Large Scale Experimental Design for Decentralized SLAM

Alex Cunningham and Frank Dellaert

Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332

ABSTRACT

This paper presents an analysis of large scale decentralized SLAM under a variety of experimental conditions to illustrate design trade-offs relevant to multi-robot mapping in challenging environments. As a part of work through the MAST CTA, the focus of these robot teams is on the use of small-scale robots with limited sensing, communication and computational resources. To evaluate mapping algorithms with large numbers (50+) of robots, we developed a simulation incorporating sensing of unlabeled landmarks, line-of-sight blocking obstacles, and communication modeling. Scenarios are randomly generated with variable models for sensing, communication, and robot behavior.

The underlying Decentralized Data Fusion (DDF) algorithm in these experiments enables robots to construct a map of their surroundings by fusing local sensor measurements with condensed map information from neighboring robots. Each robot maintains a cache of previously collected condensed maps from neighboring robots, and actively distributes these maps throughout the network to ensure resilience to communication and node failures. We bound the size of the robot neighborhoods to control the growth of the size of neighborhood maps.

We present the results of experiments conducted in these simulated scenarios under varying measurement models and conditions while measuring mapping performance. We discuss the trade-offs between mapping performance and scenario design, including robot teams separating and joining, multi-robot data association, exploration bounding, and neighborhood sizes.

Keywords: SLAM, Multi-robot, Decentralized Data Fusion, Evaluation

1. INTRODUCTION

Rapid, robust exploring and mapping challenging environments, such as in disaster response or on battlefields motivates the use of robot teams due to the advantages posed by their redundancy. For example, in a disaster response scenario, first-responders can deploy a team of small, cheap robots into a partially collapsed building to provide real-time, online feedback on the structure of the environment, the presence of hazards, or the locations of trapped survivors. In such a highly unstructured scenario, the redundancy of a multi-robot team enables greater resiliency to failures, such as communication failures or even robots being incapacitated or destroyed. A key component of fully exploiting the capabilities of a robot team is multi-robot perception, in which robots combine sensor information not only using their own internal sensor suites, but also between neighboring robots.

This sensor fusion problem, in a single-robot case, corresponds to the Simultaneous Localization and Mapping (SLAM) problem, in which a robot can recover its trajectory and a map of the environment from noisy sensor measurements. For the multi-robot case, the problem becomes more complex due to the need to transfer map information between robots balancing trade-offs in map quality and the communications and computational capabilities of the robot platforms. Evaluating these decentralized SLAM algorithms becomes more difficult due to the additional parameters to be varied, and the number of different corner cases that appear during the course of implementing such a system.

This paper presents an experimental approach for evaluating decentralized mapping systems, demonstrating an experimental procedure for planar simulated robots modeling a variety of real-world multi-robot scenario characteristics. As example decentralized inference approach, we evaluate the DDF-SAM[1–3] paradigm in a landmark-SLAM case and present results in a number of cases of interest.

Correspondence to A. Cunningham.: E-mail: alexgc@gatech.edu

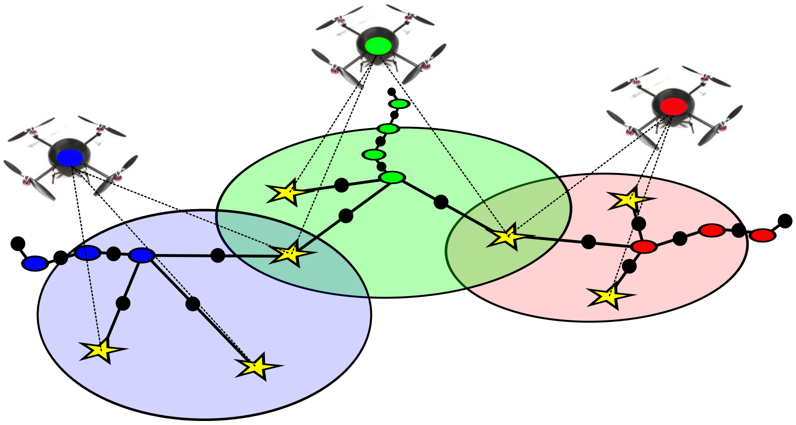

Figure 1. General structure of a multi-robot mapping scenario including three robots observing overlapping landmarks. The factor graph representing the system, including measurement factors to landmarks (stars), as well as the robot trajectory (colored circles) is projected below the robots for clarity.

