Multiple-Robot Simultaneous Localization and Mapping: A Review

Sajad Saeedi*
PhD, University of New Brunswick Fredericton, NB, Canada
e-mail: sajad.saeedi.g@unb.ca
Michael Trentini
PhD, Defence Research and Development Canada Suffield, AB, Canada
e-mail: Mike.Trentini@drdc-rddc.gc.ca
Mae Seto
PEng, PhD, Defence Research and Development Canada Halifax, NS, Canada
e-mail: mae.seto@drdc-rddc.gc.ca
Howard Li
PEng, PhD, IEEE Senior Member, University of New Brunswick Fredericton, NB, Canada
e-mail: howard@unb.ca

Received 6 October 2014; accepted 12 June 2015

Simultaneous localization and mapping (SLAM) in unknown GPS-denied environments is a major challenge for researchers in the field of mobile robotics. Many solutions for single-robot SLAM exist; however, moving to a platform of multiple robots adds many challenges to the existing problems. This paper reviews state-of-the-art multiple-robot systems, with a major focus on multiple-robot SLAM. Various issues and problems in multiple-robot SLAM are introduced, current solutions for these problems are reviewed, and their advantages and disadvantages are discussed.

1. INTRODUCTION

An autonomous robot needs to address two critical problems to survive and navigate within its surroundings: mapping the environment and finding its relative location within the map. Simultaneous localization and mapping (SLAM) is a process that aims to localize an autonomous mobile robot in a previously unexplored environment while constructing a consistent and incremental map of its environment. The interdependence of localization and mapping raises the complexity of the problem and necessitates accurately solving these two problems at the same time.

While single-robot SLAM is challenging enough, moving to a platform of multiple robots adds another layer of challenge. In a multiple-robot environment, robots must incorporate all available data to construct a consistent global map and localize themselves within the global map. Multiple-robot SLAM has benefits such as performing missions faster and being robust to failure of any one of the robots; however, these benefits come at the price of having a complex system that requires coordination and cooperation of the robots.

Various solutions have been proposed for SLAM (Whyte & Bailey, 2006). However, most of these solutions consider single-robot SLAM; few of them have considered multiple-robot SLAM. In this work, the most recent developments in multiple-robot SLAM are investigated. Potential problems in multiple-robot SLAM are listed and explained briefly. Then, the literature that addresses these problems is presented. This paper is motivated by the fact that new researchers have difficulties in appreciating all the various issues associated with multiple-robot SLAM. This paper provides a literature review on the state-of-the-art solutions and techniques. This work provides a complete literature survey of multiple-robot SLAM compared with the review provided in (Rone & Ben-Tzvi, 2013).

1.1. Applications of Multiple-robot SLAM

Multiple-robot SLAM is motivated by the fact that exploration and mapping tasks can be done faster and more accurately by multiple robots than by a single robot. In addition, in a distributed system, the whole team is more robust because the failure of one of the robots does not halt the entire mission (Birk & Carpin, 2006). Many collaboration-based operations need to be completed fast and autonomously

*COBRA Group at the University of New Brunswick, Fredericton, Canada, http://www.ece.unb.ca/COBRA/

View this article online at wileyonlinelibrary.com • DOI: 10.1002/rob.21620

and require localization and mapping. Some of these applications include

- fire fighting in forested and urban areas,
- rescue operations in natural disasters,
- cleaning operations like removing marine oil spills,
- underwater and space exploration,
- security and surveillance, and
- maintenance investigations.

Modern robots have numerous applications in industrial, military, and domestic settings. For mobile robots, such as cleaning robots, entertainment robots, and mine removal robots often deployed in large numbers, reliable perception is a key to achieving the desired tasks. Multiple-robot SLAM is a solution to the perception problem.

Often we hear news about miners trapped in mines or workers losing their lives in petroleum refineries or power plants. Using swarms of autonomous robots is an inexpensive alternative to having humans perform risky and hazardous tasks in such environments. Similarly, extraterrestrial applications such as space exploration are also highly risky for humans. By deploying robots to conduct dangerous tasks, risks can be minimized.

1.2. Outline

The rest of this review paper has been organized in eight sections. Section 2 presents a very brief introduction to SLAM. Section 3 explains the building blocks of SLAM algorithms. Section 4 introduces various SLAM algorithms for a single robot. Section 5 presents a background on multiple-robot SLAM and lists all known problems in this field. In Section 6, the available solutions for multiple-robot SLAM are reviewed. In Section 7, test beds and data sets for multiple-robot SLAM are presented. Section 8 outlines challenges and future directions. Lastly, in Section 9, conclusions are presented.

2. SIMULTANEOUS LOCALIZATION AND MAPPING: PROBLEM STATEMENT

Mobile robotic systems are developing very rapidly; however, real-world applications in GPS-denied environments require robust mapping and perception techniques to enable mobile systems to autonomously navigate complex environments. In this section, important concepts in relation to SLAM are explained briefly. For more information, see (Thrun, Burgard, & Fox, 2005).

Assume that the pose of a robot from time 1 to time t is shown by the sequence $\{x_1, x_2, ..., x_t\}$. This sequence is shown in the compact form of $x_{1:t}$. When multiple agents are involved, the identification number of each robot appears as a superscript in the state variable. Therefore, the following sequence shows the state of the i^{th} robot from time 1 to time t.

$$x^i_{1:t} \equiv \left\{x^i_1, x^i_2, ..., x^i_t\right\}, \tag{1}$$

where $i = 1, ..., n$ and n is the number of robots. Respectively, the observations made by the i^{th} robot and the control signals that drive the robot at the same times are shown as

$$z^i_{1:t} \equiv \left\{z^i_1, z^i_2, ..., z^i_t\right\},$$
$$u^i_{1:t} \equiv \left\{u^i_1, u^i_2, ..., u^i_t\right\}. \tag{2}$$

Note that for one robot the superscript is dropped.

For one robot, the goal for SLAM is to calculate the posterior over the map, m, and the trajectory given the action signals, measurement signals, and the initial pose of the robot:

$$p(m, x_{1:t}|z_{1:t}, u_{1:t}, x_0). \tag{3}$$

Equation (3) shows that estimating the map and trajectory of the robot is a coupled problem, which means both must be estimated at the same time. In the literature, this definition of SLAM is also referred to as *full SLAM*, where the whole trajectory is estimated, whereas *online SLAM* involves estimating the posterior over the current pose, x_t, and the map m:

$$p(m, x_t|z_{1:t}, u_{1:t}, x_0). \tag{4}$$

If the map of the environment is known, SLAM reduces to the *localization* problem. Localization seeks to calculate the posterior over the trajectory of the robot, given the map, the action signals, measurement signals, and possibly the initial pose of the robot:

$$p(x_{1:t}|m, z_{1:t}, u_{1:t}, x_0), \tag{5}$$

If the trajectory of the robot is known, SLAM becomes a *mapping* problem, which is the problem of estimating the map of the environment, given the pose of the robot and observations made by the robot from the environment:

$$p(m|z_{1:t}, x_{1:t}). \tag{6}$$

The probabilistic definition of SLAM in Eq. (3) can readily be extended to multiple robots. For simplicity, in the rest of the section, only two robots are considered, and the robots' identification numbers are represented by alphabetical characters.

For two robots, a and b, multiple-robot SLAM seeks to calculate the posterior over poses of the robots and the map:

$$p(x^a_{1:t}, x^b_{1:t}, m_t|z^a_{1:t}, z^b_{1:t}, u^a_{1:t}, u^b_{1:t}, x^a_0, x^b_0), \tag{7}$$

where the initial values of the poses are shown by x^a_0 and x^b_0, and m is the map of the environment. Figure 1 shows a simplified Bayes net for the two robots. In this figure, black lines show the transition of the state of each robot. Gray lines

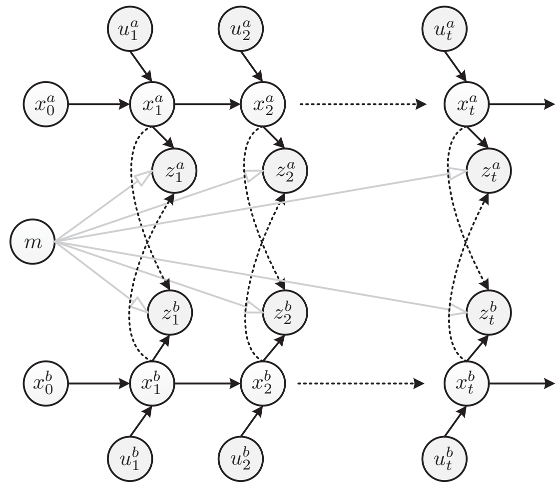


Figure 1. Bayes net for multiple-robot SLAM with two robots, a and b. Black lines correspond to the state transitions of each robot. Dashed lines correspond to the line-of-sight observations between the robots and gray lines show the relation of the map and observations.

show the relation of the map to observations. Dashed lines show the line-of-sight observations, by which the robots can see and detect each other. Line-of-sight observations will be explained later.

3. SIMULTANEOUS LOCALIZATION AND MAPPING: BUILDING BLOCKS

This section presents a brief review of the main building blocks of SLAM algorithms. The motivation of the section is to introduce the main elements of the various algorithms that make them unique. Any SLAM algorithm must deal with three main issues: 1) sensors, 2) data processing, and 3) map representation. As shown in Figure 2, ground, aerial, and underwater vehicles share many common sensors. Once a robot captures the data by suitable sensors, data processing may commence. Common data processing algorithms include filtering, smoothing, and artificial intelligence (AI). In data processing, different types of uncertainties, such as computational, algorithmic, and modeling uncertainties, are taken into account. Finally, in the map representation, a model of the environment along with the trajectory of the robot is generated. Data processing and map representation interact with one other to provide reliable results. In the following subsection, a brief overview of map representation is presented. For more information about filtering and smoothing algorithms, please refer to (Thrun et al., 2005).

Map representation is a fundamental issue with SLAM algorithms. Six types of maps commonly found in the literature include: 1) grid maps, 2) feature maps, 3) topological maps, 4) semantic maps, 5) appearance maps, and 6) hybrid maps.

Grid maps, also called *location maps*, represent the environment with a matrix of cells. Occupancy grid maps (Elfes, 1990) are the most popular maps, especially in two-dimensional (2D) SLAM algorithms. In occupancy grid maps (Elfes, 1990), each cell represents a small rectangular section of the world. The cell is modeled with a binary random variable that indicates the probability of existence of an object in the cell. Also, occupancy grid maps handle dynamic and moving objects of the environment efficiently (Thrun, 2003). Later, this method is explained in more detail. Occupancy grid maps can also be extended to three-dimensional (3D) modeling. Three-dimensional grid maps are known as volumetric pixel (voxel) maps.

Feature maps, also called *landmark maps*, represent the world with global position of features extracted from the environment. Often, the position of the feature is accompanied by the signature of the feature. The signature of a feature is the unique identifier that characterizes the feature. Various features, such as scale-invariant feature transform (SIFT) and speeded up robust features (SURF), have been introduced in (Fraundorfer & Scaramuzza, 2012). A 3D point-cloud map is an example of feature maps.

Topological maps represent an abstract model of the environment in the form of compact and connected paths and intersections (Kuipers & Byun, 1991). Humans and insects use topological maps to navigate, to determine paths and positions, and to avoid obstacles (Hawkins, 2004; Gould, 1990). As an example, pigeons have been shown to use highways and their intersections as a topological map to navigate and fly over long distances (Guilford, Roberts, Biro, & Rezek, 2004). Topological approaches for perception or autonomy can also be interpreted as symbolic mapping (Beeson, Modayil, & Kuipers, 2010). Symbolic approaches provide a concise representation for the structural alternatives, which can be used in path planning, place detection, and loop closure. Topological maps can be derived from metric maps. For instance, a Voronoi graph (Davies, 2005) is a topological map that can be computed from an occupancy grid map. A metric-free topological map, in the form of nerve complex (an abstract representation), is introduced in (Derenick, Speranzon, & Ghrist, 2013). Polygon maps are another example that assign planes to a set of features.

Semantic maps, which contain functionalities and relationships of the objects of the environment, are the most abstract type (Nuchter & Hertzberg, 2008). For instance, in an indoor environment, a semantic map may include a 3D point-cloud map in which specific objects such as humans, walls, doors, windows, and chairs are identified and labeled. Semantic maps are very useful for high-level and goal-oriented behaviors that require real-time reasoning about spatial and perspective ambiguities. Semantic maps are very similar to topological maps with the distinction that a semantic map includes more detailed information about objects and places.

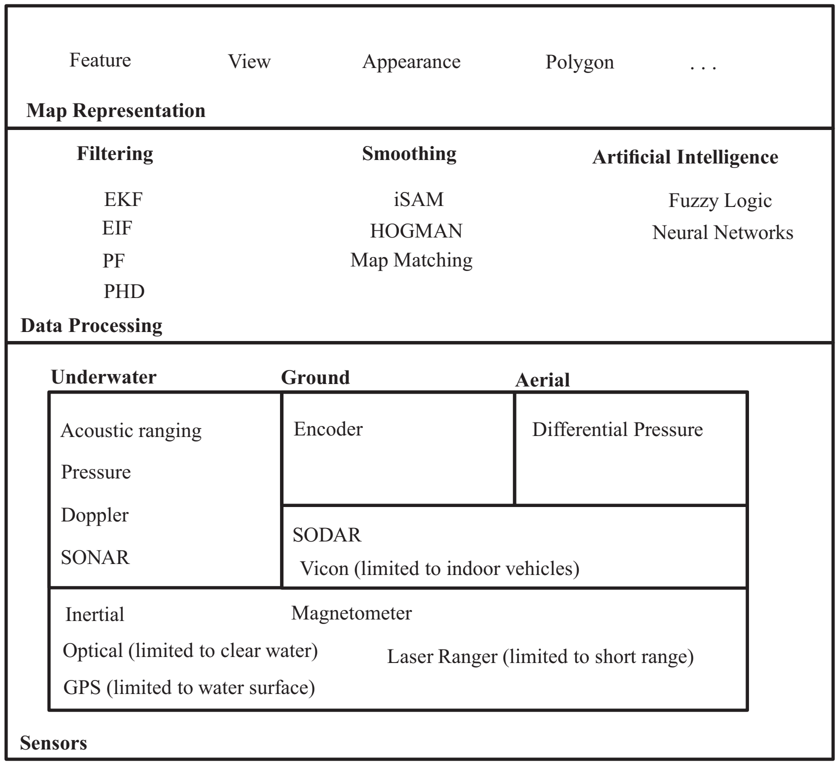


Figure 2. A SLAM algorithm must deal with three main issues: 1) sensors, 2) data processing, and 3) map representation. For each of these issues, various solutions exist, but the choice of the proper solution depends on the vehicle type, the environment, and the application of the SLAM algorithm.

Appearance maps are usually developed with a vision system and include different views associated with an undirected weighted graph. In other words, an appearance map is a graph in which each vertex is tied to an image. The edges of the graph connect sufficiently similar images to one another. The weight of each edge represents the similarity of the images at the two ends of the edge (Erinc & Carpin, 2014; Fraundorfer, Engels, & Nister, 2007). Figure 4 shows an appearance map with several edges connecting similar images to one another. Appearance maps are usually used when there is no wheel odometry available.

Hybrid maps are combinations of different mapping methods. For instance, a hybrid metric-topological map is a map that includes both topological and metric information. As an example, Tomatis et al. integrate topological and metric maps to perform hybrid simultaneous localization and mapping (Tomatis, Nourbakhsh, & Siegwart, 2003).

Figure 3 demonstrates a few sample maps, developed using laser ranger measurements: (left) occupancy grid map overlaid with Voronoi graph, (middle) semantic map where the hallways, rooms, doorways, and junctions are highlighted in different colors, (right) a hybrid metric-topological map, developed by the segmentation of the semantic map.

Table I summarizes the presented mapping methods with their advantages and disadvantages. For a review of applications of different maps in mobile robotics, see (Bonin-Font, Ortiz, & Oliver, 2008).

4. SINGLE-ROBOT SLAM: ALGORITHMS

This section presents a brief review of different algorithms in single-robot SLAM. The motivation of the section is to categorize available solutions to better understand multiple-robot SLAM. Various SLAM algorithms can be categorized based on the algorithm used for map representation and data processing. In this section, four popular SLAM algorithms, categorized based on the map representation method, are presented. Then, three main SLAM algorithms, categorized based on the data-processing algorithm, are briefly introduced. From the map representation perspective, the four reviewed algorithms are:

- feature-based SLAM,
- view-based SLAM,
- appearance-based SLAM, and
- polygon-based SLAM.

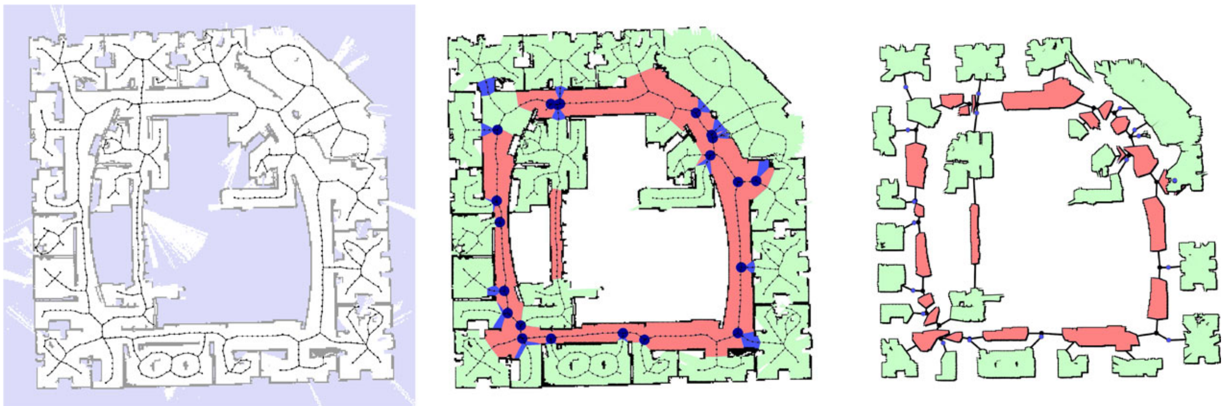


Figure 3. Examples of an occupancy grid map, a topological map, and a semantic map of Intel Labs: (left) an occupancy grid map overlaid with Voronoi graph; (middle) a semantic map in which different places in the map are labeled with different colors: hallways in red, rooms in green, doorways in light blue, and junctions in dark blue; (right) a hybrid topological-metric map. Image courtesy of Stephen Friedman, Hanna Pasula, and Dieter Fox (Friedman, Pasula, & Fox, 2007). Copyright © 2007, Association of Advancement of Artificial Intelligence.

