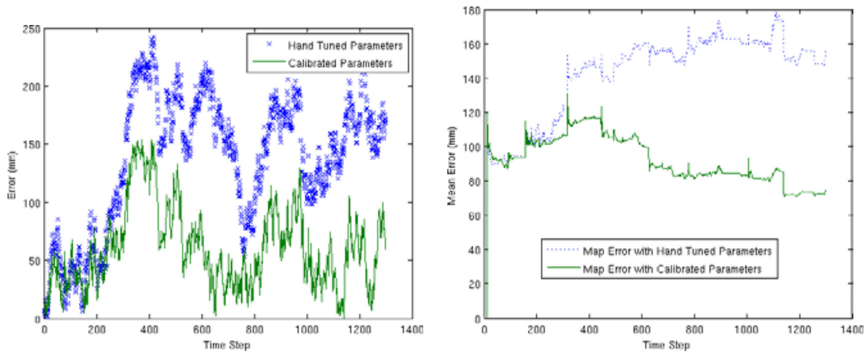
the quality of the posterior distribution. In our method to select the number of particles, we perform successive experiments with increasing number of particles until performance convergences.

3 Experiments and Results

We used the Festo Robotino robot [18] as the hardware platform. It has omni-directional locomotion ability and a webcam which provides RGB images with dimensions 320x240 and 50 degrees field of view. We also equipped the robot with a URG laser range finder device [19] to generate occupancy grid map and to avoid obstacles. The laser range finder has 5 meters range and 270 degrees field of view, however physical placement of the device allows 140 degrees of field of view. We implemented the software with the *Player/Stage* framework [20], which includes a 2D simulator and provides the ability to test our algorithms on both simulation environment and real-world by only changing the configurations. In the simulation environment, ground truth positions are directly accessed. In the real world, we used an overhead camera system [21] to measure the global position of the robots.

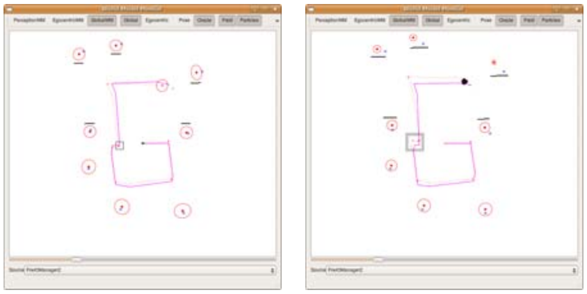
3.1 Parameter Calibration Results

The first experiment is designed to compare the hand-tuned and the calibrated parameters in the simulation environment. The simulator provides perfect knowledge, however we inject Gausian noise to both observations and the odometry. The search for the optimal set is initiated from the hand tuned parameters. Figure 4 shows the result of the path and map estimations of both parameter sets. The straight line is the exact path, the dots are the estimated path, the cross marks are the exact location of the landmarks and the ellipses are the landmark estimations. For the simulation scenarios, the robots followed a predefined path.



(a) Path estimation error (b) Map estimation error

Fig. 5. Performance comparison of the hand tuned and calibrated parameters in a single episode



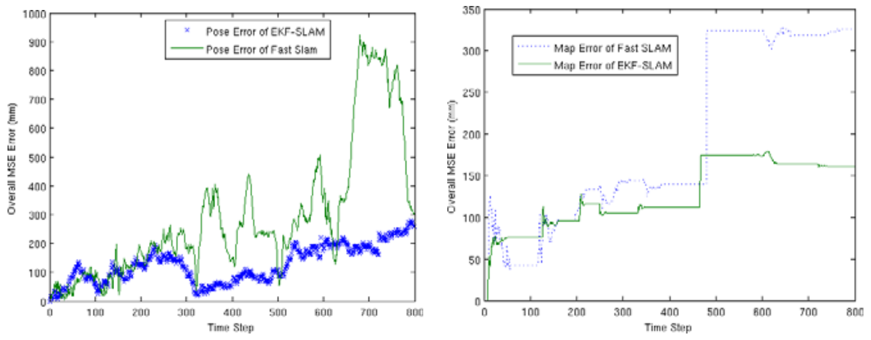
(a) EKF-SLAM (b) Fast-SLAM

Fig. 6. Path and map estimation results in the simulated map-merging experiment

In Figure 5, the errors in landmark and path estimations are given. The path error plots show that the calibrated parameters bounded the odometric error more accurately. The map estimation error is the mean error of known landmark estimations. Another interesting result is that the accuracy of the map increases even if we discard the orientation and landmark errors in the fitness function. This shows that an accurate map estimation requires an accurate path estimation.