In order to present a complete system, we will briefly describe single-robot SLAM, the multi-robot SLAM, and finally DDF-SAM to provide a background on the algorithms under evaluation, as well as highlight important challenge cases. The remainder of the paper will describe the simulation methodology and the experiments conducted, with results for DDF-SAM cases.

2. DECENTRALIZED SLAM

In recent years, the SLAM problem has had a flurry of work, with general surveys available[4–6] covering a variety of techniques, but this paper will focus on distributed graphical SLAM in a planar, landmark-based approach. Because much of multi-robot SLAM builds on single-robot SLAM, we will first cover the single-robot case before moving to the multi-robot case.

The basic structure of a typical single-robot SLAM system can be broken into two phases:

1. *Data Association Front-End*: This initial phase finds correspondences between newly observed features and previously observed features. These correspondences can come in the form of labellings for discrete features, as well as relative pose constraints that match a current sensor scan against a previous scans.
2. *Inference Back-End:* This phase optimizes over the constraints created in the data association front-end to find the most probable set of values for all variables.

While the front-end aspects the SLAM problem vary significantly by application domain, the inference machinery forms a more generalizable problem using abstractions afforded by probabilistic modeling. To narrow the focus of this paper, we will assume the known data associations throughout and focus on the back-end inference problem at the single-robot level.

The core representation underlying the smoothing and mapping (SAM) formulation is a *factor graph*, which is an undirected bipartite graph consisting of variable nodes, which represent the robot poses and landmarks, and *factors*, which encode measurements, such as those from odometry or visual feature detection. With the problem represented as a large sparse system, we cast fusing the measurements as a nonlinear optimization problem, where we minimize the error in all constraints over time. Fig. 1 illustrates a simple three-robot scenario with landmarks, robot poses and overlapping map regions.

$$\Theta^{*} = argmin_{\Theta}\frac{1}{2}\left\|h(\Theta) - Z\right\|_{\Sigma}^{2} \tag{1}$$

The standard cost function for this optimization problem is Equation 1, where the error is the difference between a generative measurement model $h(\Theta)$ for the current state Θ and the actual measurement Z, weighted by the measurement covariance matrix Σ. This paradigm has enabled the SLAM community to apply techniques from sparse linear algebra and graph theory to improve performance in batch solving cases or allow for efficient incremental solutions. This graphical approach has produced a large variety of work, including batch[4,7–14] and incremental solvers.[15–17]

In the multi-robot case, the underlying inference problem is an extension of the single robot case which specifically accounts for multi-robot system requirements. This problem is known as Distributed Data Fusion (DDF),[18–24] originally developed for distributed sensor networks, and specifies a set of constraints due to the decentralized nature of the system. A DDF solution must be robust to a communication or node failure, and seek minimize the both the communication cost of transmitting data to neighboring robots and the amount of computation performed on any individual platform. For a SLAM system, this motivates an approach where each robot performs local SLAM and then distributes this local mapping solution to other robots so that each robot can fuse them into a map containing both its local map and the map information from its neighbors. This problem is closely related to cooperative localization,[25–27] as well as true multi-robot multi-map approaches.[28–31]

The DDF-SAM approach[1,2] implements this pattern of individual robots computing a local solution for distribution to neighbors by having each robot maintain its own local map using standard incremental SAM,[17] and then create *condensed map*, which contains a summarized version of the local map in which all but a subset of variables are marginalized out. In a standard case, we marginalize out all of the robot poses from the local system, yielding a Gaussian density on the landmarks. These condensed maps become the basic pieces of data shared between robot, where each robot maintains a timestamped cache of all condensed maps it has received from neighbors. When robot is in contact with a neighboring robot, the robots shared condensed maps to update their respective caches with the latest versions. At any time, a robot can combine the condensed maps in its cache to form an *neighborhood graph*, which constrains all of the landmarks from the local robot and the neighboring robots, and optimize this new system to solve for the positions of landmarks in neighboring regions.

3. METHODOLOGY

The goal of the evaluation methodology presented in this paper is to use simulation to investigate a variety of common and important multi-robot scenarios with randomly generated scenarios controlled by relatively small number of parameters. The parameters controlling the generation of the scenario manage construction of the simulated environment, as well as the driving of the robots. This section describes general design goals for a landmark-based decentralized SLAM system and how these components should be exercised by an evaluation system. Implementation details for our custom simulation environment appear in the next section.