Table I. Comparison of different mapping methods.

Type of Map	Description
Occupancy Grid Maps	Map is defined as a grid, each cell of the grid holds a probability for occupancy **Pros**: probabilistic, suitable for 2D mapping, suitable for dynamic environments **Cons**: expensive on fine resolution, expensive for 3D mapping
Feature Maps	Map is represented by features **Pros**: efficient for localization, scales well **Cons**: needs feature extraction, data association of features is difficult
Topological Maps	Map is represented by abstract spatial information **Pros**: well suited for high-level planning **Cons**: needs map processing, limited in waypoint following
Semantic Maps	Map is represented by semantic information **Pros**: well suited for high-level and goal-oriented reasoning **Cons**: needs training, object recognition, and classification
Appearance Maps	Map is represented by a weighted graph and multiple views **Pros**: very intuitive and useful for human and robot interaction **Cons**: requires high storage capacity to record all views
Hybrid Maps	Combination of different mapping methods **Pros**: suitable for loop closure, can handle map inconsistency **Cons**: needs map processing, requires coordination between maps

And, from the data-processing perspective, the three reviewed algorithms are:

- filtering SLAM,
- smoothing SLAM, and
- AI SLAM.

Topological SLAM and semantic-based SLAM are reviewed under AI SLAM. At the end of the section, 2D and 3D SLAM algorithms are discussed.

4.1. Feature-based SLAM

Feature-based SLAM was the first solution proposed for SLAM. This solution extracts features or landmarks in the environment and keeps a list of them as a map (Smith, Self, & Cheeseman, 1987; Guivant & Nebot, 2001). This method requires feature extraction and is therefore limited to environments with features. Most feature-based solutions use vision sensors (Davison & Murray, 2002), underwater sonar (Jaulin, 2009; Newman, Leonard, & Rikoski, 2005), or occasionally LIDAR (Bailey, 2000). In feature-based SLAM, features from consecutive frames are extracted and cross-matched to calculate the relative motion of the sensor. This process is referred to as visual odometry and is used in almost all feature-based algorithms. A thorough review of the visual odometry algorithms is presented in (Scaramuzza & Fraundorfer, 2011; Fraundorfer & Scaramuzza, 2012).

Although the feature-based paradigm has been shown to be efficient and effective in certain environments, in general it is not. By extracting features, there is possible loss of data. Thrun et al. state that "a lot of information is sacrificed by using features instead of full measurement vector. This lost information makes solving certain problems more difficult, such as the data association problem of determining

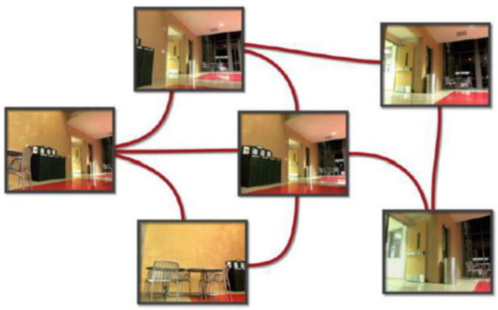


Figure 4. An appearance map is composed of a graph, where each vertex of the graph represents a specific view of the environment. Edges of the graph connect similar views to one another. Image courtesy of Gorkem Erinc and Stefano Carpin (Erinc & Carpin, 2014).

whether or not the robot just revisited a previously explored location" (Thrun et al., 2005, p. 181). Since onboard processing has become so powerful, there is no longer a need to extract features from sensor data and potentially lose useful information in the process. This applies especially to dense range sensors such as laser rangers.

4.2. View-based SLAM

View-based SLAM, also called location-based SLAM, is based on raw sensor data-processing techniques. View-based SLAM usually requires a range/bearing sensor, such as scanning laser ranger (Grisetti, Stachniss, & Burgard, 2007; Hahnel, Burgard, Fox, & Thrun, 2003a). The process of calculating the relative motion of the laser ranger from the consecutive scans is referred to as scan matching, or laser odometry (Lu & Milios, 1994; Nieto, Bailey, & Nebot, 2007; Censi, 2008). In general, scan matching algorithms are either based on the different variant of the iterative closest point (ICP) algorithm, for example (Segal, Haehnel, & Thrun, 2009), or matching a scan with the most recent map of the environment, for instance (Kohlbrecher, Stryk, Meyer, & Klingauf, 2011).

View-based SLAM differs from feature-based SLAM in that no features are explicitly extracted (Hahnel et al., 2003a). Instead, entire scans are matched using scan matching algorithms (see Figure 5), and the map is represented in occupancy grid format (Thrun, 2003; Elfes, 1990). The state vector in view-based SLAM includes either the current pose of the robot or the whole trajectory. A map also accompanies the trajectory. The most important advantage of view-based SLAM over feature-based SLAM is that no information is lost because no feature is extracted. In addition, view-based SLAM contains information not only about the occupied spaces and the objects in the environment but also about the object-free spaces; therefore, view-based SLAM can handle dynamic environments much better than feature-based SLAM (Thrun et al., 2005, p. 152).

4.3. Appearance-based SLAM

Appearance-based SLAM is an effective method for the loop closure problem (Koenig, Kessler, & Gross, 2008; Cummins & Newman, 2011). It can also be used in combination with feature-based or view-based SLAM; however, in the literature, appearance-based SLAM is often performed with a camera (Ulrich & Nourbakhsh, 2000). In this method, new observations, either features or views, are matched against available reference observations to identify a previously observed location.

For instance, the ATLAS framework by Bosse et al. provides occupancy grid maps from outdoor urban environments using a scanning laser ranger mounted on top of a car (Bosse & Zlot, 2008). In this method, salient characteristics of maps are used to develop an appearance recognition

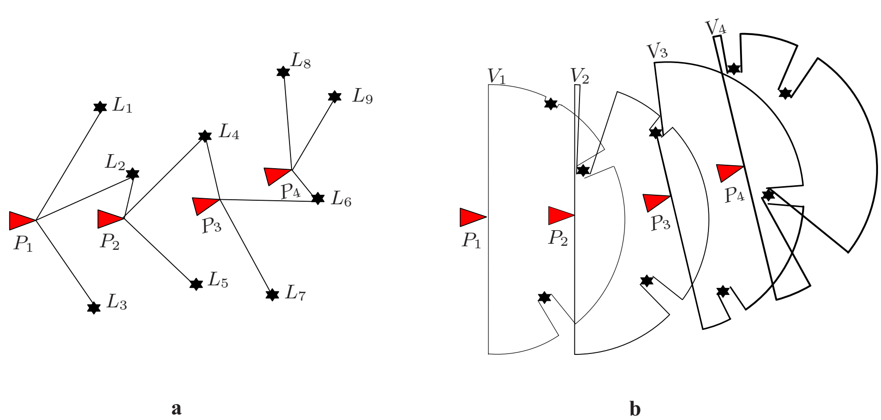


Figure 5. P_1 to P_4 are poses of a robot at four time instances. (a) Feature-based SLAM. L_1 to L_9 are detected features used for localization. (b) View-based SLAM. V_1 to V_4 are corresponding views. No features are extracted; instead, views are matched for localization (Paull, Saeedi, Seto, & Li, 2014).

system, which recalls an already visited location. In (Kawewong, Tongprasit, Tangruamsub, & Hasegawa, 2010), position-invariant robust feature (PIRF) detection is used to detect features of images. These features are matched against reference features and used to recall previous scenes. Generally, due to the necessity of matching a large number of views with reference views, appearance-based SLAM creates high computational demand and often requires a fast processor and relatively large memory to store all views.

4.4. Polygon-based SLAM

Polygon-based SLAM is used for 3D mapping (Hahnel, Burgard, & Thrun, 2003b). In this method, planar segments, composed of infinite planes, are associated with features of the environment. For example, Weingarten et al. perform 3D SLAM with a robot equipped with a tilting scanning laser ranger (Weingarten & Siegwart, 2006). Using polygon sets, features are represented with planar segments. Pathak et al. propose a 3D GraphSLAM (see Section 4.6) by registering large planar segments, extracted from a 3D laser ranger (Pathak et al., 2010). Expectation maximization has also been used for planar mapping (Thrun et al., 2004b). As explained in relation to the topological and semantic mapping, the advantage of the polygon-based SLAM is that it produces highly detailed small-sized maps and is well-suited for higher-level tasks such as interacting with the environment; however, the process of fitting planar segments creates extra computational demand.

4.5. Filtering SLAM

Particle filters, different variations of the Kalman filter, and the information filter are major filtering methods derived from the Bayesian filter. These techniques are used to estimate the current pose of the robot or to optimize the whole trajectory of the robot. For a detailed explanation of various filtering algorithms and their applications in SLAM, see (Thrun et al., 2005, Ch. 3-4). Popular and well-studied SLAM algorithms based on filtering include extended Kalman filter SLAM (EKF-SLAM), extended information filter SLAM (EIF-SLAM), and particle filter SLAM (PF-SLAM). In recent years, many improvements have been made to these algorithms. For instance, the distributed particle (DP) filter, proposed by Eliazar et al., is a fast and reliable implementation of particle filtering for SLAM (Eliazar & Parr, 2003), which optimizes memory requirements using an efficient data structure for maintaining grid maps. Other examples including the improved grid mapping (Grisetti et al., 2007) and FastSLAM (Montemerlo, Thrun, Koller, & Wegbreit, 2002) reduce the complexity of the particle filter by reducing the number of particles.

The EKF-SLAM , EIF-SLAM, FastSLAM, and GraphSLAM (Section 4.6) algorithms are considered as vector-based solutions, in which the map and the pose of the robot are estimated by a random vector. Typically in vector-based, feature-based SLAM algorithms, state estimation and data association (determining the correspondence between measurements and landmarks) are considered as two separate problems. The data association problem is usually solved using heuristics or advanced filtering methods, such as the joint probabilistic data association (Dezert & Bar-Shalom, 1993). In addition, determining the number of the features is also considered as a separate problem.

Set-based algorithms are emerging solutions based on the concept of the random finite set (RFS) (Mullane, Vo, Adams, & Vo, 2013) and finite set statistics (FISST) (Mahler, 2007). In a set-based SLAM algorithm, key problems, such as the number of the features and data association, are formulated and incorporated to the Bayesian filter; thus, there is no need to handle these problems outside the filtering algorithm separately. Set-based algorithms have better efficiency under various measurement issues, including inaccurate measurements (noise), false detections (clutter), and failure to detect targets (missed detections).

Similar to the vector-based SLAM algorithms, there are various implementations for set-based SLAM. One approach is based on the first-order moment of an RFS, known as the probability hypothesis density (PHD), or intensity function. Intuitively, the PHD is the expected value of an RFS. Peaks of the PHD correspond to likely locations of the features, and the integral of the PHD determines the expected number of the features. For example, Rao–Blackwellized (RB) PHD-SLAM is proposed and tested in (Mullane, Vo, Adams, & Vo, 2011; Adams, Vo, Mahler, & Mullane, 2014; Leung, Inostroza, & Adams, 2014). The results show improvement over vector-based SLAM algorithms, such as EKF-SLAM and FastSLAM.

4.6. Smoothing SLAM

Smoothing methods estimate the whole trajectory of the robot by minimizing the process and observation constraints. Two general smoothing methods in SLAM are GraphSLAM and submap matching. In general, in GraphSLAM, poses of a robot are represented as nodes in a graph. The edges that connect nodes are modeled with motion and observation constraints. Next, these constraints need to be optimized to calculate the spatial distribution of the nodes and their uncertainties (Thrun et al., 2005, Ch. 11).

GraphSLAM was first formulated by Lu and Milios (Lu & Milios, 1997). They optimized a system of equations to decrease the error caused by constraints. Since then, researchers have proposed different solutions for GraphSLAM, including (Gutmann & Konolige, 1999), relaxation on a mesh (Howard, Mataric, & Sukhatme, 2001), multilevel relaxation (Frese, Larsson, & Duckett, 2005), iterative alignment (Olson, Leonard, & Teller, 2006), square root simultaneous location and mapping (SAM) (Frank

& Michael, 2006), incremental smoothing and mapping (iSAM) (Kaess, Ranganathan, & Dellaert, 2007), the finite element method (FEM) (Takeuchi & Tsubouchi, 2008), g^2o (Kummerle, Grisetti, Strasdat, Konolige, & Burgard, 2011), and finally three works by G. Grisetti et al. in (Grisetti, Stachniss, & Burgard, 2009; Grisetti, Kümmerle, Stachniss, & Burgard, 2010b), and hierarchical optimization for pose graphs on manifolds (HOGMAN) (Grisetti, Kümmerle, Stachniss, Frese, & Hertzberg, 2010c).

Submap matching is also related to GraphSLAM. Submap matching is an efficient method of mapping in which small local maps are matched with one another and mosaicked to produce a global map. This prevents global error accumulation due to the odometry and measurement inaccuracies and therefore is a good solution for large-scale and outdoor environments. Bailey's work (Bailey, 2000), hierarchical SLAM (Estrada, Neira, & Tardos, 2005), the Atlas framework (Bosse & Zlot, 2008), and Tectonic SAM (Ni, Steedly, & Dellaert, 2007) are examples of such a solution. In the Atlas framework (Bosse & Zlot, 2008), a global map management solution is proposed that can close loops in structured environments. Atlas provides view-based local maps using an EKF and attaches them together by comparing the normals of the maps. In Bailey's work (Bailey, 2000), a similar solution is proposed, which is developed using a 2D feature-based SLAM. A similar approach is proposed in Tectonic SAM (Ni et al., 2007) in which optimization is performed to align local maps. Local maps in this method are generated using square root information smoothing (Frank & Michael, 2006). Submap matching is very similar to an approach of multiple-robot SLAM, which is called map merging (Birk & Carpin, 2006; Saeedi, Paull, Trentini, Seto, & Li, 2014a,b), introduced in Section 6.8, with one notable difference. In the case of map merging, we assume that there is no *a priori* knowledge of the transformation that relates the maps. In single-robot submap matching, an initial estimate of the transformation is available, so it is feasible to perform an exhaustive search for the best match within the neighborhood of the estimate.

4.7. Artificial Intelligence SLAM

Artificial intelligence has also been used to solve the SLAM problem. In these solutions, filtering, or smoothing, is realized using AI algorithms. For instance, in (Chatterjee, 2009), a solution based on fuzzy logic is proposed to tune covariance values of the measurement model. The ratSLAM algorithm is a technique that models a rodent's brain using neural networks (Wyeth & Milford, 2009). In fact, this method is a neural network-based data fusion using a camera and an odometer. In (Saeedi, Paull, Trentini, & Li, 2011b), self-organizing map (SOM), or Kohonen network, is used to perform SLAM with multiple robots. The SOM is a competitive neural network which is trained without supervision.

Abstract geometrical perception is a foundation for high-level reasoning. The abstract information facilitates faster information processing. Semantic approaches to SLAM, such as SLAM++ (Salas-Moreno, Newcombe, Strasdat, Kelly, & Davison, 2013), and topological methods, which are based on the abstract information, are also another form of the AI SLAM, where the abstract representation is used to perform localization and mapping. A common example of topological representation is the generalized Voronoi diagram (GVD) (Bunke, 2000), which represents the skeleton of a map.

There are many topological solutions for single-robot SLAM such as the works by Choset et al. (Choset & Nagatani, 2001; Choset, Walker, Eiamsa-Ard, & Burdick, 2000), annotated generalize Voronoi graph (AGVG) (Wallgrum, 2010), the works in (Beeson, Jong, & Kuipers, 2004) and (Beeson et al., 2010), underground mines (Silver, Ferguson, Moms, & Thayer, 2004), Bayesian inference (Ranganathan, Menegatti, & Dellaert, 2006), and the semantic approach with place labeling (Friedman et al., 2007). Boal et al. present a thorough and recent survey of topological approaches for SLAM in (Boal, Sanchez-Miralles, & Arranz, 2014).

The presented methods in Section 4.5, 4.6, and 4.7 are summarized in Table II. For each method, a few references with various types of maps are presented. Main advantages and disadvantages of each method are also briefly listed.

4.8. 2D and 3D SLAM

Robotic environments are in general 3D, involving translation in three directions, x, y, and z, and rotation around three axes, roll, pitch, and yaw. However, in most cases, especially indoors, the robot has no change in the z direction and has negligible roll and pitch angles. In these cases, the SLAM problem can be simplified to estimating only x, y, and the yaw angle; this is referred to as 2D SLAM. However, in outdoor environments and when the robot demonstrates changes in the z direction or has considerable rotation for roll and pitch angles, 3D SLAM is required for pose estimation.