3.2 Simulated Experiment Results

In the first experiment of map merging, we demonstrated the applied methods for both the EKF-SLAM approach and the Fast-SLAM approach in the simulation environment. In Figure 6, the path and map estimations are given for one robot. The squared region in the path is the meeting point where map-merging



(a) Pose Errors (b) Map Errors

Fig. 7. Pose and map errors for EKF-SLAM and Fast-SLAM methods in the simulated map-merging experiment

occurred. The overlined landmarks also exists in the incoming message, so they are merged, and underlined landmarks are added from the incoming message. Note that the added landmarks have the same biased error. This is the effect of the errors in position estimation and the robot observations in the map-merging step. In Figure 7, the path and map estimation errors are given. The biased error can be observed in the map estimation error. The error increases considerably when the landmarks with biased error are added to the map.

3.3 Real World Experiment Results

We also performed real world experiments for the map-merging algorithm using Fast-SLAM algorithm. In Figure 8, the experiment setup for the real-world is

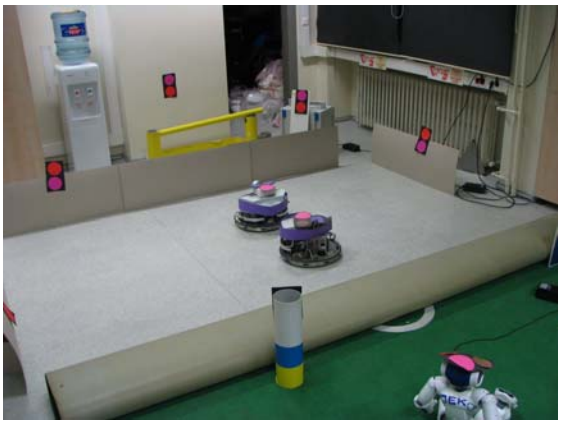


Fig. 8. Experiment setup of the real-world map-merging experiment

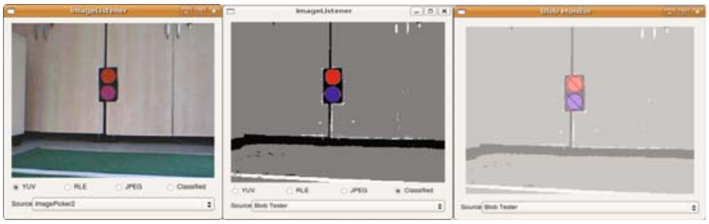
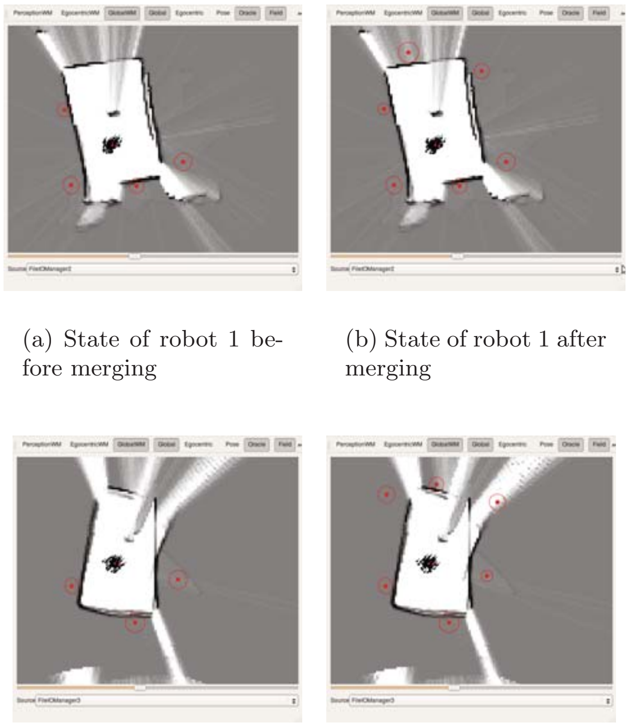


Fig. 9. Marker observation process. From left to right: raw image, color segmented image, and formed blobs.


(a) State of robot 1 before merging

(b) State of robot 1 after merging

(c) State of robot 2 before merging

(d) State of robot 2 after merging

Fig. 10. Map-merging result in the real world with Fast-SLAM method

given. The robots can observe colored landmarks around the world and start facing opposite directions. They are also marked with colored paper in order to enable robot to robot observations.

Landmark Detection. We used two-colored markers as landmarks. The images taken from the camera are in RGB format. To recognize the regions in the markers, we segmented the color space into four colors using the Generalized Regression Neural Network (GRNN) [22]. After segmenting the pixels, markers are observed as regions of interested colors. The snapshots of these steps are given in Figure 9. The robot has monocular camera and to calculate the distance of the marker, we exploit the known size of the markers. We simply calibrated a non-linear function of region size which gives the distance of the marker.