The basic scenario for multi-robot SLAM consists of a team of robots in a landmark-rich environment, in which each robot can maintain a local map solution, with measurements consisting of a motion model (from either robot wheel odometry or control inputs) and landmark observations. Throughout this paper, we use the planar SLAM case, in which robots have 2D poses with rotation (x, y, θ), and landmarks are 2D points (x, y). Each of these robots can then traverse the environment while maintaining a local SLAM solution.

In order to combine maps (in DDF-SAM, this is the neighborhood graph), it is necessary to find correspondences between features observed by separate robots, which forms the basis of the multi-robot data association problem. For planar landmark-based SLAM, the basic challenge is in matching constellations of landmarks to determine correspondences. The between-robot data association problem is beyond the scope of this paper, and we refer readers to our previous work[2] for details and additional background material. The key intuition for multi-robot data association is that the greater the overlap between local maps of neighboring robots, the easier it is to calculate landmark correspondences.

To develop scenarios of varying difficulties and expose key challenges presented in a DDF approach, we build examples from simpler cases to more complicated case. These cases provide the basis for the design parameterized scenarios.

The simplest case for a map fusion case to solve is one in which all of the robots start in the same location, with known locations of other robots. In this case, each robot can initially build the same local map, making the initialization of between-robot correspondences substantially easier. The difficulty of calculating these correspondences can be parameterized by changing the distances between the robots at the start of the scenario.

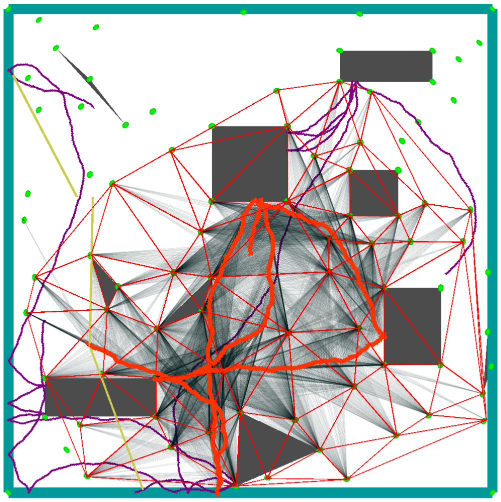

Figure 2. Example visualization of the simulation, showing the triangulation-based multi-robot data association technique,[2] as well as the ground truth positions of the landmarks and trajectories for three robots (purple). This also displays the mapping solution by one of the robots, with trajectory shown in red. The network connections between ground truth robot trajectories are shown in yellow. The translucent lines in this image are measurements between the red robot trajectory and landmarks.

We structure the main parameters for the construction the system as follows:

- *Size of Area*: The region that robots operate within is a square of fixed size, in which a 20m square provides enough space for 5 robots, while a 100m square is enough for 50 robots.
- *Number of Obstacles*: These obstacles limit line of sight and affect robot trajectories, and the
- *Number of Landmarks*: In addition to the landmarks at the corners of obstacles, we add landmarks to free space to provide sufficient landmark density for between-robot matching. If there are regions sparsely covered with landmarks, robots may have difficulty
- *Communication Range and Model*: By changing the constraints on communication between robots, such as by bounding the range or limiting by robot line of sight, we throttle the rate at which information propagates through the network. Communication delays and constraints also induce an information lag in between-robot map fusion.
- *Exploration Region Bound*: This bound forces a robot to only explore inside a bounded region of variable radius. For large values, robots will explore and grow their maps to much larger sizes from both direct observations of landmarks and information supplied by neighboring robots. For small values, the size of compressed maps remains well-bounded as few landmarks are added, though robots may not have as much map overlap with neighboring robots.
- *Start Location Distance*: This controls the distance between initial robot locations, such that small values correspond to robots starting next to each other, while large values force robots to explore before joining with other robots.

4. EXPERIMENTAL DESIGN

We developed a custom simulator covering the planar landmark-SLAM scenario, incorporating obstacles, limited line-of-sight, dynamic communication topology, and simple randomized trajectory control by robots. This simulator first appeared in our previous work[2] as a demonstration of a large scale simulated system, focusing on evaluating a multi-robot data association algorithm, and the effect of neighborhood bounding the size of the robot neighborhoods on communication bandwidth and map summarization times. In this paper, we will provide additional implementation details for the methodology to enable evaluation of varying robot systems.

The simulation and mapping system is implemented as nodes within the ROS middleware framework, which enables message passing between separate nodes. A message passing system not only provides a straightforward means of distributing the system, but also to provide measurement of communications bandwidth, expressed in terms of the serialized size of each ROS message. ROS also provides the visualization interface for the system. For small systems of 3-5 robots, we can treat each robot as having separate nodes for each DDF-SAM module, with a single simulator node broadcasting messages out to the robot system.