The complexity of 3D SLAM is not as simple as estimating three more parameters. This is mainly because most sensory information lacks full 3D perception, and therefore, it becomes challenging to estimate parameters that are not observable directly. Moreover, if sensors exist to provide full 3D information, the complexity of processing algorithms increases significantly. For instance, solving a nonlinear equation for n variable requires an $n \times n$ matrix inversion, which, by the fastest implementation, is of the order $\mathcal{O}(n^{2.373})$ (Virginia Vassilevska Williams, 2012).

Stereo or RGBD cameras, such as Kinect or Asus Xtion, and 3D laser rangers, such as Velodyne laser scanners, are among the well-known sensors used for 3D SLAM. The

Table II. Comparison of some common SLAM techniques.

	SLAM Method	Map	Description
Filtering	EKF-SLAM		• based on the extended Kalman filter
	(Smith et al., 1987)	feature	**Pros**: works well if features are distinct
	(Eustice, Singh, & Leonard, 2006)	view	**Cons**: adding a new feature to state space needs quadratic time, requires feature extraction in feature-based SLAM, cannot identify absence of a feature
	(Newman, Cole, & Ho, 2006)	appearance	
	(Weingarten & Siegwart, 2006)	polygon	
	EIF-SLAM		• based on the extended information filter
	(Thrun et al., 2004a)	feature	**Pros**: measurement updates are performed in constant time, effective for multiple-robot SLAM due to additivity of information
	(Hahnel, Burgard, Wegbreit, & Thrun, 2003c)	view	**Cons**: information matrix needs to be sparsified. Recovering map and the pose requires large matrix inversion
	PF-SLAM		• based on particle filtering. FastSLAM is an efficient implementation of particle filtering
	(Montemerlo, Thrun, Koller, & Wegbreit, 2003)	feature	**Pros**: effective for loop closure, performs full and online SLAM, logarithmic complexity in number of features, no need for parametrization
	(Grisetti et al., 2007)	view	**Cons**: quality of the estimation is dependent on the number of particles
	DP-SLAM		• a fast implementation of particle filtering for SLAM
	(Eliazar & Parr, 2003)	view	**Pros**: effective for large-scale maps, optimizes memory requirements **Cons**: requires processing to recover maps
	Set-based SLAM		• based on RFS and finite set statistics (FISST)
	(Adams et al., 2014)	feature	**Pros**: number of features and data association are estimated with the Bayesian filter
	(Lee, Clark, & Salvi, 2012)	feature	
	(Mullane et al., 2011)	feature	**Cons**: higher time complexity than vector-based solutions
Smoothing	GraphSLAM		• uses smoothing techniques to estimate the trajectory and the map
	(Kaess et al., 2007)	feature	**Pros**: the whole trajectory is updated
	(Grisetti et al., 2010c)	view	**Cons**: computationally demanding, hard to recover covariances
	(Pathak et al., 2010)	polygon	
	Submap Matching		• matches small local maps to make a large global map
	(Bosse & Zlot, 2008)	view	**Pros**: the whole trajectory is updated, suitable for large-scale environments
	(Ni et al., 2007)	feature	**Cons**: size of local maps should be adjusted
AI	AI SLAM		• based on artificial intelligence
	(Wyeth & Milford, 2009)	appearance	**Pros**: efficient, usually no mathematical model is required
	(Chatterjee, 2009)	feature	**Cons**: error-prone, requires training or parameter tuning, training can be time consuming
	(Choset & Nagatani, 2001)	feature	

core of a 3D SLAM algorithm is visual or laser odometry. The odometry algorithm should run in real time on robotic platforms. For RGBD and stereo cameras, KinectFusion (Newcombe et al., 2011), ScaViSLAM (Strasdat, Davison, Montiel, & Konolige, 2011), fast odometry (Dryanovski, Valenti, & Xiao, 2013), fast odometry from vision (FOVIS) (Huang et al., 2011), and dense tracking and mapping (Sturm, Bylow, Kahl, & Cremers, 2013) are examples of the 3D visual odometry and SLAM algorithms. For laser rangers, generalized ICP is used in a 3D scan matching algorithm for 3D laser odometry (Segal et al., 2009). The structure of the filtering, or smoothing, algorithms for 2D and 3D SLAM is very similar. For instance, in (Henry, Krainin, Herbst, Ren, & Fox, 2010) and (Engelhard, Endres, Hess, Sturm, & Burgard, 2011), a method called rgbdSLAM is proposed to perform 3D SLAM using stereo and Kinect cameras. Their visual odometry is based on the ICP algorithm, and the optimization over the poses is performed using HOGMAN GraphSLAM.

5. BACKGROUND ON MULTIPLE-ROBOT SLAM

This section first presents a brief description of multirobot systems and multiple-robot SLAM. Generally, multiple-robot SLAM algorithms are built on top of single-robot SLAM algorithms; therefore, data processing and map representation are based on the approaches reviewed earlier. However, in multiple-robot SLAM, additional challenging problems are due to the increased number of robots. At the end of this section, 10 major problems in multiple-robot SLAM, together with some solutions for each, are introduced.

5.1. Multiple-Robot Systems

In addition to multiple-robot SLAM, other aspects of multiple-robot systems, such as task allocation, formation, and coordination, are also interesting research topics. For a list of major areas of multiple-robot systems, see (Jennings, Sycara, & Wooldridge, 1998; Arai, Pagello, & Parker, 2002). One of the first initiatives on multiple-robot systems was the Autonomous Vehicle Aerial Tracking and Reconnaissance (AVATAR) project, developed by Defense Advanced Research Projects Agency (DARPA). This project was targeting military surveillance and demonstrated the effectiveness of a multiple-robot system in cooperative localization and surveillance (Sukhatme, Montgomery, & Vaughan, 2001). In the Perception of Offroad Robotics (PerceptOR) project, DAPRPA focused on cooperation between an autonomous helicopter and an autonomous ground vehicle. The autonomous helicopter was helping the ground robot to navigate autonomously in an unknown environment (Kelly et al., 2006). The Mobile Autonomous Robot Software (MARS) program is another multiple-robot framework by DARPA, which addresses issues such as cooperative target localization and surveillance, search and rescue, and maintaining connectivity during cooperative mapping (Chaimowicz, Grocholsky, Keller, Kumar, & Taylor, 2004; Grocholsky, Bayraktar, Kumar, Taylor, & Pappas, 2004). The Multi Autonomous Ground-robotic International Challenge (MAGIC) is a multirobot competition, in which the robotics teams were requested to explore and map large indoor and outdoor environments while identifying and neutralizing threats. MAGIC was funded by the Australian Department of Defence and the U.S. Army, and the robotic teams were pursuing more than $1 million in prize money (Olson et al., 2013).

Barca et al. present a thorough review on swarm robotics (Barca & Sekercioglu, 2013). Ren et al. (Ren, Beard, & Atkins, 2007, 2005) and Olfati-Saber et al. (Olfati-Saber, Fax, & Murray, 2007) review the consensus problems in the coordination of multiagent systems. Murray reviews cooperative control (Murray, 2007). Bullo has mathematically presented the motion coordination problem (Bullo, Cortés, & Martínez, 2009). Olfati-Saber has presented and reviewed three flocking algorithms (Olfati-Saber, 2006). Multiple-robot exploration (Rekleitis, Dudek, & Milios, 2001; Zlot, Stentz, Dias, & Thayer, 2002; Burgard & Schneider, 2002; Sheng, Yang, Tan, & Xi, 2006; Burgard, Moors, Stachniss, & Schneider, 2005; Wurm, Stachniss, & Burgard, 2008) is also another interesting topic tightly coupled with multiple-robot SLAM. A review of the exploration methods can be found in (Rone & Ben-Tzvi, 2013). Multiple-robot localization, a simpler form of multiple-robot SLAM, has been presented in (Roumeliotis & Bekey, 2002; Hao & Nashashibi, 2013; Fox, Burgard, Kruppa, & Thrun, 2000; Schneider & Wildermuth, 2012; Zhou & Roumeliotis, 2008) and in (Paull et al., 2014) for underwater environments.

5.2. Data for Multiple-Robot SLAM

In a multiagent system, there are a few key issues that need to be taken into account. The way these issues are handled makes each solution unique. These issues are phrased in the form of the following four questions. First, what type of data is shared among agents? Second, how is this data shared among agents? Third, where is this data processed? Finally, how is the processing performed? In the next few subsections, each of these questions is answered in the context of multiple-robot SLAM. Figure 6 shows these different issues in multiple-robot SLAM.

5.2.1. Data Communication

Availability of a system to share data among robots is a key requirement in multiple-robot SLAM (Leung, Barfoot, & HT Liu, 2012; Bar-Shalom, 2002). Information between robots can be exchanged via communication channels that might not be available at all times, in all places. Bandwidth and coverage of the communication network are two important factors in the performance of SLAM.

5.2.2. Data Sharing

Sharing data among robots is a fundamental issue in multiple-robot SLAM. Past approaches to collaborative SLAM can generally be categorized based on whether they share raw sensor data (Howard, 2006) or processed data (Birk & Carpin, 2006). Raw sensor data means that sensed information, such as laser ranger measurements and wheel odometry readings, are not processed. In processed data, such as a map or poses of robots, sensor readings are processed through filtering, or smoothing, or other methods. Sharing raw sensor data results in more flexibility but requires high bandwidth and reliable communication between robots as well as more processing power. In contrast, sharing maps use less bandwidth, and there is no need to

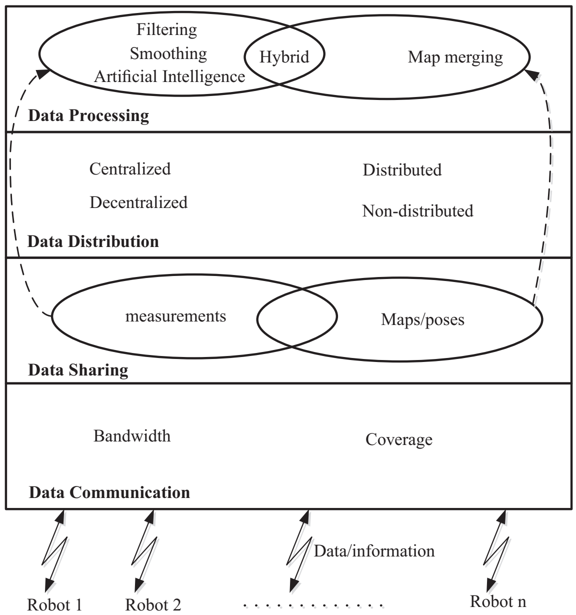


Figure 6. Four major issues in multiple-robot SLAM are data communication, data sharing, data distribution, and data processing. Choosing the right solutions for these issues depends on many factors, such as the robots, environment, and applications.

process raw data redundantly; however, the performance is dependent on the quality of maps. The latter method is usually referred to as map merging or map fusion. The work by S. Thrun (Thrun, 2001) or Howard's work (Howard, 2006) are examples of sharing raw laser and odometry data among robots. Works by Carpin et al. (Carpin, Birk, & Jucikas, 2005) and Birk et al. (Birk & Carpin, 2006) are examples of sharing processed maps between robots. The choice of the method depends on several factors such as the application and the available resources.

5.2.3. Data Distribution

In general, multiple-robot SLAM can be centralized or decentralized, distributed or nondistributed (Leung, 2012), as follows:

- **Centralized**: In a centralized system, the computation for a task is performed by a predetermined robot on the team or by an external agent. The central agent processes the incoming data and provides required information and feedback to other agents. The work in (Kim et al., 2010) is an example of this solution.
- **Decentralized**: In a decentralized system, the computation for a task is performed by more than one robot on the team. Obviously, this structure requires that the robots have enough computational power to respond to processing demands. The work in (Howard, 2006) is an example of this solution.
- **Distributed**: In a distributed system, the computation for a task is divided among the robots on the team.

By these definitions, a centralized or decentralized system can be distributed or nondistributed. For example, a system might be distributed where a task is performed with collaboration of a few robots, but a central external agent integrates all information to generate the final result. The collaborative mapping with multiple microaerial vehicles is an example of a distributed and centralized system (Forster, Lynen, Kneip, & Scaramuzza, 2013). Moreover, hybrid solutions are also applicable, where some robots process data and some use the results only. The work in (Howard, Parker, & Sukhatme, 2006a) is an example of such a solution in which 80 robots were cooperating at different levels to make a global map.

5.2.4. Data Processing

The choice of the method for estimating the poses of robots and the global map depends on many factors, such as the available memory, processing capability, and the type of the sensory information. In genernal, the spectrum of SLAM algorithms includes many different approaches each with remarkable capabilities. At the two extreme ends of the spectrum lie two methods: EKF-SLAM and GraphSLAM (Thrun et al., 2005, p. 385). EKF-SLAM approximates the nonlinearity of the system by linearizing the system model. Also, EKF-SLAM is computationally expensive. Each time acquired information is resolved into a probability distribution that makes it proactive but slow. GraphSLAM solutions, such as iSAM (Kaess et al., 2007) and HOGMAN (Grisetti, Kümmerle, Stachniss, Frese, & Hertzberg, 2010a), also depend on approximation by Taylor expansion. Its difference with EKF-SLAM is that it accumulates information and therefore is considered to be a lazy and an offline algorithm (Thrun et al., 2005, p. 385). The particle filtering approaches are nonlinear filtering solutions; therefore, the system models are not linearized. Although particle filtering is computationally expensive, it is shown that a good implementation, such as FastSLAM (Thrun et al., 2005, Ch. 13) and grid mapping (Grisetti et al., 2007), can reduce the complexity of the algorithm.

5.3. Problems in Multiple-Robot SLAM

In this section, problems for multiple-robot SLAM are listed and explained briefly. The list includes 10 major problems, and a summary of the problems, with a few solutions for each problem, presented at the end of the section.

5.3.1. Relative Poses of Robots

In multiple-robot SLAM, the map provided by each robot in its own reference coordinates is called the local map. Each local map is generated from coordinated measurements such as laser scans. Each robot tries to integrate all of the local maps provided by the other robots to generate a global map of the environment. However, this is a difficult task because the required alignments or transformation matrices, which relate these maps to each other, are in general unknown. The problem of the relative pose of the robot is coupled with the multiple-robot data association problem. Knowledge of one renders the other one as a simple problem.

5.3.2. Uncertainty of the Relative Poses

Uncertainty, which according to Thrun et al. arises if the robot lacks critical information for carrying out its task, is identified as having five main roots (Thrun et al., 2005, p. 3): environment, sensors, robots, models, and computations. There is an uncertainty associated with the relative transformation matrix represented in the form of a covariance matrix. This uncertainty is mainly because of modeling uncertainties, sensory noises and linearization effects. Updating the maps and poses should be performed using the covariance matrix. Depending on the method used for finding the relative transformation matrix, covariance matrix takes different values.

5.3.3. Updating Maps and Poses

Once the relative transformation is found, a procedure is required to fuse local maps. The resulting map should integrate all information from given local maps. As a result of updating the maps, poses of the robots should also be updated. This requires considering the trajectory of the robots and new information received from other maps. Because of the nature of multiple-robot SLAM, updating poses and maps is a coupled problem. In feature-based SLAM, data association and finding the correspondence between duplicate landmarks across the robots is an important part of updating maps.

5.3.4. Line-of-sight Observations

On some occasions, robots can see and detect one another. Each robot might already have an estimate of the pose of other robots. However, when robots can see one another through line-of-sight and direct observations, these estimates can be improved. This fact can help robots to reduce mapping and localization error. In most applications and especially in close-range localization, line-of-sight observations are much more reliable than other indirect estimation techniques. Figure 7 shows an application in which the line-of-sight observation is its key component. In this application, a rotary wing UAV lands on a moving platform autonomously. The UAV is equipped with a camera that identifies a known target pattern on the moving platform. Then, the distance and orientation of the UAV with respect to the moving platform is calculated. The results are used to control and land the UAV (Richardson et al., 2013).

Figure 7. Using line-of-sight observations, a rotary wing UAV lands on a moving platform autonomously. Image courtesy of Thomas Richardson et al. (Richardson et al., 2013).

5.3.5. Closing Loops

Loop closure, also called cycle detection, is defined as identifying a place observed previously but not very recently (recently is defined in relation to the mission duration). Loop closure for a single robot is challenging enough. Extending this problem for a team of multiple robots requires solving it using all resources of information from individual robots. In multirobot SLAM, various events can trigger loop closure, such as direct encounter of the robots or rendezvous and indirect encounter, when the robots see the same area or features in the world.

5.3.6. Complexity

Robotics applications are usually real time. Thus, it is very important to design an algorithm capable of solving the above-mentioned problems with minimum time and memory requirements. In multiple-robot SLAM, space complexity and time complexity are two important issues. The complexity of a multirobot algorithm directly affects its scalability.

5.3.7. Communications

Availability of a medium for data sharing among robots is an important requirement in multiple-robot SLAM. Information between robots can be exchanged via communication channels. The quality of the communication channels is dependent on the environment. For instance, communication issues are a challenging problem for a team of robots in underwater environments, where the environment imposes limitations on the bandwidth and data rate.

Figure 8. Heterogeneous robots used to map an earthquake-damaged building: (left) Kenaf ground robot, (middle) Quince ground robot and a Pelican quadrotor. Kenaf and Pelican are equipped with scanning laser rangers. Quince transports Pelican from one site to another. (right) A sample 3D map of three floors, developed using scanning laser rangers of Kenaf and Pelican. Image courtesy of Nathan Michael et al. (Michael et al., 2012).