The robots detect obstacles with the laser range finder in the direction of movement. The detected obstacles are represented as a line formed by laser readings. The robots always turn right as parallel to the obstacle line when they encounter an obstacle. Since they start facing opposite directions, they explore separate portions of the area and meet in the middle. In Figure 10, the states of both robots are given before and after the merging steps. As stated before, we also generate occupancy grid-world, which is visualized with intensity of the grids. The other robot can be observed on the occupancy map of the both robots. Before merging, the robots are not aware of all the landmarks, but after merging, they know all of the landmarks.

4 Conclusions

In this paper, we have two notable contributions to the literature. First, we calibrated the noise parameters of EKF-SLAM algorithm using Evolutionary Strategies. The other contribution is that we adapted the map merging method for EKF-SLAM method from the literature to the Fast-SLAM method. The main difficulty in this adaptation is the representation of multiple maps in the Fast-SLAM method. To overcome this, we extracted a single map estimation from the Fast-SLAM belief state along with the uncertainty information and merged this map with each particle's map. This method also allows execution of different SLAM algorithms on different robots. The exchanged map estimation between robots has a common format, so the estimator of the incoming map estimation is not important for the merger algorithms.

We tested the methods on simulation using data with artificially introduced noise for evaluation and training. The algorithms are also implemented and tested on autonomous robots in indoor environment. We showed that the multi-robot map-merging method can also be applied to the Fast-SLAM algorithm without loss of uncertainty knowledge.

The major obstacle between this work and a real life scalable multi-robot SLAM application is the assumption of unique markers in the robots. With non-unique robot markers, correspondence problem for robot observations makes the problem ore complicated and should be addressed.

Acknowledgments

This project is supported by Boğaziçi University Research Fund Project 06HA102 and Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 106E172 and 2228.

References

1. Robocup official site maintained by the RoboCup Federation (2009), `http://www.robocup.org/`
2. Welch, G., Bishop, G.: An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA (1995)
3. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: Autonomous robot vehicles, pp. 167–193 (1990)
4. Montemerlo, M.: FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association. In: CMU Robotics Institute (2003)
5. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Gottlob, G., Walsh, T. (eds.) IJCIA, pp. 1151–1156. Morgan Kaufmann, San Francisco (2003)
6. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
7. Carew, B., Belanger, P.: Identification of optimum filter steady-state gain for systems with unknown noise covariances. IEEE Transactions on Automatic Control 18(6), 582–587 (1973)
8. Ghahramani, Z., Hinton, G.E.: Parameter estimation for linear dynamical systems. Technical report (1996)
9. Howard, A.: Multi-robot simultaneous localization and mapping using particle filters. Int. J. Rob. Res. 25(12), 1243–1256 (2006)
10. Thrun, S., Liu, Y.: Multi-robot slam with sparse extended information filers. I. J. Robotic Res. 15, 254–266 (2005)
11. Birk, A., Carpin, S.: Merging occupancy grid maps from multiple robots. Proceedings of the IEEE 94(7), 1384–1397 (2006)
12. Carpin, S.: Fast and accurate map merging for multi-robot systems. Auton. Robots 25(3), 305–316 (2008)
13. Huang, W.H., Beevers, K.R.: Topological map merging. Int. J. Rob. Res. 24(8), 601–613 (2005)
14. Konolige, K., Fox, D., Limketkai, B., Ko, J., Stewart, B.: Map merging for distributed robot navigation. In: Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003 (IROS 2003), October 2003, vol. 1, pp. 212–217 (2003)
15. Fox, D., Konolige, K., Limketkai, B., Ko, J., Schulz, D., Stewart, B.: Distributed multi-robot exploration and mapping. In: Proceedings of the 2nd Canadian Conference on Computer and Robot Vision, 2005, pp. XV–XV (May 2005)
16. Zhou, X.S., Roumeliotis, S.I.: Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2006, pp. 1785–1792 (2006)

17. Beyer, H.-G.: The Theory of Evolution Strategies. Springer-Verlag, Heidelberg (2001)
18. Festo robotino, `http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino/`
19. Urg laser range finder, `http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html`
20. Gerkey, B., Vaughan, R., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: 11th International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal (June 2003)
21. Kavaklıoğlu, C.: Developing a probabilistic post perception module for mobile robotics. Master's thesis, Boğaziçi University, Turkey (2009)
22. Alpaydın, E.: Machine Learning. MIT Press, Cambridge (2004)