An important implementation detail for ensuring that timing can scale up to a large number robots is to perform updates for individual robot mappers in series rather than in parallel. In this structure, the simulator node remains unchanged from the distributed form, but all local mapping, communication and map fusion updates are performed in a single process, with each robot updated sequentially. We use this approach for larger robot teams which will quickly use all of the available hardware cores on a single development machine and impact timing results for stages in the process due to OS overhead from a large number of threads. With individual mapping updates performed sequentially, we can support 50 robots and ensure that the system load for any individual update step does not impact the results.

In order to ensure that the mapping algorithm does not become hand-tuned to a specific scenario, we generate the scenario randomly, placing fixed numbers of polygonal obstacles and point landmarks in the environment. The simulated environment is square with landmark features at the corners, and obstacles, landmarks and robots starting points are initialized in order. Obstacles consist of triangles and rectangles distributed throughout the simulated region, with edge lengths and angles chosen from a normal distribution specified as parameters for the scenario. Each obstacle corner also forms an observable landmark, and the obstacles are constructed such that no landmarks ever fall inside an obstacle, ensuring that no landmarks will become unobservable. After adding obstacles, we place a fixed number additional landmarks to the environment to allow for more dense landmark observations. An important note regarding choice of landmark and obstacle location: we choose candidate points from a uniform distribution over the environment, but throw out landmarks inside of an existing obstacle or within a sensing threshold of another landmark. Checking for points inside obstacles maintains landmark observability, and creating a minimum distance between landmarks ensures that nearest neighbor local data association techniques can use a simpler gating function.

In order to make the generation of robot trajectories easier to set up, we employ simple control models to drive each robot through the environment. At the most basic level, we use random walks through the environment in which robots will deflect off of obstacles and continue. We specify the robot movement with a constant-velocity unicycle model, using the steering rate as the primary control input. In addition to a random walk, in which robots will diffuse through the entire environment given enough time, we introduce simple biases to the robot trajectories to circle around their start locations. This biasing approach provides a parameterization for determining when robots transitioning from a exploratory role, in which they regularly observe new landmarks, a role in which they observe only known landmarks. Combined with constraints on minimum or maximum distance between robot starting locations, we can also address the effect of map overlap.

5. RESULTS

While we have not tested all combinations of scenario parameters, we present some comparisons of parameter choice to illustrate some of the scenarios discussed. In this particular case, we focused on the effect of biasing the robot trajectories on optimization performance and neighborhood system size.

In Fig. 3, we created a baseline scenario for comparison in which each robot was allowed unbounded exploration, resulting in a rapidly increasing neighborhood map size as the robots explored. This experiment used 50 robots, but was run for only 100 timesteps due to the rapidly increasing optimization time per robot.

To further examine the effect of biasing robot trajectories, we conducted several more tests, this time varying the bounds on robot exploration, with an unbiased control case. Fig. 4 shows the result of this experiment, in which 50 robots were run for 400 timesteps. Unbiased refers to unlimited exploration for each robot, and radius sizes are in meters. All experiments varying the size of a robot's exploration region were performed for a

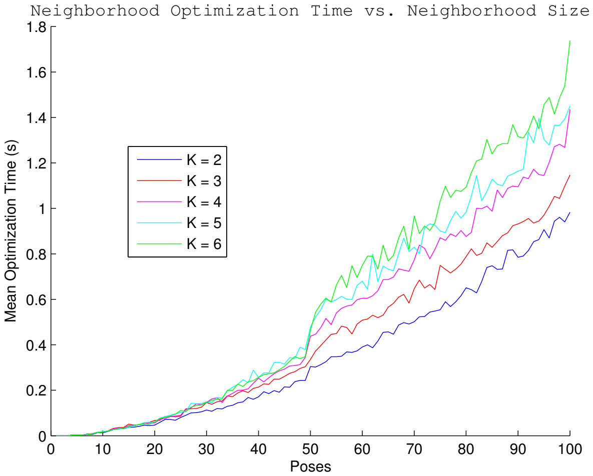

Figure 3. Comparison of neighborhood optimization times vs. time, separated by the neighborhood size K. Simulated in 50 robot case, with 100 poses per robot.