5.3.8. Heterogeneous Vehicles and Sensors

An important advantage of team-based mapping is that different types of robots, equipped with different sensors, can provide a better model of the environment. For instance, a ground robot may see features that a quadrotor cannot, and at the same time, a quadrotor may have access to different areas that a ground robot does not. However, this advantage requires processing and integrating different types of information. For example, if a ground robot provides an occupancy grid map and a quadrotor generates a feature map, then these maps must be combined to generate a global and consistent map. (This issue has been studied in (Wurm, Stachniss, & Grisetti, 2010)). Because of the variety of sensors and maps, this problem can be presented in many different forms, such as integrating topological maps with grid maps, integrating topological maps with feature maps, integrating scanning laser range-finder measurements with camera measurements, integrating satellite and aerial views with ground views (Hussein, Renner, Watanabe, & Iagnemma, 2013), and many more. Michael et al. present a very good example of a team of heterogenous robots (see Figure 8), including a quadrotor and two ground robots, which map a multifloor earthquake-damaged building collaboratively (Michael et al., 2012).

5.3.9. Synchronization

As a general rule, each acquired sensor reading should have a time stamp field, which shows the time of the acquisition of the data. An important issue in a system of multiple agents and multiple sensors is the synchronization of the clocks. Synchronization can be considered at two levels: first, local synchronization, which means the sensors of each robot should be synchronized and second, global synchronization, which means all robots on the team must have synchronized clocks.

To synchronize time on different robots, Chrony (Curnow, 2014) is a suitable choice, which is also used in robot operating system (ROS) middleware. Chrony supports online and offline time adjustment by two different applications. In the online case, a network time protocol (NTP) daemon runs in the background and synchronizes the time with time servers. For an isolated machine, one might enter the time periodically. The synchronized time appears as a label in the header of each acquired data. A similar approach is used by Leung et al. in (Leung, Halpern, Barfoot, & HT Liu, 2011).

5.3.10. Performance Measure

In multiple-robot SLAM, evaluating the accuracy of results is a challenging problem due to the lack of the model of the environment and the actual trajectory of the robots. Additionally, evaluating the accuracy of SLAM becomes more critical when the robots rely on the SLAM to perform autonomous behaviors. Therefore, performance measure is always required to determine the reliability of multiple-robot SLAM.

Table III summarizes the reviewed problems in multiple-robot SLAM with a short description for each one and the references addressing the problems. These references are explained in detail in the next section.

6. CURRENT SOLUTIONS FOR MULTIPLE-ROBOT SLAM

In this section, filtering, smoothing, and other solutions in the literature for multiple-robot SLAM are reviewed. In each solution, it is mentioned that which problems in multiple-robot SLAM have been addressed.

6.1. EKF-SLAM

For multiple-robot SLAM, various solutions based on the EKF have been proposed, such as the cooperative EKF estimator (Fenwick, Newman, & Leonard, 2002), outdoor elevation mapping (Madhavan, Fregene, & Parker, 2004),

Table III. Problems in multiple-robot SLAM.

No	Problem	Description	Solutions
a	Relative Poses of Robots	Initial poses or transformation between poses	Rendezvous: (Howard, 2006), (Zhou & Roumeliotis, 2006) Map merging: (Birk & Carpin, 2006), (Saeedi, Paull, Trentini, Seto, & Li, 2012b) Relative localization: (Ko, Stewart, Fox, Konolige, & Limketkai, 2003), (Fox et al., 2006) Known: (Gil, Reinoso, Ballesta, & Miguel, 2010), (Thrun, 2001)
b	Uncertainty of the Relative Poses	Uncertainty of the relative poses should be propagated to the maps and poses of the robots	Linearized uncertainty propagation: (Saeedi, Paull, Trentini, & Li, 2011c), (Zhou & Roumeliotis, 2006)
c	Updating Maps and Poses	Once the relative poses are known, past and future poses and maps should be updated	Particle filter: (Howard, Sukhatme, & Mataric, 2006b) Batch update: (Saeedi et al., 2014a) Least squares: (Kim et al., 2010)
d	Line-of-sight Observations	Include line-of-sight observation to improve mapping and localization	Manifold representation: (Howard, 2004)
e	Closing Loops	Detect loop closure with multiple robots	Filtering: (Howard, 2006), (Zhou & Roumeliotis, 2006), (Thrun, 2001) Least squares: (Kim et al., 2010)
f	Complexity	Time and space complexity with respect to the size of the team	Information filter: (Thrun & Liu, 2005)
g	Communications	Out-of-sequence measurements and blackouts	Central equivalent: (Leung et al., 2012) Buffering: (Carlone et al., 2010)
h	Heterogeneous Vehicles and Sensors	Different types of robots and sensors	Aerial and ground robots: (Vidal-Calleja, Berger, Sola, & Lacroix, 2011) (Kim et al., 2010), Different ground robots: (Kim et al., 2010)
i	Synchronization	Synchronized clocks between robots	Network Time Protocol (NTP): (Leung et al., 2011)
j	Performance Measure	Measuring the accuracy of the model	(Erinc & Carpin, 2014)

performance prediction (Mourikis & Roumeliotis, 2006), robot rendezvous (Zhou & Roumeliotis, 2006), and distributed multi-robot SLAM (Jafri & Chellali, 2013). Some other algorithms, such as heterogeneous submap matching (Vidal-Calleja et al., 2011) or constrained local submap filter (Williams, Dissanayake, & Durrant-Whyte, 2002b), are a combination of the EKF and other techniques, which are reviewed in the other sections of this paper.

Extension from a single robot to multiple robots using EKF is straightforward, and this motivates researchers to solve various issues in multiple-robot SLAM using an EKF-based framework. Thus, most EKF-based approaches are very similar. In the rest of this section, one of the many EKF-based solutions is reviewed in detail.

Robot rendezvous (Zhou & Roumeliotis, 2006) is a feature-based SLAM and uses EKF for filtering robot poses and landmark positions. In robot rendezvous, four main problems are addressed: unknown relative poses (problem a), uncertainty of the relative poses (problem b), updating maps and poses (problem c), and complexity (problem f). The state vector includes poses of the robots and positions of the landmarks. For example, for two robots a and b, and N features in the map, the state vector is

$$x_t^{ab} = \left[\, x_t^a \quad x_t^b \quad x_t^{l_1} \quad y_t^{l_1} \quad \ldots \quad x_t^{l_N} \quad y_t^{l_N}, \right]^T \qquad (8)$$

where x_t^a and x_t^b are the poses of robots a and b at time t, and $x_t^{l_i}$ and $y_t^{l_i}$, $i = 1, ..N$, are 2D coordinates of the i^{th} feature. The covariance of x_t^{ab} is shown by Σ_t^{ab}.

In robot rendezvous, robots should meet either by random or by arrangement at least once. The advantage of this method is that it does not require overlaps between maps; however, the robots need to meet at a point. When robots are in the line-of-sight, robot-to-robot measurements are used to calculate the transformation between maps. The uncertainty of the relative transformation is calculated from the line-of-sight observations, by linearizing the observation equations. The resulting uncertainty is presented as a covariance matrix.

Now that the relative poses are known, the maps are merged using the calculated transformation. During map merging, it is highly possible that there are duplicates of the same landmark due to the uncertainty of the position of the landmark. The so-called *Sequential Nearest Neighbor Test* is proposed to detect these cases and combine duplicate landmarks. The test is based on the Mahalanobis distance. If the distance of two landmarks in the fused map is smaller than a threshold, they are considered to be duplicates. To update the map, state and covariance row and column of the duplicate landmarks are deleted. If the landmarks' error is bigger than the distance between them, then this method

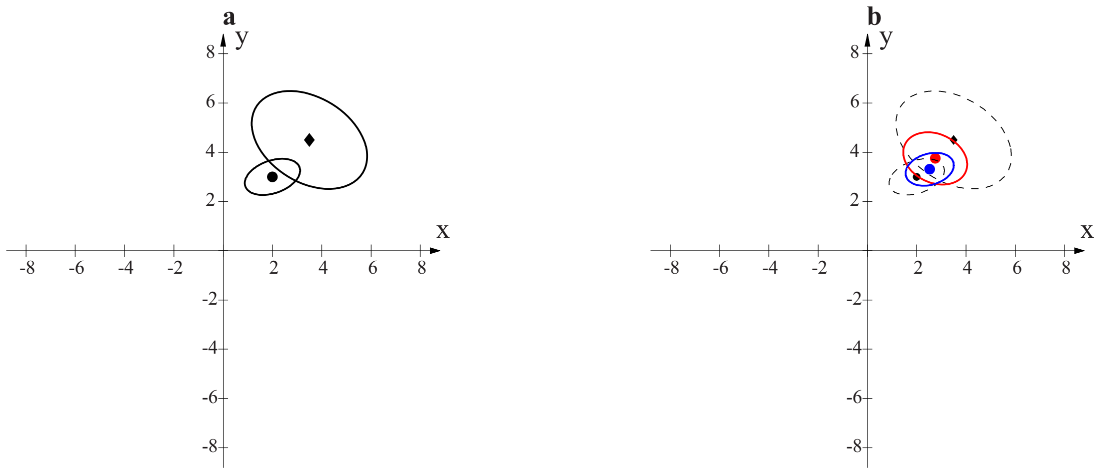


Figure 9. Merging two features. a) Two features whose separation is smaller than a given threshold. b) The new merged feature is shown in red. Features are merged by averaging their mean positions. If weighted averaging is used, the uncertainty after fusion decreases (shown in blue). In this case, the information matrices are added to generate the information matrix of the fused feature. It is also possible to average the features with equal weights (shown in red).

will not be effective. Therefore, to avoid false duplicate detection, this process is performed in the vicinity of two robots, where it is more likely to have less error in the position of the landmarks. This process is performed sequentially.

In Figure 9(a), two features are depicted. One of them is shown by a circle, the other one by a diamond. Assume the diamond feature has been transformed from another map, and the circle one was already existing in the current map. For each pair of such features, their distance is calculated. If the distance is less than the threshold, then the features are averaged to generate a new feature that represents both features. If weighted averaging is performed, in which the weights are proportional to the information matrices of the features, the information matrix of the fused feature is the summation of the information matrices of the features. In this case, the fused feature has less uncertainty than the original features (see Figure 9(b), the blue ellipse). The weighted averaging requires matrix inversion and is computationally demanding. An alternative approach is performing averaging with equal weights. In this case, the covariance matrices are also averaged; however, the uncertainty of the resulting feature might be larger than the uncertainty of the most certain feature (see Figure 9(b), the red ellipse).

Algorithm 1 summarizes the pseudocode for EKF implementation of feature-based multiple-robot SLAM, assuming the poses of the robots are known. Functions predict(·) and update(·) are standard procedures of the Kalman filter. These functions are explained in (Thrun et al., 2005, Ch. 10). Function augment(·) adds new features to the state vector, defined in Eq. (8). Finally, function merge(·) merges duplicate features as explained in the previous paragraph.

Algorithm 1 EKF-based SLAM for two robots, a and b, with known relative poses.

Input: posterior at time $t-1$: x^{ab}_{t-1}, Σ^{ab}_{t-1}
control signals at time t: u^a_t, u^b_t,
observations at time t: z^a_t, z^b_t.
Output: posterior at time t: x^{ab}_t, Σ^{ab}_t.

```
for robot id = a → b do
    (x̄_t^ab, Σ̄_t^ab) ← predict(x_{t-1}^ab, Σ_{t-1}^ab, u_t^id)
    (x_t^ab, Σ_t^ab) ← update(x̄_t^ab, Σ̄_t^ab, z_t^id)
    (x_t^ab, Σ_t^ab) ← augment(x_t^ab, Σ_t^ab, z_t^id)
end for
(x_t^ab, Σ_t^ab) ← merge(x_t^ab, Σ_t^ab)
return (x_t^ab, Σ_t^ab)
```

To efficiently search for the nearest landmarks, a kd-tree is used, which reduces the computational complexity. Moreover, the EKF estimator works iteratively; thus, previous measurements do not need to be processed repeatedly. The recursive operation reduces both memory and computational requirements. The simulated and real-world experiments show the consistency of the final fused map.

6.2. EIF-SLAM

Nettleton et al. were the first who applied the information formulation to the multiple-robot SLAM problem (Nettleton, Gibbens, & Durrant-Whyte, 2000). Inherently, the EIF is more suitable for a multirobot system than EKF, because the information has the additivity property. The key discovery in this early work was the complexity of the EIF (problem f). They noticed that updating maps is possible in time logarithmic in the number of the robots on the team. Later, Thrun et al. improved on this work and proposed feature-based multiple-robot SLAM based on Sparse EIF (Thrun & Liu, 2005).

In SEIF, two main problems are addressed: finding the relative poses (problem a) and updating maps and poses (problem c). The relative poses are calculated by matching the features of the maps. This can be done by a kd-tree or similar algorithms. Once an estimate is calculated, it can be optimized by minimizing the quadratic displacement of the features. After calculating the relative transformation, maps and poses, in the information domain, are transformed to the global coordinates.

As mentioned, additivity is an interesting property of the information filter. Robots can integrate their information by simply adding them. Also fusion of duplicate landmarks is straightforward. For instance, assume in the following information matrix and vector of a map with four features, features 2 and 4 are duplicate features.

$$\begin{bmatrix} \Omega_{11} & \Omega_{12} & \Omega_{13} & \Omega_{14} \\ \Omega_{21} & \Omega_{22} & \Omega_{23} & \Omega_{24} \\ \Omega_{31} & \Omega_{32} & \Omega_{33} & \Omega_{34} \\ \Omega_{41} & \Omega_{42} & \Omega_{43} & \Omega_{44} \end{bmatrix}, \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix}, \tag{9}$$

In the EIF, these two features collapsed to one feature as follows:

$$\begin{bmatrix} \Omega_{11} & \Omega_{12}+\Omega_{14} & \Omega_{13} \\ \Omega_{21}+\Omega_{41} & \Omega_{22}+\Omega_{42}+\Omega_{24}+\Omega_{44} & \Omega_{23}+\Omega_{43} \\ \Omega_{31} & \Omega_{32}+\Omega_{34} & \Omega_{33} \end{bmatrix}, \begin{bmatrix} \zeta_1 \\ \zeta_2+\zeta_4 \\ \zeta_3 \end{bmatrix}, \tag{10}$$

The presented experimental results, with 10^8 features, demonstrate the effectiveness of this algorithm. Derivation and detail algorithmic implementation of EIF-based multiple-robot SLAM is explained in (Thrun et al., 2005, p. 424).

6.3. PF-SLAM

Multiple-robot SLAM using particle filtering was first proposed in 2001 by Thrun at Carnegie Mellon University (Thrun, 2001). Later in 2006, his work was extended by Howard at the University of Southern California (Howard, 2006). These two works proposed view-based SLAM. Since then, many other researchers tried to extend boundaries of multiple-robot SLAM. In this section, three key works in this area are explained.

Thrun, in his work, solves two key problems in multiple-robot SLAM: updating maps and poses (problem c) and closing loops (problem e). The core of Thrun's work is "a statistical framework that combines an incremental maximum likelihood estimator with a posterior pose estimator" (Thrun, 2001). This method combines maximum likelihood mapping with Monte Carlo localization. In this method, the probabilistic motion model and the probabilistic measurement model based on scan matching are used. For the measurement model, it is assumed that each scan constitutes a local map, which is composed of three types of areas: free space, occupied space, and occluded (unknown) space. The same type of areas are used to develop maps from scans.

Compared with the extended Kalman filter, Thrun's work has the advantage that it can model multimodal and non-Gaussian distribution. This method is robust and can close loops. However, there are two main limitations. First, the robots must begin their mapping in nearby locations to have overlaps in their range scans (to have large overlaps in their initial maps). Second, the initial poses of the robots are assumed to be known approximately prior to the start of the mission. In the next algorithm (Howard, 2006), more detail of this approach is explained.

Howard extends the work in (Thrun, 2001) and proposes a multiple-robot SLAM using Rao–Blackwellised particle filter (Howard, 2006). Howard et al. also extend the work in (Howard, 2006) to include path planning and exploration with a central agent and a team of 80 robots (Howard et al., 2006a).

The proposed method in (Howard, 2006) addresses three of the introduced key problems in multiple-robot SLAM: relative poses of robots (problem a), updating maps and poses (problem c), and closing loops (problem e). Robots without knowing their relative poses cannot generate a global map. In this work, this problem has been addressed by line-of-sight observation, using cameras mounted on the robots. Moreover, updating past information, prior to the knowledge of relative poses of the robots, is a key problem. If this problem is addressed properly, it may help to close loops using the past information.

In this method, it is assumed that either robots' relative poses are given or robots will meet one another at a point. At the meeting point, which happens within the line-of-sight, robots identify their relative positions using cameras and unique fiducials mounted on them. At the first meeting point, a particle filter is applied to the data in reverse temporal sequence to fuse all the data from the initial point. The future meetings are ignored. Assume that the pose of the robots are known. Using Rao–Blackwellization for multiple robots, the desired posterior of multiple-robot SLAM for two robots a and b, introduced in Eq. (7), is written as

$$\begin{aligned} & p\left(x^a_{1:t}, x^b_{1:t}, m_t | z^a_{1:t}, z^b_{1:t}, u^a_{1:t}, u^b_{1:t}, x^a_0, x^b_0\right) \\ & = p\left(m_t | z^a_{1:t}, z^b_{1:t}, x^a_{1:t}, x^b_{1:t}, x^a_0, x^b_0\right) \\ & \quad \times p\left(x^a_{1:t}, x^b_{1:t} | z^a_{1:t}, z^b_{1:t}, u^a_{1:t}, u^b_{1:t}, x^a_0, x^b_0\right) \end{aligned} \tag{11}$$

By ignoring line-of-sight observation and assuming that the trajectories of the robots are decoupled, the following factorization can be written:

$$p\left(x^a_{1:t}, x^b_{1:t}, m_t | z^a_{1:t}, z^b_{1:t}, u^a_{1:t}, u^b_{1:t}, x^a_0, x^b_0\right)$$
$$= p\left(m_t | z^a_{1:t}, z^b_{1:t}, x^a_{1:t}, x^b_{1:t}, x^a_0, x^b_0\right)$$
$$\times\, p\left(x^a_{1:t} | z^a_{1:t}, u^a_{1:t}, x^a_0\right)$$
$$\times\, p\left(x^b_{1:t} | z^b_{1:t}, u^b_{1:t}, x^b_0\right) \tag{12}$$

$p(m_t | z^a_{1:t}, z^b_{1:t}, x^a_{1:t}, x^b_{1:t}, x^a_0, x^b_0)$ is the mapping problem with known poses and is calculated using ray tracing (Thrun et al., 2005, p. 154). The other two terms, which are posteriors of poses of each robot, can be calculated using a particle filter.