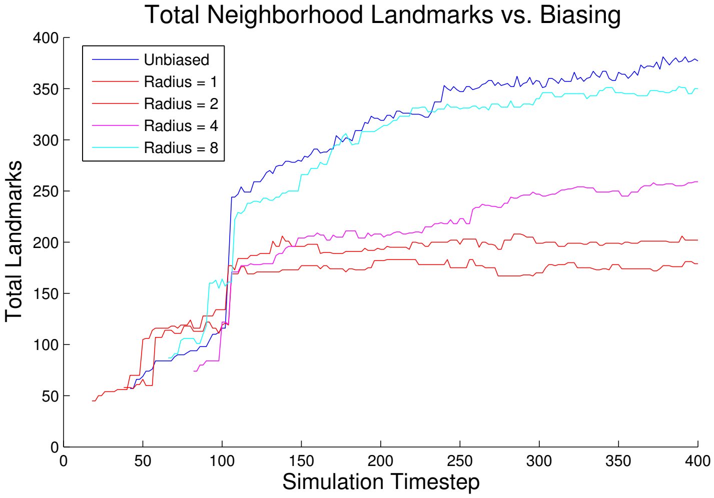

Figure 4. Plot of neighborhood system size in landmarks vs. time, while varying the exploration bound parameter.

neighborhood size $K = 4$ and are for only one of the 50 robots in the scenario to reduce experimental runtime. In this plot, the large jumps in the size of the neighborhood system indicate the addition of a new neighboring robot into the map.

Further illustrating the effect of robot trajectory bounding, Fig. 5 shows the growth in the size of incoming communication message sizes over time. This communication phase provides a bound on the largest messages transmitted between robots, as it denotes the combined size of the condensed map messages into each individual DDF-SAM communication module.

As another means of evaluating the complexity of the neighborhood system, Fig. 6 illustrates the growth in the number of data associations necessary.

6. SUMMARY

This paper presented analysis of a multi-robot experimental framework, incorporating parameters to adapt the scenario to highlight strengths and weaknesses of a decentralized SLAM system. The decentralized system under test in this case was the DDF-SAM approach, in which each robot performs local SLAM and then summarizes

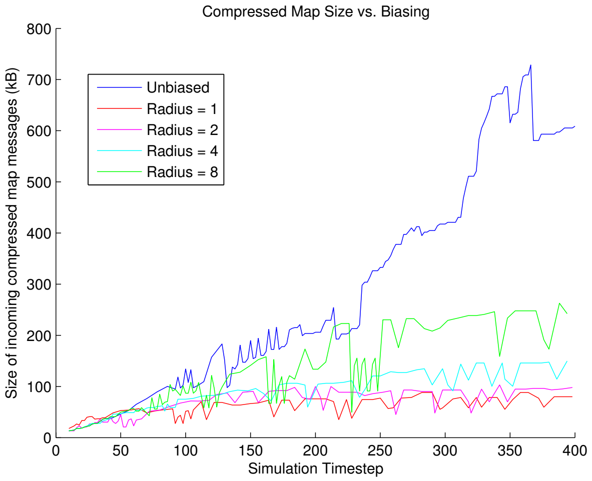

Figure 5. Plot of incoming communication message size vs. time, while varying the exploration bound parameter.

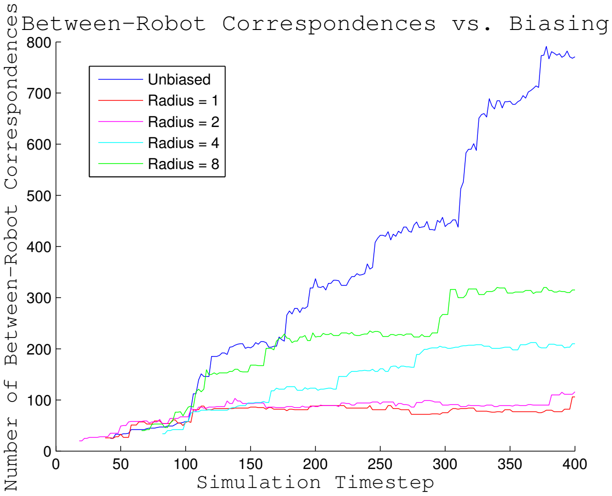

Figure 6. Plot of number of correspondences in neighborhood graph vs. time, while varying exploration bound parameter.

its map to distribute to its neighboring robots. The simulator provides a means to model various significant scenarios in multi-robot SLAM.

As a result from these experiments, the DDF-SAM approach does not scale well to long-term exploration, which will motivate future work in exploiting incremental techniques to allow more smaller message transfers and more manageable neighborhood graphs.