The particle set for Eq. (12) is constructed as follows. The i^{th} particle is given as

$$< x^{a(i)}_t, x^{b(i)}_t, m^{(i)}_t, w^{(i)}_t > \tag{13}$$

where $x^{a(i)}_t$ and $x^{b(i)}_t$ are poses of two robots at time t, $m^{(i)}_t$ is the map, and $w^{(i)}_t$ is the weight of the particle.

Algorithm 2 summarizes the particle filter implementation for Eq. (12). In lines 4-5, predictions based on the motion models are performed. The sign $\sim$ denotes that the samples on the left-hand side are generated according to the distribution given on the right-hand side. In line 6, weights of particles are updated. In line 7, the map of each particle is updated, given the pose and measurement of each robot. The map is updated by ray tracing, explained in (Thrun et al., 2005, p. 154), and function map$(\cdot)$ performs this operation. No matter how many pairs of poses and measurements are in the argument of the function map$(\cdot)$, each pair is added to the map using ray tracing. In line 8, the updated particle is added to the posterior particle set. Finally, in line 10, resampling is performed.

Algorithm 2 Particle filter for multiple robots with known relative poses and decoupled trajectories.

Input: set of particles representing posterior at time $t-1$: S_{t-1},
control signals at time t: u^a_t, u^b_t,
observations at time t: z^a_t, z^b_t.
Output: set of particles representing posterior at time t: S_t.

1: $S_t \leftarrow \emptyset$
2: **for** $i = 1 \rightarrow M$ **do**
3: $< x^{a(i)}_{t-1}, x^{b(i)}_{t-1}, m^{(i)}_{t-1}, w^{(i)}_{t-1} > \leftarrow S^{(i)}_{t-1}$
4: sample $x^{a(i)}_t \sim p(x^a_t | x^{a(i)}_{t-1}, u^a_t)$
5: sample $x^{b(i)}_t \sim p(x^b_t | x^{b(i)}_{t-1}, u^b_t)$
6: $w^{(i)}_t \leftarrow p(z^a_t | x^{a(i)}_t, m^{(i)}_{t-1})\, p(z^b_t | x^{b(i)}_t, m^{(i)}_{t-1})\, w^{(i)}_{t-1}$
7: $m^{(i)}_t \leftarrow \mathrm{map}(m^{(i)}_{t-1}, z^a_t, z^b_t, x^{a(i)}_t, x^{b(i)}_t)$
8: $S_t \leftarrow S_t \cup < x^{a(i)}_t, x^{b(i)}_t, m^{(i)}_t, w^{(i)}_t >$
9: **end for**
10: resample(S_t)
11: return S_t

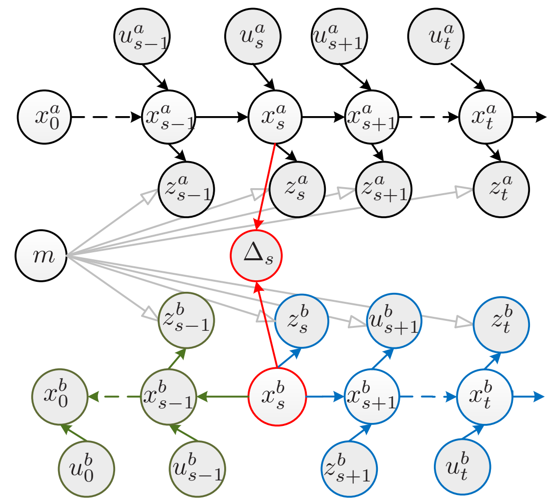


Figure 10. Bayes net for particle filter with a virtual robot. At time s, Δ_s is known, shown in red. Past data, shown in green, are updated using the virtual robot. Future data, shown in blue, are updated using a particle filter (Howard, 2006).

When the robots do not know their initial poses, they start SLAM independently and store all raw data until they meet each other. Assume that the local coordinates of robot a is the global coordinates system of all robots. At the meeting (at time s, $0 < s < t$), where they can calculate their relative pose, the desired posterior of SLAM is defined as:

$$p(x^a_{1:t}, x^b_{1:t}, m_t | z^a_{1:t}, z^b_{1:t}, u^a_{1:t}, u^b_{1:t}, x^a_0, \Delta_s), \tag{14}$$

where Δ_s is the relative transformation of the poses of the robots at time s by ignoring the uncertainty of the relative transformation, this posterior is calculated in two steps using the concept of the virtual robot. For time $0:s$, a filter is constructed that processes all data from time s to time 0 in reverse temporal order, which is named as the virtual robot. Once all past data are processed, for time $s:t$, algorithm 2 is applied. Figure 10 shows the Bayes net of this approach. At time s, Δ_s is known, shown in red. Past data, shown in green, are updated using the virtual robot. Future data, shown in blue, is updated using a particle filter.

This approach is effective and the maps and poses are updated in real time, but it cannot be applied when robots cannot see one another or they do not know their relative poses initially. Also, no uncertainty for relative poses is considered. When the robots have mutual observations, mutual laser observation, which are induced by other robots, are excluded. This approach has been tested on a team of four robots.

Carlone et al. also apply a particle filter for multiple-robot SLAM (Carlone, Ng, Du, Bona, & Indri, 2010), extending Howard's method (Howard, 2006). Their contribution includes considering the uncertainty of the relative transformation (problem b) in the mapping, using grid mapping

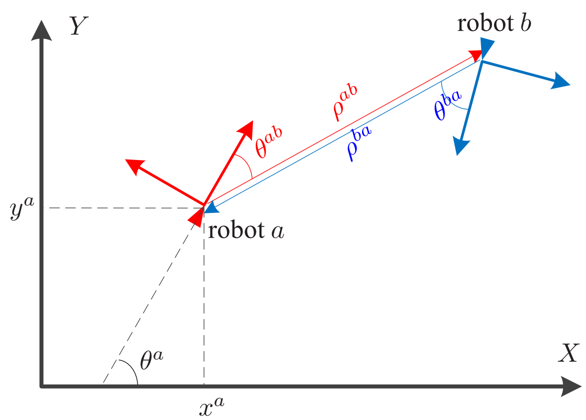


Figure 11. Two robots, a and b, detect each other using line-of-sight observations. The relative poses and the uncertainty of the relative pose is calculated from the line-of-sight observations (Carlone et al., 2010).

(Grisetti et al., 2007) for particle filtering. Also, they consider a mild assumption on wireless communication (problem g). The work by Carlone et al. is motivated by the fact that incorporating the uncertainty of the relative transformation, which is calculated using line-of-sight observation, can improve the accuracy of mapping and localization.

Based on the work in (Carlone et al., 2010), assume two robots, a and b, can see each other. ρ^{ab} and θ^{ab} are the line-of-sight range and bearing of b, measured by robot a. The uncertainties associated with these measurements are $\sigma_{\rho^{ab}}$ and $\sigma_{\theta^{ab}}$. ρ^{ba} and θ^{ba} are the same but measured by robot b. Similarly, the uncertainties of the measurements are $\sigma_{\rho^{ba}}$ and $\sigma_{\theta^{ba}}$. Because there are two sources for the range measurements, they are averaged using the following relations:

$$\rho = \frac{\sigma^2_{\rho^{ba}}}{\sigma^2_{\rho^{ab}} + \sigma^2_{\rho^{ba}}}\rho^{ab} + \frac{\sigma^2_{\rho^{ab}}}{\sigma^2_{\rho^{ab}} + \sigma^2_{\rho^{ba}}}\rho^{ba}, \tag{15}$$

$$\frac{1}{\sigma^2_{\rho}} = \frac{1}{\sigma^2_{\rho^{ab}}} + \frac{1}{\sigma^2_{\rho^{ba}}}, \tag{16}$$

where ρ is the averaged range measurement, and σ^2_ρ is its uncertainty. The weights of the averaging is determined from the uncertainty of the measurements. Now the relative pose can be describe as

$$\left[\, \rho\cos\theta^{ab} \quad \rho\sin\theta^{ab} \quad \pi + \theta^{ab} - \theta^{ba} \,\right]. \tag{17}$$

The first-order approximation of the corresponding uncertainty is given according to the following matrix

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_{\theta\theta} \end{bmatrix}, \tag{18}$$

where

$$\sigma_{xx} = \sigma^2_\rho \cos^2\theta^{ab} + \sigma^2_{\theta^{ab}}\rho^2\sin^2\theta^{ab} \tag{19}$$

$$\sigma_{yy} = \sigma^2_\rho \sin^2\theta^{ab} + \sigma^2_{\theta^{ab}}\rho^2\cos^2\theta^{ab} \tag{20}$$

$$\sigma_{xy} = \frac{1}{2}\left(\sigma^2_\rho - \sigma^2_{\theta^{ab}}\rho^2\right)\sin(2\theta^{ab}) \tag{21}$$

$$\sigma_{x\theta} = -\rho\sigma^2_{\theta^{ab}}\sin\theta^{ab} \tag{22}$$

$$\sigma_{y\theta} = \rho\sigma^2_{\theta^{ab}}\cos\theta^{ab} \tag{23}$$

$$\sigma_{\theta\theta} = \sigma^2_{\theta^{ab}} + \sigma^2_{\theta^{ba}}. \tag{24}$$

If $[x^a \;\; y^a \;\; \theta^a]$ is the pose of robot a in the global coordinates, and $[\hat{x}^b \;\; \hat{y}^b \;\; \hat{\theta}^b]$ is the pose of robot b in its own local coordinates, the following relation calculates the pose of robot b in the global coordinates ($\delta_\theta = \theta^{ab} - \theta^{ba}$):

$$\begin{bmatrix} x^b \\ y^b \end{bmatrix} = \begin{bmatrix} \cos\delta_\theta & -\sin\delta_\theta \\ \sin\delta_\theta & \cos\delta_\theta \end{bmatrix}\begin{bmatrix} \hat{x}^b \\ \hat{y}^b \end{bmatrix} + \begin{bmatrix} x^a \\ y^a \end{bmatrix} + \begin{bmatrix} \rho\cos(\theta^a + \theta^{ab}) \\ \rho\sin(\theta^a + \theta^{ab}) \end{bmatrix} \tag{25}$$

$$\theta^b = \theta^a - \hat{\theta}^b + \pi + \delta_\theta. \tag{26}$$

To apply the calculated uncertainty to the particles, in the prediction step of the particle filter, instead of using odometry measurements, the Guassian distribution given in (17) and (18) is used.

The work by Carlone et al. does not hold a joint posterior for trajectories and relative poses of the robots. In the proposed solution, robots exchange data only in encounters. This way, having access to a communication channel is not required at all times, but it can cause buffer overflow, or it can occupy all processing resources for processing all buffered data. Also, in this work, all encounters are taken into account to improve relative poses, unlike the work by A. Howard which only takes into account the first encounter (Howard, 2006). In this work, one experiment with two robots is presented. This work also has exponential space complexity.

Gil et al. propose a feature-based multiple-robot SLAM (Gil et al., 2010). This algorithm extends FASTSLAM1.0 (Montemerlo et al., 2002) to multiple robots in a feature-based environment. The proposed method addresses updating maps and poses problem (problem c), which is managing the data association of visual features, assuming the visual descriptors have been affected by noise. The extension to FASTSLAM1.0 is done by adding poses of all robots to the particles (similar to (13), where in this case the map is composed of the locations of the features). Consequently, to better model the multivariate distribution of the poses, the number of particles should be increased exponentially with

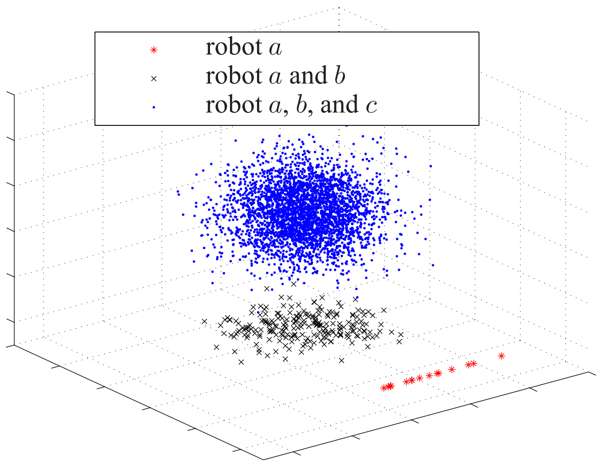


Figure 12. In a team, as the number of the robots increases, the number of the particles increases. For instance, for only one robot, robot a, 15 particles are required. For two robots, a and b, 15×15 particles are required, and for three robots $15 \times 15 \times 15$ particles are needed.

respect to the number of the robots. Figure 12 shows the growth of the number of the particles with respects to the number of robots. The pose of only one robot, robot a, has been modeled with $n_1 = 15$ particles, shown with red stars. For two robots, a and b, the proper size to account for all possible cases is $n_2 = 15 \times 15$, shown with black crosses. For three robots, $n_3 = 15 \times 15 \times 15$ particles are used, shown in blue dots.

When a feature is observed multiple times or observed at the same time by multiple robots, the feature either already exists in the map or is a new feature. This decision is not easy, because the features position and its descriptor (signature) are affected by noise. To account for this problem, two distance criteria are defined: a Mahalanobis distance (D) for the position of the feature and a Euclidean distance (E) for the descriptor of the feature:

$$D = (v - \hat{v})^T \Sigma^{-1} (v - \hat{v}) \tag{27}$$

$$E = (s - \hat{s})^T (s - \hat{s}), \tag{28}$$

where Σ is the uncertainty of the measurements and v and s are position and descriptor of an associated landmark with corresponding measurements $\hat{v}$ and $\hat{s}$. If D and E are smaller than given thresholds, then the data association is correct; otherwise, the feature is considered as a new feature. The proposed algorithm by Gil et al. has been verified in simulated environments, in which robots can send their information to a central agent. Moreover, it is assumed that initial relative poses of the robots are approximately known, and the direct encounters between the robots are ignored.

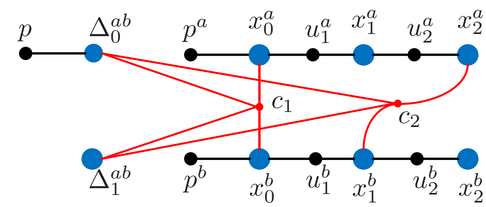


Figure 13. GraphSLAM for two robots, a and b. Anchor nodes for initial poses (p^a and p^b) and relative poses between two robots (Δ_0^{ab} and Δ_1^{ab}) are defined. In GraphSLAM for multiple robots, the optimization is performed on the motion and observation constraints of each robot (black lines) and also on the observations between the robots (red lines). Node p is the anchor node that connects all robots to the global coordinates (Kim et al., 2010).

6.4. GraphSLAM

GraphSLAM is a smoothing approach (see Section 4.6), which is considered as a least squares error problem. Although there are other approaches for smoothing, such as the work in (Pillonetto, Bell, & Del Favero, 2013), in general, most smoothing approaches are based on least squares. GraphSLAM has been used for multiple robots, for instance, C-SAM, a batch algorithm for feature-based map merging (Andersson & Nygards, 2008), decentralized SLAM (Cunningham & Dellaert, 2012; Cunningham, Paluri, & Dellaert, 2010; Cunningham, Wurm, Burgard, & Dellaert, 2012; Indelman, Nelson, Michael, & Dellaert, 2014), and the work proposed in (Kim et al., 2010).

Kim et al. extend iSAM for multiple robots based on multiple pose graphs in which the relationship between multiple pose graphs are formulated and optimized to generate a consistent solution (Kim et al., 2010). The proposed solution addresses the unknown relative poses (problem a) using direct encounters and updating maps and poses (problem c) using GraphSLAM. Moreover, this paper applies the algorithm to a team of heterogeneous robots, including a ground robot and a quadrotor (problem h).

Encounters are defined to be direct, when the robots can see each other, or indirect, when the robots see the same environment. Direct encounters, used to calculate the relative poses of the robots, are determined using a camera on the quadrotor to detect a checkerboard pattern on the ground robot. When an encounter occurs, the relative poses can be further refined using scan matching.

Kim et al. consider constraints between the robots, when they see each other (direct encounter), and constraints due to observing the same environment (indirect encounter). In Figure 13, the graph corresponding to the poses and observations of two robots a and b are shown. The blue dots are poses of the robots, connected to each other by control actions (black small dots). For simplicity, observations, except encounters, are not shown. The encounters between the robots are shown by red lines. c_1 is a direct encounter, but c_2 is an indirect encounter, because it happens at different times. The trajectory of each robot is anchored by priors, p^a and p^b, on its first pose and is chosen arbitrarily. Δ^a and Δ^b are anchors of the trajectories that specify the offset of each trajectory with respect to a common global frame.

The GraphSLAM solution is a least squares solution and corresponds to the maximum a posteriori (MAP) estimate of the trajectories of the robots ($X^\star$). Assume there are two robots, $r = a, b$ and there exist M_r pose for each robot, N_r observations for each one, and C encounters, defined as a set $\{c_k\}_{k=1}^{C}$. The optimization problem is defined as follows

$$X^\star = \underset{X}{\operatorname{argmin}} \left\{ \left(\sum_{r,r'=a,b,\, r'\neq r} \left|\left| p^r - x_0^r \right|\right|_{\Sigma}^2 + \sum_{i=1}^{M_r} \left|\left| f(x_{i-1}^r, u_i^r) - x_i^r \right|\right|_{\Lambda_i^r}^2 + \sum_{j=1}^{N_r} \left|\left| h(x_j^r, m) - z_j^r \right|\right|_{\Omega_j^r}^2 + \sum_{k=1}^{C} \left|\left| g(x_k^{r'}, c_k, \Delta^r, \Delta^{r'}) - x_k^r \right|\right|_{\Gamma_k}^2 \right) \right\} \tag{29}$$

In this formulation, $f(\cdot)$ is the motion model, $h(\cdot)$ is the observation model that each robot makes from the world, and $g(\cdot)$ is the observation model that models the encounters. As this formulation shows, to extend iSAM for multiple robots, direct and indirect encounters are considered as constraints in the optimization process. Therefore, the same solution and factorization which is used for single-robot SLAM is easily applied to multiple-robot SLAM. The optimization is performed by linearizing equation (29) and is an efficient solution. However, the optimization is performed over all trajectories of the robots. Therefore, for very long trajectories, high computational processing is required.

Nagatani et al. present a 3D GraphSLAM for cooperative mapping by multiple robots (Nagatani et al., 2011). Their work has been developed for autonomous navigation in rescue missions and has been presented in RoboCupRescue 2009. The main focus of the work is addressing the problem of updating maps and poses (problem c). The work by Nagatani et al. is an extension to the method proposed in (Takeuchi & Tsubouchi, 2008). For the cooperative mapping, digital elevation maps are built, and by applying odometric and scan matching constraints, optimization in the global perspective is performed. Figure 14 shows a sample 3D map developed using laser ranger measurements (Nagatani et al., 2011).

The underlying idea for GraphSLAM solutions are the same: an optimization is performed over nonlinear motion and observation constraints. However, their difference is the approach chosen for the optimization. Kim et al. use QR factorization combined with the variable reordering method

Figure 14. A 3D map developed by laser ranger measurements using GraphSLAM. Image courtesy of Keiji Nagatani et al. (Nagatani et al., 2011).

(Kim et al., 2010), but Nagatani et al. use the skyline method, a popular analysis approach in FEM, and the Cuthill-Mckee reordering method (Nagatani et al., 2011).

Algorithm 3 summarizes the pseudocode of GraphSLAM implementation for two robots, with known relative poses. Function linearize$(\cdot)$ linearizes the system model given in Eq. (29). Then the linearized equations are solved in the next line. As explained, all GraphSLAM algorithms are similar; however, the approach taken to solve the linearized equations makes the algorithms different. The cycle of linearization and solving for the results continues until the solution converges.

Indelman et al. also present a GraphSLAM algorithm that addresses unknown relative poses and the multiple-robot data association problem (problem a) (Indelman et al., 2014). The key idea in the algorithm is to distinguish between inliers and outliers in multiple-robot data association using the fact that while each multirobot correspondence can be used to determine the transformation between the poses of the robots, only inlier correspondences will generate similar results. This key idea is used to identify the inliers (overlaps) and calculate the relative poses of the robots.

As explained in Section 4.6, submap matching and GraphSLAM are two major types of smoothing approaches for SLAM. While there are many algorithms for multiple-robot GraghSLAM, there are few methods for

Algorithm 3 GraphSLAM for two robots, a and b, with known relative poses.

Input: control signals: $u_{1:t}^a, u_{1:t}^b$,
observations: $z_{1:t}^a, z_{1:t}^b$.
Output: posterior: $x_{1:t}^{ab}, \Sigma_{1:t}^{ab}$.

1: **repeat**
2: $(\tilde{x}_{1:t}^{ab}, \tilde{\Sigma}_{1:t}^{ab}) \leftarrow \text{linearize}(u_{1:t}^a, u_{1:t}^b, z_{1:t}^a, z_{1:t}^b)$
3: $(x_{1:t}^{ab}, \Sigma_{1:t}^{ab}) \leftarrow \text{solve}(\tilde{x}_{1:t}^{ab}, \tilde{\Sigma}_{1:t}^{ab})$
4: **until** convergence
5: return $(x_{1:t}^{ab}, \Sigma_{1:t}^{ab})$

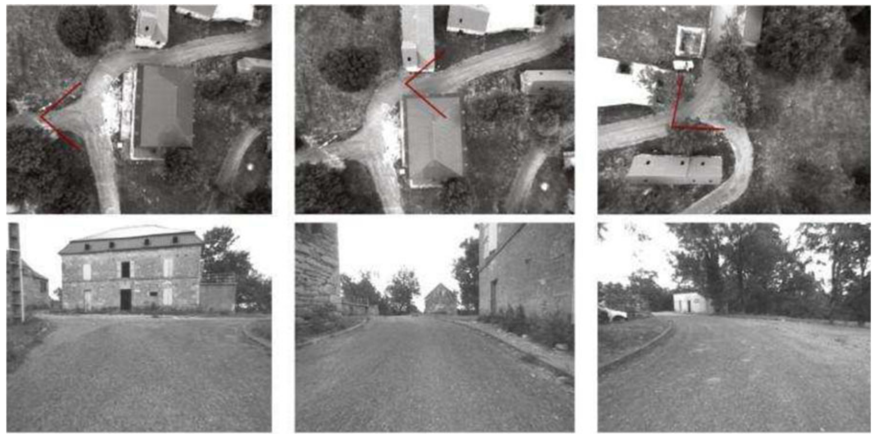


Figure 15. Three aerial (top) and ground (bottom) images. The aerial images are captured at about 40 m altitude. The pose of the ground robot in the aerial images is shown by red lines. A key problem in multiple robot SLAM is updating maps by images captured from different view. Image courtesy of Teresa Vidal-Calleja et al. (Vidal-Calleja et al., 2011).

multiple-robot submap matching. Vidal-Calleja et al. developed a heterogeneous multiple-robot SLAM using submap matching (Vidal-Calleja et al., 2011). The proposed solution addresses unknown relative poses (problem a), updating maps and poses (problem c), and heterogeneous robots (problem h).

The team of robots is composed of one ground robot and one helicopter. The proposed approach develops a heterogeneous map in which the map features are composed of points and line segments. On the helicopter, a monocular camera, which provides bearing-only observations, is used to identify features. To estimate 3D points, inverse-depth parameterization is used. To estimate 3D line segments, anchored Plucker coordinates are extended by adding endpoints. Updating maps and poses with different views from the ground and the aerial vehicle is a challenging problem. For example, Figure 15 shows an example of two views, captured at the same time, from the ground and the aerial vehicles. The optimization is an extension to hierarchical SLAM by Estrada et al., in which each robot builds local maps using an EKF, and then the spatial consistency of the maps of the robots is achieved by an optimization process (Estrada et al., 2005). The algorithm has been tested with a helicopter and a ground vehicle in a large-scale semistructured terrain.

6.5. Cooperative Positioning System

In 1994, Kurazume et al. proposed the cooperative positioning system (CPS) (Kurazume, Nagata, & Hirose, 1994). Tobata et al. extend CPS to develop a 3D cooperative mapping system for a team of heterogeneous ground robots (Tobata et al., 2012). CPS, as will be explained, is a simple algorithm, and Tobata et al. use it to address the line-of-sight observations (problem d) and update maps and poses (problem c). Also, the robots on the team do not need to be homogeneous, and usually one robot, called the parent robot, has all the main sensors, such as the scanning laser ranger. The other robots, called child robots, act as moving landmarks, helping the parent robot to perform better localization.

As mentioned, the cooperative mapping is based on the concept of the CPS. Using CPS, robots know their locations accurately, and therefore, the cooperative mapping problem is reduced to the mapping problem with known poses. In CPS, robots are divided into two groups: parent and child. One group stays still and acts as stationary landmarks while the other group moves. Then, the moving group stops, and the stationary group starts to move. This alternating behavior continues until the team reaches the target position. This procedure allows the robots to perform accurate relative localization using *portable* landmarks.

Figure 16 shows this procedure. The parent robot is shown in blue, and the child robots are shown in red.

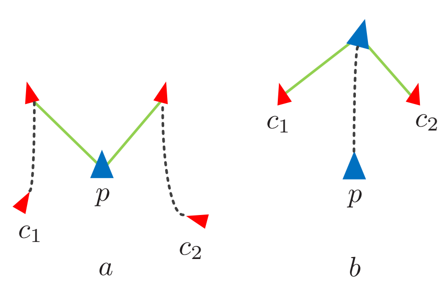


Figure 16. In cooperative positioning systems, robots are divided into two groups: parent (the blue triangle, marked with p) and child (the red triangles, marked with c_1 and c_2). These two groups move alternatively. Child robots move and stop, then the parent robots measures their poses. Next, the parent robot moves, stops, and measures the poses of the child robots.

Algorithm 4 CPS implementation for a parent and child robots.

1: Child robots move and stop
2: $z_1 \leftarrow$ the parent robot identifies child robots
3: The parent robot moves and stops
4: $z_2 \leftarrow$ the parent robot identifies child robots
5: triangulate z_1 and z_2 to find poses

Algorithm 4 summarizes the pseudocode of CPS implementation for the parent and child robots. As shown in Figure 16(a), the parent robot is still, and the child robots move. Then, the child robots stop (line 1). Now, the parent robot measures their poses (line 2, where z_1 is the first measurement of the parent robot). In the next step, the child robots stay still, and the parent robot moves and stops (line 3). Then, the parent robot measures the poses of the child robots (line 4, where z_2 is the second measurement of the parent robot). Using triangulation, the pose of all robots are calculated (line 5). Once the poses of the robots are known, mapping with known poses is performed. To minimize error accumulation, Tobata et al. perform GraphSLAM over the history of the poses. The major advantage of CPS-based mapping is that it can be used in unknown environments easily; however, it requires coordination between the agents to alternate the landmark role consistently. Figure 17 shows two 3D maps developed using the CPS algorithm in outdoor environments. A similar algorithm, called intrinsic localization and mapping (ILM), is proposed in (Dellaert, Alegre, & Martinson, 2003); however, in ILM, the restriction of having stationary robots is not imposed on the robots. It is also shown that by having stationary robots, the global accuracy of the solution is improved (Dellaert et al., 2003).

6.6. Submap Matching: Constrained Local Submap Filter

In 2002, Williams et al. proposed a solution for multiple-robot SLAM in feature-based environment using submap matching (Williams et al., 2002b). The proposed solution is called constrained local submap filter (CLSF). CLSF was first proposed by Williams et al. for single-robot SLAM and tested in underwater environments (Williams, Dissanayake, & Durrant-Whyte, 2002a). In fact, CLSF is a submap matching approach in which local maps are built using an EKF. As mentioned in Section 3, single-robot SLAM, submap matching is very efficient for large-scale mapping. The main reason for this efficiency is that the accumulated estimation error is set to zero when a new submap is constructed. The same concept can easily be extended to multiple robots.

The proposed CLSF for multiple-robot SLAM addresses three main problems: unknown relative poses (problems a), complexity (problem h), and updating maps (problem c). It is motivated by the fact that processing submaps, in regular intervals, saves computational resources compared with processing each measurement separately. In other words, CLSF reduces the complexity of multiple-robot SLAM by periodically fusing the local submap of the

Figure 17. Examples of 3D maps of outdoor environments using the cooperative positioning system. Image courtesy of Yukihiro Tobata et al. (Tobata et al., 2012).

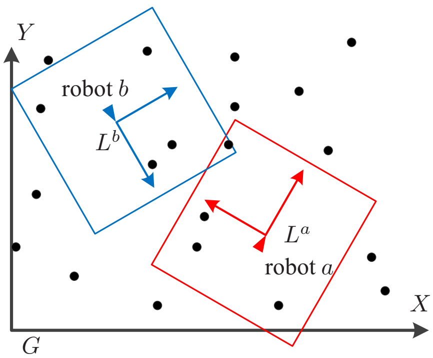


Figure 18. Two robots, a and b, develop two local maps. These maps are used to generate a global map. Robot a and its submap is shown in blue. Robot b and its submap is shown in red. The black dots are features.

features, generated from features nearby the robot, to the global map. This way, every observation is not fused directly into the global map, and therefore, the complexity decreases.

Figure 18 illustrates the key concepts of the solution. Two robots, a and b, are shown in red and blue. The global coordinate frame is marked with G. When the robots build new local maps, shown by frames L_a and L_b, the global coordinates of the robots are used to initialize the coordinates of the local maps. The local frames are centered at the current estimated pose of the robots. The state of the system is defined as follows:

$$\begin{pmatrix} X_t \\ {}^{L_a}X_t^a \\ {}^{L_b}X_t^b \end{pmatrix}. \tag{30}$$

Note the presuperscript of a variable denotes the coordinate frame of the variable. For example, ${}^{L_a}X_t$ means that X_t is measured in frame L_a. ${}^{G}X_t$, or simply X_t, means that X_t is measured in the global frame, G. The elements of the state vector are defined as follows:

- X_t: coordinates of local frames, L_a and L_b, in the global frame and the global map,
- ${}^{L_a}X_t^a$: pose and map of robot a in local frame L_a,
- ${}^{L_b}X_t^b$: pose and map of robot b in local frame L_b.

These elements are given as

$$X_t = \begin{pmatrix} x_t^{L_a} \\ x_t^{L_b} \\ m_t \end{pmatrix}, \quad {}^{L_a}X_t^a = \begin{pmatrix} {}^{L_a}x_t^a \\ {}^{L_a}m_t^a \end{pmatrix}, \quad {}^{L_b}X_t^b = \begin{pmatrix} {}^{L_b}x_t^b \\ {}^{L_b}m_t^b \end{pmatrix}, \tag{31}$$

where $x_t^{L_a}$ and $x_t^{L_b}$ are the global poses of frames L_a and L_b, and m_t is the global map. ${}^{L_a}x_t^a$ and ${}^{L_a}m_t^a$ are the pose and submap of robot a in its local frame, L_a. Similarly, ${}^{L_b}x_t^b$ and ${}^{L_a}m_t^b$ are defined for robot b. Once these local maps are created, the covariance between each of these element; X_t, ${}^{L_a}X_t^a$, and ${}^{L_b}X_t^b$; is zero. Moreover, because the uncertainty of the pose and features in the local map is usually small, the data association problem is also improved. In addition, data association can even be deferred until an improved local map is generated. Note that, in CLSF, the relative pose of the robots is determined by matching their map features. Also, uncertainties associated with poses and maps are maintained by the covariance matrix. The pseudocode of the CLSF algorithm is similar to algorithm 1, except that the state vector is defined according to Eq. (30).

6.7. Manifold Representation

Howard et al. propose a manifold representation for multiple-robot SLAM (Howard et al., 2006b). The proposed algorithm is designed for 2D environments. In conventional mapping methods, the map is represented in a 2D plane; however, in the proposed method, the map is represented as a manifold.

In a manifold map, while the robot develops a map, the trajectory and map of the robot form a spiral in the 3D space. Figure 19(a) shows a 2D map developed by a robot, overlaid on the blueprint of the environment. Figure 19(b) shows the manifold map of the same environment. The manifold representation has two main properties. The first property is *self-consistency*. As shown in Figure 19(a), due to error accumulation, the map is not consistent. However, on the manifold, the map is self-consistent. In other words, the inconsistency in the 2D map is not shown in the manifold representation. The second property is the *lazy loop closure*. It is obvious that in the manifold representation, one point in the world, might be represented by multiple points on the manifold. This many-to-one correspondence is useful for loop closure. In general, to perform loop closure, two points in the map that correspond to the same point in the world must be identified. It is important to know when to check for the correspondence between the points. Often this process might fail because of lack of enough information. However, if the loop closure is missed, the map will be

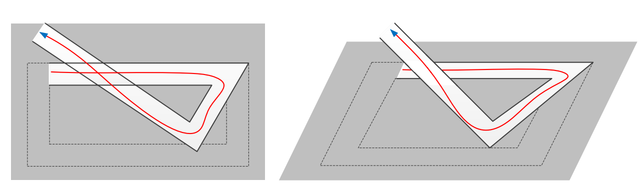


Figure 19. Manifold representation of a map. a) An inconsistent map due to error accumulation. b) Manifold representation of the same map. The manifold representation does not show up the self-inconsistency of the map.

Algorithm 5 Manifold algorithm for multiple-robot SLAM.

```
1: Make incremental manifold
2: if line-of-sight observation occurs then
3:     try loop closure
4:     try island merging
5: end if
```

inconsistent. In the manifold representation, the decision to make correspondence between the points can be delayed until the robots acquire enough information to establish the correspondence conclusively.

Manifold representation for multiple-robot SLAM addresses three main problems: unknown relative poses (problem a), line-of-sight observations (problem d), and loop closure (problem e). The unknown relative poses are determined using line-of-sight observations. The loop closure problem is performed using the manifold. For example, if two robots arrange a rendezvous at two points on the manifold, if the robots meet, the loop closure happens; otherwise, the points are distinct points, and there is no loop closure.

In manifold representation, multiple-robot SLAM is composed of three steps: incremental localization and mapping, loop closure, and island merging. Algorithm 1 summarizes the pseudocode of these steps. Incremental localization and mapping is the process of exploring the unknown environment, while incrementally making a map as a manifold (line 1). This step is similar to other mapping approaches, except the map is represented as a manifold. Incremental mapping and localization is accompanied with two other events: loop closure and island merging. Both events are triggered by line-of-sight observations between the robots and are used to make a global map. Loop closure occurs when two largely separated parts of a manifold map are brought together using line-of-sight observations between the robots (line 3). Island merging occurs when two unconnected parts of the manifold are fused together into a single representation (line 4). Island merging is also determined by line-of-sight observations, and it is similar to map merging, explained in Section 6.8. Island merging in multiple-robot SLAM is used when robots with unknown initial poses see each other. This process fuses local maps of the robots into a global map. This algorithm handles mutual observations and loop closure; however, because of the complexity of the algorithm, it poorly scales with the size of the team.