7. ACKNOWLEDGEMENTS

The research was performed as part of the Army Research Laboratory - Collaborative Technology Alliance on Micro Autonomous Systems Technology. The support is gratefully acknowledged.

REFERENCES

1. A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed slam using constrained factor graphs," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
2. A. Cunningham, K. Wurm, W. Burgard, and F. Dellaert, "Fully distributed scalable smoothing and mapping with robust multi-robot data association," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012. to appear.
3. J. G. R. III, A. Cunningham, M. Paluri, H. I. Christensen, F. Dellaert, N. Michael, V. Kumar, and L. Mathies, "Results of mast joint experiment 3.1," in *Defense, Security and Sensing*, SPIE, (Orando, Fl.), April 2011.
4. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT press, Cambridge, MA, 2005.
5. T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II state of the art," *Robotics & Automation Magazine* , Sep 2006.
6. H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *Robotics & Automation Magazine* , Jun 2006.
7. F. Dellaert, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," in *Robotics: Science and Systems (RSS)*, 2005.
8. F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. J. of Robotics Research* **25**, pp. 1181–1203, Dec 2006.
9. E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 2262–2269, May 2006.
10. G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Robotics: Science and Systems (RSS)*, Jun 2007.
11. G. Grisetti, C. Stachniss, and W. Burgard, "Non-linear constraint network optimization for efficient map learning," *Trans. on Intelligent Transportation systems* , 2009.
12. K. Ni and F. Dellaert, "Multi-level submap based slam using nested dissection," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
13. K. Ni, D. Steedly, and F. Dellaert, "Tectonic SAM: Exact; out-of-core; submap-based SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, (Rome; Italy), April 2007.
14. F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. Thorpe, "Subgraph-preconditioned conjugate gradient for large scale slam," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
15. M. Kaess, V. Ila, R. Roberts, and F. Dellaert, "The Bayes tree: An algorithmic foundation for probabilistic robot mapping," in *Intl. Workshop on the Algorithmic Foundations of Robotics*, Dec 2010.
16. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, (Shanghai, China), May 2011.
17. M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics* **24**, pp. 1365–1378, Dec 2008.
18. H. Durrant-Whyte and M. Stevens, "Data fusion in decentralized sensing networks," in *4th Intl. Conf. on Information Fusion*, 2001.

19. A. Makarenko and H. Durrant-whyte, “Decentralized data fusion and control in active sensor networks,” in *In Proceedings of the Seventh International Conference on Information Fusion*, 2004.

20. S. Moorehead, R. Simmons, and W. Whittaker, “Autonomous exploration using multiple sources of information,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001.

21. S. Grime and H. Durrant-Whyte, “Data fusion in decentralized sensor networks,” *Control Engineering Practice* **2**(5), pp. 849 – 863, 1994.

22. A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *Computer* **22**, pp. 46 –57, June 1989.

23. P. Stepan, M. Kulich, and L. Preucil, “Robust data fusion with occupancy grid,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **35**(1), pp. 106 –115, 2005.

24. A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, “The sensor-based random graph method for cooperative robot exploration,” *Mechatronics, IEEE/ASME Transactions on* **14**(2), pp. 163 –175, 2009.

25. A. Bahr, M. Walter, and J. Leonard, “Consistent cooperative localization,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 3415–3422, May 2009.

26. E. Nerurkar, S. Roumeliotis, and A. Martinelli, “Distributed maximum a posteriori estimation for multi-robot cooperative localization,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 1402–1409, May 2009.

27. S. Roumeliotis and G. Bekey, “Distributed multi-robot localization,” *IEEE Trans. Robot. Automat.* , August 2002. To Appear.

28. S. Williams, G. Dissanayake, and H. Durrant-Whyte, “Towards multi-vehicle simultaneous localisation and mapping,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2002.

29. D. Rodriguez-Losada, F. Matia, and A. Jimenez, “Local maps fusion for real time multirobot indoor simultaneous localization and mapping,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, **2**, pp. 1308–1313, 2004.

30. A. Howard, G. S. Sukhatme, and M. J. Matarić, “Multi-robot mapping using manifold representations,” *Proceedings of the IEEE - Special Issue on Multi-robot Systems* **94**, pp. 1360 – 1369, Jul 2006.

31. M. Pfingsthorn, B. Slamet, and A. Visser, “A scalable hybrid multi-robot SLAM method for highly detailed maps,” in *RoboCup 2007: Robot Soccer World Cup XI*, U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, eds., *Lecture Notes In AI*, pp. 457–464, Springer-Verlag, July 2008.