6.8. Map Merging

Map merging or map fusion is a solution to the multiple-robot SLAM, in which map data from robots are fused to generate a global map. This approach has many applications, such as in cooperative exploration (Fox et al., 2006), where the decision on the next exploration frontier requires a global map built by all robots. Map merging can be solved in two steps: 1) finding the required alignment between the maps, and 2) merging maps by integrating information from the aligned maps into one map. In general, the alignment between the maps can be determined from two sources: maps and poses. Therefore, if the relative poses are known or the common areas in the maps are determined, it is possible to calculate the transformation between the maps. The map merging problem can be studied in four different cases (Rone & Ben-Tzvi, 2013):

- **Known initial poses**: This is the simplest case of map merging. In this case, because the initial poses are known, the relative poses of the robots can be estimated at any time. Therefore, maps at any time can easily be merged. The assumption of knowledge of the initial poses is very limiting and in most applications does not hold true.
- **Rendezvous**: In rendezvous (Zhou & Roumeliotis, 2006), robots meet at a point. At the meeting point, the relative poses can be calculated through the line-of-sight measurements. Once the relative poses are known, maps can be merged. This approach adds another layer of the challenge to the problem, which is coordination for meeting.
- **Relative localization**: A simpler form of the rendezvous is relative localization. In this case, one robot localizes other robots in its own map (Ko et al., 2003); therefore, without rendezvous, the relative poses of the robots can be calculated. The challenge with this approach is that the other robots have to be present in the map of the localizing robot; otherwise the localization would fail. This approach might provide false results; therefore, it is important to have a verification procedure to reject false hypotheses. For instance, rendezvous is a good approach to verify the results (Fox et al., 2006). Figure 20 shows an example of relative localization. Two robots, *a* and *b*, have mapped different parts of an unknown environment. Robot *a* and its laser ranger measurements are shown in blue. The true pose of robot *b* and its laser ranger measurements are shown in red. Robot *a* uses its own map to localize robot *b*. To perform the localization, the measurement acquired by robot *b* is used. In reality, robot *b* is not located inside the map of robot *a*. Thus, performing relative localization fails to determine the true pose of robot *b*. In other words, because of the self-similarity of the environment, robot *b* is localized incorrectly, as shown by the green robot.
- **Relying on Overlaps**: In this approach, overlaps between the maps are used to calculate the relative transformation between the maps (Birk & Carpin, 2006). The challenge with this approach is finding the overlaps; however, this method does not need rendezvous and also robots can be out of each others' maps at any time.

In the rest of the section, map merging based on the overlaps of the maps is presented in detail for 2D SLAM. To

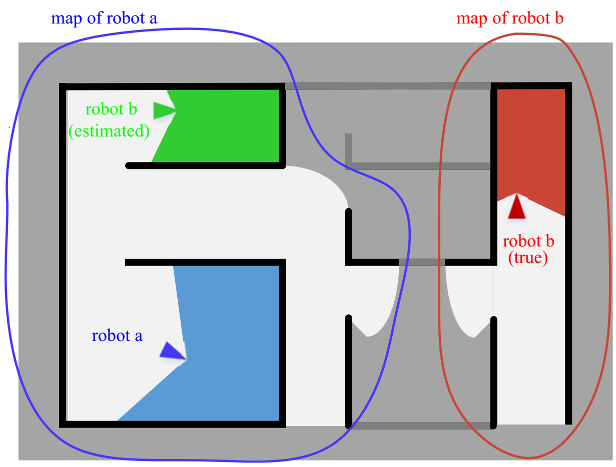


Figure 20. Relative localization. The measurement by robot b, shown in red, is used by robot a to localize robot b inside the map developed by robot a. Because of the self-similarity of the environment, robot b has been localized incorrectly, shown in green.

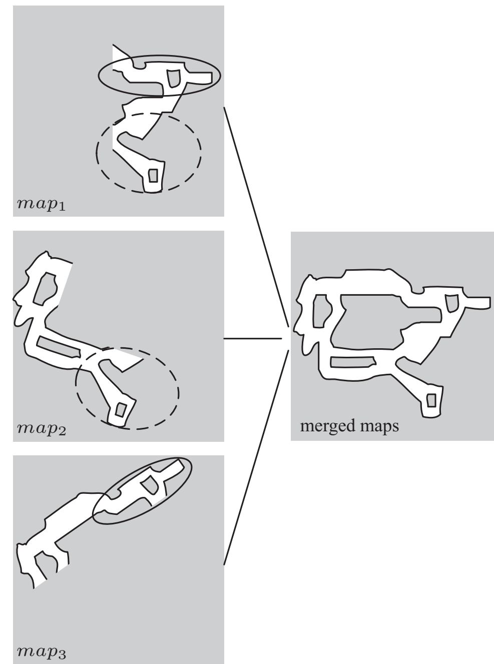


Figure 21. An example of map merging. Three maps are merged to form a complete map of the environment. The overlap between map_1 and map_2 is shown with a dashed ellipse. The overlap between map_1 and map_3 is shown with a solid line ellipse (Saeedi et al., 2014b).

the best of the authors' knowledge, there is no solution in the literature addressing 3D volumetric map merging. Let a transformation be composed of a 2×2 rotation matrix R_ψ and a 2×1 translation vector T as follows:

$$R_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}, \quad T = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}. \tag{32}$$

Let m_1 and m_2 be two occupancy grid maps. Assuming that m_2 is merged into m_1, the map merging problem is defined as follows: find a rotation matrix, R_ψ, and a translation vector, T, that transforms m_2 such that the overlaps of m_1 and the transformed m_2, m_2', fall squarely on top of each other. To do this, it is required to maximize a verification index defined on the merged map:

$$(R_\psi, T) = \underset{\psi,\delta_x,\delta_y}{\operatorname{argmax}}\ V(m_1, m_2'), \tag{33}$$

where m_2' means that the m_2 is transformed according to R_ψ and T. $V(\cdot)$ is a criterion, such as a similarity index (Birk & Carpin, 2006), which evaluates the merging process. This optimization is not easy to solve analytically, and some methods, such as (Carpin et al., 2005), use an exhaustive search to find a transformation matrix and then checks (33) to see whether it is satisfied.

Figure 21 depicts an example of map merging with three maps. The overlap between map_1 and map_2 is shown with a dashed ellipse, and the overlap between map_1 and map_3 is shown by a solid line ellipse. To find the required transformation, the first step is to identify overlaps, use the overlaps to calculate the alignment, and finally verify the results using (33). The map-merging problem can be interpreted as a special case of image registration in computer vision in which images are maps with special geometry. However, the problem is that, in multiple-robot map merging, the percentage of overlap between maps is usually low. If it is guaranteed that the overlaps between the images are large, then image registration techniques can be used for mapping. This concept is very similar to visual odometry, in which the consecutive image frames are matched to determine the relative motion between the frames. For instance, Elibol et al. apply a feature-based image registration to map an underwater environment with multiple robots (Elibol, Kim, Gracias, & Garcia, 2014). Figure 22 shows a map of the underwater environment developed by two robots.

As mentioned, merging the aligned maps is the second step of map merging. This is not a challenging problem and can be considered as an extension of mapping by a single robot.

Assume map m^a, developed by robot a, is an occupancy grid map with N grid cells, each with an associated binary random variable, representing its occupancy:

$$m^a = \{m_i^a\}, i = 1, \ldots, N. \tag{34}$$

The binary random variable specifies whether a cell is occupied ('1') or free ('0'), then $p(m_i^a = 1)$ or $p(m_i^a)$ represents the probability that the cell is occupied.

Assuming initially a cell i has *unknown* occupancy, $p(m_i^a) = 0.5$, at time t; if the cell is in the perception field

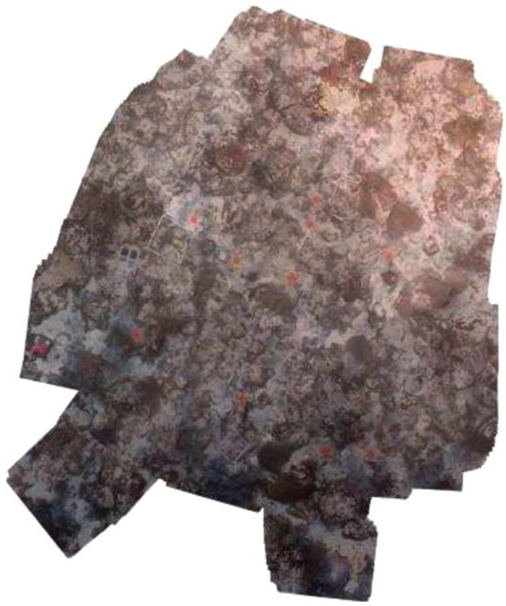


Figure 22. Map of an underwater environment, developed by two robots using feature-based image registration. Image courtesy of Armagan Elibol et al. (Elibol et al., 2014).

of the range sensor, its value is calculated using the following recursive relation:

$$l_{t,i}^a = l_{t-1,i}^a + \log \frac{p(m_i^a | z_t^a, x_t^a)}{1 - p(m_i^a | z_t^a, x_t^a)}, \tag{35}$$

where $l_{t,i}^a$ is referred to as the *log odds* and is defined as

$$l_{t,i}^a = \log \frac{p(m_i^a | z_{1:t}^a, x_{1:t}^a)}{1 - p(m_i^a | z_{1:t}^a, x_{1:t}^a)}. \tag{36}$$

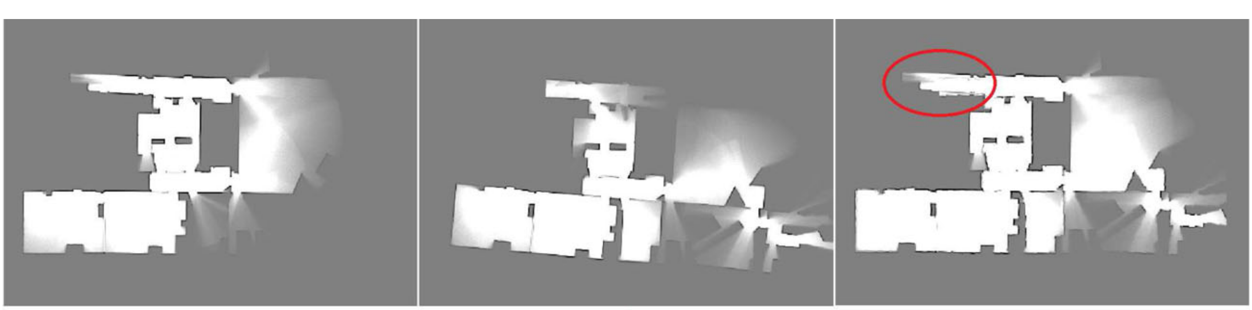
For the occupancy grid maps, after finding the relative transformation between two maps, m^a and m^b, the probabilities are combined to produce the final map. The data received from the transformed map, $m^{'b}$, are akin to a batch of sensor data and should be incorporated by using the additive property of the log odds representation of occupancy, originally defined in Eq. (36):

$$l_{t,i}^{fused} = l_{t,i}^a + l_{t,i}^{'b}, \tag{37}$$

for all $i = 1..N$. The superindices are used to identify the maps, l^a is for m^a and $l^{'b}$ is for $m^{'b}$. l^{fused} represents the fused map, m_{fused}. Once $l_{t,i}^{fused}$ is calculated, the probability of the fused map is recovered from Eq. (36).

It should be mentioned that the accuracy of a multiple-robot SLAM algorithm based on the map merging is dependent on the accuracy of the individual local maps developed by the robots. Thus, to have a consistent global map, it is important to use a reliable and robust SLAM algorithm to generate the local maps. For instance, Figure 23(a) and Figure 23(b) show two local maps developed by two robots. These two maps are generated using a well-known data set presented in (Howard, 2006). This data set, known as Fort AP Hill, is commonly used as a benchmark data set. In these two figures, the maps are generated using FastSLAM1.0 for dense scanning laser ranger mapping. Compared with an implementation of FastSLAM2.0 for dense scanning laser, such as improved grid mapping (Grisetti et al., 2007), FastSLAM1.0 for dense scanning laser ranger mapping requires at least 10 times more particles to produce good quality maps. In other words, if the number of the particles is insufficiently high, the maps will not be of good quality. To generate these maps, 50 particles, a relatively small number, are used. The fused map is shown in Figure 23(c). The area enclosed inside the ellipse shows a mismatch due to the distortion in the original map. This supports the claim that if the original maps, used for map merging, are of insufficient quality, the resulting fused map will also be distorted. This fact is one of the main limitations of map merging. To tackle this problem, it is very important to develop the individual maps of the robots using a consistent and reliable algorithm.

Blanco et al. propose a map merging solution that takes into account the uncertainty and ambiguity arising from the



a b c

Figure 23. A major limitation of map merging is its dependency on good quality maps. If the maps of the robots are distorted, the fused map will also be distorted. a and b show two maps developed by two robots, and c shows the fused map. The area shown by the red ellipse is distorted because of the low quality of the original maps at these areas.

matching maps (Blanco, González-Jiménez, & Fernández-Madrigal, 2013). This method mainly addresses two problems: unknown relative poses (problem a) and the uncertainty of the relative poses (problem b). In this method, a dual representation of local maps is used, where both point and occupancy grid maps are maintained. First, the grid maps are matched; then the corresponding point maps are matched. The point map matching helps to remove false matches. The matching process starts with removing high-frequency noises and extracting the Harris (Harris & Stephens, 1988) or Kanade Lucas Tomasi (KLT) (Shi & Tomasi, 1994) detectors. The transformation is determined from these detectors. Often, multiple candidates exist; the number of the candidates are reduced by a customized RANSAC algorithm, which is based on the sum of Gaussian (SOG) distributions. Using SOG, uncertainty of the detectors is taken into account. Further refinement is performed by matching the point maps. This method is effective in rejecting ambiguity results.

As an important part of the unknown relative poses problem (problem a), knowing the right time for map merging, to avoid failures or false matches, increases the efficiency of map merging. Dinnissen et al. proposes a method that solves this problem (Dinnissen, Givigi, & Schwartz, 2012). In this approach, robots are trained by reinforcement learning to find the best time for map merging. The efficiency of the proposed solution is verified using simulated data sets. Determining the right time for map merging is a challenging problem, and efficient and thorough training is needed to verify the effectiveness of the proposed solution.

To address the unknown relative pose problem (problem a), Saeedi et al. propose an occupancy grid map merging solution, which finds the relative transformation between the robots by comparing featured sections of the maps (Saeedi, Paull, Trentini, & Li, 2011a). These sections are smoothed, and their suitability for map merging is determined by a histogram filter. This approach is tested with three robots in different indoor environments.

An algorithm based on the artificial intelligence is used in (Saeedi et al., 2011c) to address the unknown relative pose problem (problem a) and the complexity problem (problem f). In this algorithm, map merging is achieved by clustering the occupancy grid maps using neural networks self-organizing maps (SOM) or the Kohonen network (Kohonen, 1987). SOM can reduce the dimensionality of the metric space; therefore, an occupancy grid map with thousands of cells are represented by a reduced number of neurons (or clusters). The clusters represent the spatial geometry of the map in an abstract form. Thus, processing the clusters reduces the computational demand. Then, norms of the clusters are used to find the relative transformation between the maps. Although this approach scales down the maps to clusters and therefore is relatively fast, some useful information may be lost by clustering the maps.

Similarly, to address the relative poses problem (problem a) for vehicles in outdoor environments, Li et al. use genetic algorithms (GA) to determine when the best match between the two maps occurs (Li, Tsukada, Nashashibi, & Parent, 2014). The experiments demonstrate that the maps have large overlaps, which guarantee success in map merging using GA.

Finding overlaps in the Euclidean space is difficult. Thus, by transforming the map into a different space, certain properties can be exploited to make the problem easier. The Hough transform turns out to have many such properties. For example, line segments, which are common in most structured environments, are modeled as an intensity point in the Hough space. Relative relations of line segments are represented by the distances between intensity points. In addition, since the Hough transform is generated over an angle range of 360°, all views in the Hough space are included and the problem of partial views can be eliminated. In robotic applications, the Hough transform has also been used in scan matching (Censi, Iocchi, & Grisetti, 2005) and localization (Yun, Latt, & Glennon, 1998; Indelman, Nelson, Michael, & Dellaert, 2001; Iocchi & Nardi, 2002; Grisetti, Iocchi, & Nardi, 2002).

There are a variety of methods to extract geometric information like line segments from maps or laser ranger measurements (Nguyen, Fox, Martinellii, Tomatis, & Siegwart, 2005; Harati & Siegwart, 2007). The Hough transform (Davies, 2005) and the Radon transform (Radon, 1986) are two classical methods for line extraction that have different applications. The results of the Hough and the Radon transforms on a map are almost identical, although the approaches of these two methods are different. The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle, while the Hough transform is a parametric model of each nonbackground point. Other methods like split-merge are fast and effective for line extracting.

Carpin proposes map-merging algorithm using the Hough spectrum (Carpin, 2008). This algorithm addresses the unknown relative poses problem (problem a). The Hough spectrum approach is based on correlation of Hough images, and there is no assessment on overlaps between maps. In this method, a spectrum function, originally proposed in (Censi et al., 2005), is defined on the Hough image. The function is applied to both Hough images of maps, and then the correlation of the results is used to find potential rotations. There are multiple candidates that are all processed to find the translation. For each candidate, a translation is calculated, and at the end, only one rotation and translation are accepted. Carrying over all rotation candidates requires more processing time and computational power and makes the algorithm slow. For each rotation, the translation is found using correlation of projections of the Hough images along x and y directions. For example, if projections of the Hough image of map m_1 are shown by P_{x_1} and P_{y_1}

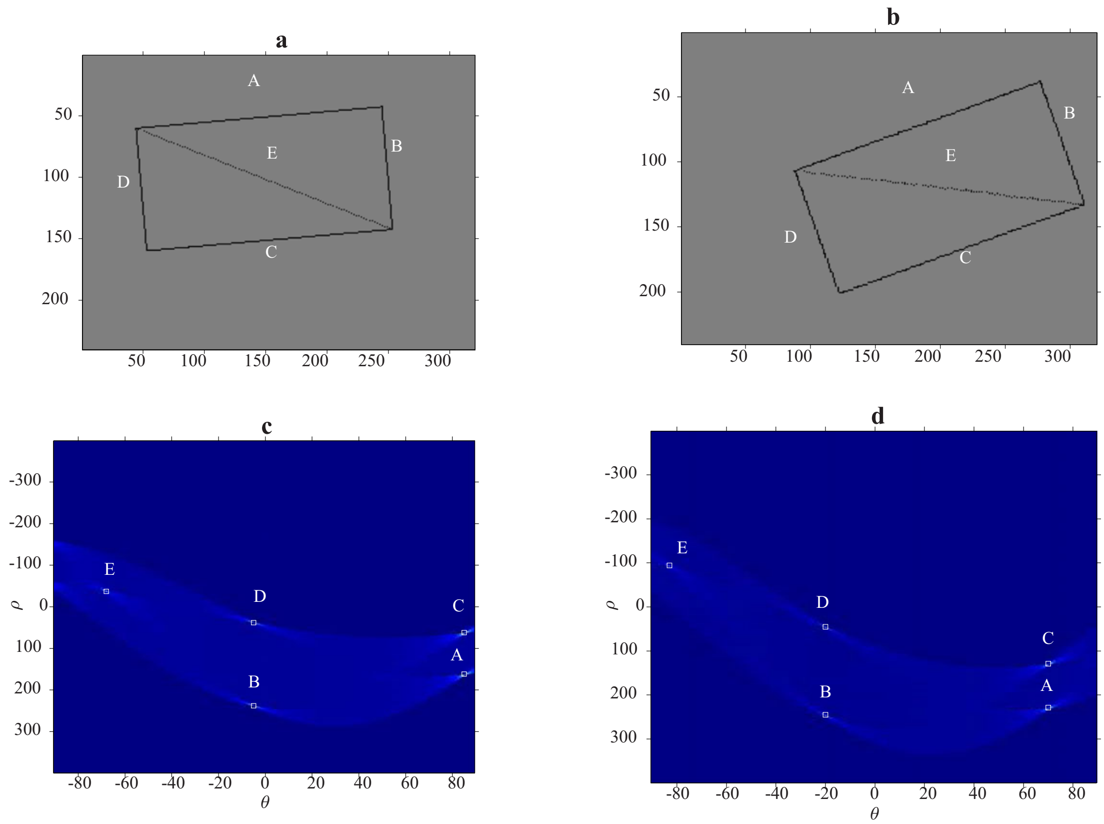


Figure 24. Effect of the transformation of maps on the Hough images. a) An example occupancy grid map. b) The map is rotated 15° and translated 50 cells along both axes. c) Hough image of the map over 180 degrees. Labeled points correspond to the line segments. d) Hough image of the transformed map over 180 degrees (Saeedi et al., 2014b).

and for the rotated map m_2 by P_{x_2} and P_{y_2}, then the peak of correlation of P_{x_1} and P_{x_2} results in the translation along x and the peak of correlation of P_{y_1} and P_{y_2} generates the translation along y. This approach is effective if projections have enough similar patterns, and this similarity happens only when there is a significant amount of overlap between maps. Therefore, finding the translation fails in maps with less overlaps.

Saeedi et al. propose a map merging for occupancy grid maps in the Hough space, using Hough peak matching (Saeedi et al., 2014b, 2012b). The proposed algorithm addresses the unknown relative poses problem (problem a) and the uncertainty of the relative poses (problem b). In this method, map fusion is achieved by transforming individual maps into the Hough space where they are represented in an abstract form. Using peak points of the Hough images and properties of the Hough transform, common regions in the maps are determined. These regions are then used to calculate the unknown transformation between the maps. The rotation in the Euclidean space is a nonlinear function, composed of trigonometric functions. An interesting property of the Hough space is that the rotation in the Hough space is a linear function. Figure 24 shows this property. Figure 24(a) shows a map with five line segments. Figure 24(b) shows the same map rotated 15° and translated 50 cells along both axes. Figure 24(c) and Figure 24(d) show the Hough images of the two maps. Peak points are marked with white squares, and each peak point corresponds to a line segment. The rotation in the Euclidean space is equivalent to a shift in the Hough images.

In indoor and structured environments, it is very common to have multiple hypothesis for the rotation, due to the shape of the environment. In the proposed algorithm, the true hypothesis is selected using cross correlation of the Hough images. The simulated and real-world experiments show the effectiveness of the proposed algorithm.

As reviewed in the previous sections, the concept of map merging also applies to feature maps, where overlaps need to be identified (Zhou & Roumeliotis, 2006; Konolige, Fox, Limketkai, Ko, & Stewart, 2003). Once two feature maps are aligned by a transformation, they must be merged to generate a global map. Merging maps can be done by

Figure 25. Two partial maps of the same environment with their probabilistic GVDs. There is a displacement between the maps. a) PGVD of the first map. b) PGVD of the second map (Saeedi et al., 2014a).

Figure 26. Two edges from Figure 25-a a) edge number 3. b) edge number 7. The grayscale intensity of each cell in these edges represents the probability of that cell in the PGVD (the probability that it is a valid GVD cell) (Saeedi et al., 2014a).

copying all features from one map to another; however, because of noise and uncertainty of the position of the features, duplicate features might exist. These duplicates should be merged to avoid incorrect mapping. To do this, if the distance between two features from aligned maps is less than a predefined threshold, then they should be merged and considered as one feature. This has been shown for two features in Figure 9.

6.9. Topological Approaches

When multiple robots are supposed to explore and map an unknown environment cooperatively, there is a need to provide a logical infrastructure so that robots can share their spatial perception and decide about how to use the shared knowledge. If the salient information from maps is extracted and shared among robots, the speed and accuracy of the mutual perception will improve and communication channels will not be burdened with large amounts of unprocessed data. Topological approaches to multiple-robot SLAM is often used in map merging, which was explained in the previous section.

Topological approaches have also many applications in map merging. Most topological methods address two main problems: unknown relative poses (problem a) and complexity (problem f). For instance, multirobot SLAM based on topological map merging using both structural and geometrical characteristics of the Voronoi graph is proposed in (Huang & Beevers, 2005). The assumption in this work is that a robot will be able to recognize areas of the map that correspond to vertices. In this case, the topological map is built on the occupied space as opposed to the free space. The method in (Huang & Beevers, 2005) is claimed to be fast; however, a limitation is that the maps are not updated. Chang et al. proposes a hybrid topological/metric multiple-robot SLAM method (Chang, Lee, Hu, & Lu, 2007), in which graphlike topological representation of the maps is used to find the relative alignment of the maps. These methods, as reported, need to be improved for scalability.

Erinc et al. propose a multiple-robot topological map merging for appearance maps (Erinc & Carpin, 2014). In their approach, appearance maps are matched by maximizing an algebraic connectivity metric, defined for the merged graph. The same metric is used to evaluate the quality of the merged map. The main novelty of their work is putting forward the algebraic metric to assess the quality of the map. This is very important as in the research community, measuring the performance of the appearance maps is still an open problem. Another important issue, which is left as future work, is that the redundant images in both original maps are not removed from the merged map.

A probabilistic topological representation is proposed in (Saeedi, Paull, Trentini, Seto, & Li, 2014a, 2012a). In addition to problems a and f, unknown relative pose and complexity, the proposed algorithm addresses the uncertainty of the relative poses (problem b). The key idea is to extend the generalized Voronoi diagram to include probabilistic information of the occupancy grid maps. The extended diagram, called probabilistic generalized Voronoi diagram (PGVD), is used to preferentially match parts of the maps that are more certain. Also to address the uncertainty of the relative poses, an algorithm is proposed to transform a map with an uncertain transformation matrix. This algorithm is important, since it helps to propagate the uncertainty of the transformation matrix to the transformed map.

The concept of probabilistic GVD is demonstrated in Figure 25(a)-(b). Figure 25(a) and Figure 25(b) show the

Table IV. Comparison of common map-merging algorithms.

No	Method	Pros and Cons
1	Adaptive Random Walk (Birk & Carpin, 2006)	• **Pros**: easy to implement, works in unstructured settings • **Cons**: slow
2	Hough Peak matching (Saeedi et al., 2014b)	• **Pros**: fast, works with minimal overlaps • **Cons**: requires preprocessing (Hough transform), works in structured settings
3	Probabilistic GVD (Saeedi et al., 2014a)	• **Pros**: fast, works with minimal overlaps • **Cons**: requires preprocessing (extracting GVD)
4	Hough Spectrum (Carpin, 2008)	• **Pros**: fast, requires preprocessing (Hough transform) • **Cons**: requires large overlaps
5	Fourier-Hough transform (Chunhavittayatera, Chitsobhuk, & Tongprasert, 2006)	• **Pros**: fast, requires preprocessing (Hough transform) • **Cons**: requires very large overlaps, less accurate

probabilistic GVDs for each map where short edges have been removed. The edges of each map are marked with numbers. The first map has seven edges, and the second map has eight edges. Figure 26(a) and Figure 26(b) show two enlarged edges of the first map, number 3 and number 7. The grayscale intensity of each cell in these edges represents the probability of that cell in the PGVD (the probability that it is a valid GVD cell). Edge number 3 is located in the area of the original map which has high certainty, so the probability of the cells of this edge being in the GVD is high. However, edge number 7 is located at the end corner of the map, which is less certain. Therefore, the probability of the cells in this edge is lower. The proposed algorithm has a few advantages: first, it reflects the probabilities of the occupancy grid maps to the GVDs of the maps. Therefore, the topological map is probabilistic. Second, it propagates the uncertainty of the relative transformation to the map, and finally, based on the experiments, the complexity of the algorithm is low.

Table IV compares pros and cons of a few map-merging algorithms, including topological and nontopological algorithms, for occupancy grid map merging in 2D space. To the best of the authors' knowledge, there is no solution in the literature addressing map merging in 3D space. An important bottleneck in 3D map merging is dealing with computational limitations.

6.10. Other Issues in Multiple-Robot SLAM

In this section, other challenges in multiple-robot SLAM, such as communication issues and performance measure, with available solutions are explained.

6.10.1. Communications Issues

Communications problem, such as blackouts, delays, and out-of-sequence packets, are major problems that affect the performance of the real-time multiple-robot SLAM algorithms. Two important communication-related issues are *cyclic update* and *out-of-sequence measurements*. Cyclic update, or overconfidence, happens when robots use a measurement repeatedly. This happens when one or multiple robots use the same information more than once. If a measurement arrives at its destination in the wrong order, for example, if it is delayed, it is an out-of-sequence measurement. This happens during buffering or communications latency problems. Dealing with these issues is an important part of multiple-robot SLAM. Specifically, environments such as underwater, low bandwidth, and low-speed communications impose limitations on the accuracy and scalability of a team-based operation.

Leung et al. studied communication problems in multiple-robot localization (Leung, Barfoot, & HT Liu, 2010) and multiple-robot SLAM (Leung et al., 2012). In their approach, a centralized-equivalent estimate is obtained by all robots in a decentralized manner, assuming the network is never fully connected. A centralized-equivalent estimate means that a robot can find an estimate that is equivalent to the estimate generated by a centralized estimator. This solution enforces delay on the state estimation; however, by applying the Markov property at certain times, a robot does not need to keep track of what other robots know. Lazaro et al. use condensed measurements to exchange map information between the robots (Lazaro, Paz, Pinies, Castellanos, & Grisetti, 2013). The condensed measurements address communications and computational problems (problems g and f), with a very little decrease in accuracy of the results. In (Aragues, Cortes, & Sagues, 2011), an algorithm for merging feature-based maps with limited communication is introduced. Other aspects of communication issues, such as out-of-sequence measurements, have been studied in (Bar-Shalom, 2002; Howard, Mataric, & Sukhatme, 2003; Bar-Shalom, Chen, & Mallick, 2004; Hollinger & Singh, 2012).

6.10.2. Performance Measure

Evaluating the accuracy of SLAM algorithms has always been a challenging problem in the SLAM research community. There are few works that have studied and developed a performance measure for single-robot SLAM (Burgard et al., 2009; Kümmerle et al., 2009; Schwertfeger & Birk, 2013). The most intuitive approaches are comparing the robot trajectory with the ground-truth trajectory, or comparing the robot-developed map with the ground-truth map. These approaches have limitations. For instance, in real-world environments, the ground-truth trajectory requires a motion capture system. Similarly, acquiring the ground-truth map requires using cartography techniques or having many accurate measurements from the environment.

Schwertfeger et al. propose a performance measure to compare a robot-built map with a ground-truth map (Schwertfeger & Birk, 2013). To calculate the performance measure, the topological structures of both maps are extracted using generalized Voronoi diagram; then, these graphs are matched to determine the similarity of the maps. The matching can be based on specific attributes of the graphs, such as the number of the edges connected to a vertex or the angles between the edges of a vertex. Also the matching can be based on the structural similarity of the occupied cells in the vicinity of vertices. This approach is based on the abstract and topological structures of the maps; therefore, the matching is relatively fast. Moreover, the proposed matching techniques are invariant to the initial orientation of the maps.

If the ground-truth information, such as blueprints of the environment, location of features, or the true pose of the robots through a motion capture system, is available, the results of SLAM can be evaluated and verified; however, this type of information is not available in all environments. When no ground-truth information is available, the self-consistency of the global map can be used to evaluate a SLAM algorithm. For instance, (Kümmerle et al., 2009) propose a method that relies on the relative geometric relations between the poses of the robot. Although this approach does not require the blueprint or ground-truth trajectory, it needs some manual work, carried out by a human, to determine the topology of the environment.

In multirobot autonomous applications in which the localization of the robots is based on multiple-robot SLAM, each robot on the team relies on the global map to perform autonomous tasks; thus, the accuracy and consistency of the global map affects the decision of the robots and therefore the success of the mission. Few studies have been done on measuring the performance for multiple-robot SLAM, and it is considered as a challenging and unsolved problem. For appearance-based map merging, Erinc et al. propose a metric that measures how the local maps are interlaced in the merged map (Erinc & Carpin, 2014). A similar metric is developed for occupancy grid map merging, in which the performance of map merging is based on counting the number of the known cells in the merged map that are either free or occupied in both local maps (Birk & Carpin, 2006). These metrics do not rely on the ground-truth information and may represent invalid results, especially when there is little overlap between the maps.

6.10.3. 3D SLAM

As mentioned in Section 4.8, 3D SLAM is the ultimate goal for robotic perception. However, because of the complexity of multiple-robot SLAM in 3D, most papers have focused on multiple-robot SLAM in 2D. In some of the papers, because there is no change in the altitude and roll and pitch angles of the robots, multiple-robot SLAM is performed using 2D SLAM, but the mapping is in 3D space. For instance, 2D SLAM with 3D mapping is performed in (Tobata et al., 2012). Moreover, some papers assume that the world is rectilinear and the scanning laser ranger measurements are projected onto 2D space, in which 2D coordinates and the heading angle are calculated (Kim et al., 2010). Among other works, the following papers, which were reviewed in the previous sections, perform 3D SLAM, recovering full position and orientation: (Michael et al., 2012; Vidal-Calleja, Berger, Sola, & Lacroix, 2011; McDonald, Kaess, Cadena, Neira, & Leonard, 2013). The last paper is about multisession SLAM, which is different from multiple-robot SLAM but shares similar concepts.

6.11. Summary of the Algorithms

Table V lists the presented algorithms for multiple-robot SLAM. Each algorithm in the table addresses one or more of the following problems explained in Table III: (a) relative poses of robots, (b) uncertainty of the relative poses, (c) updating maps and poses, (d) line-of-sight observations, (e) closing loops, (f) complexity, (g) communications, (h) heterogeneous vehicles and sensors, (i) synchronization, and (j) performance measure.

7. MULTIROBOT TEST BEDS AND DATA SETS

Developing a test bed and collecting experimental data often takes a lot of time and resources. The collected data sets and developed test beds can assist researchers to save their effort to develop efficient algorithms. Moreover, with a known data set, researchers can compare their results and efficiency of the algorithms. Michael et al. present an experimental test bed for large teams of multiple robots (Michael, Fink, & Kumar, 2008). The test bed is composed of targets and ground and aerial robots, called Scarab and Khepri robots, respectively. These robots can be used for applications such as formation control and perception. Rocco et al. developed a fleet of differential-drive wheeled robots, called SAETTA, for indoor applications (Di Rocco, La Gala, & Ulivi, 2013). SAETTA robots have several key advantages, such as being low cost, easy to replicate,

Table V. Multiple-robot SLAM algorithms. Each algorithm addresses one or more of the following problems explained in Table III: (a) relative poses of robots, (b) uncertainty of the relative poses, (c) updating maps and poses, (d) line-of-sight observations, (e) closing loops, (f) complexity, (g) communications, (h) heterogeneous vehicles and sensors, (i) synchronization, and (j) performance measure.

Algorithm	Addressed Problems	Algorithm	Addressed Problems
EKF-SLAM			
(Fenwick et al., 2002)	c	(Madhavan et al., 2004)	c
(Mourikis & Roumeliotis, 2006)	d, j	(Zhou & Roumeliotis, 2006)	a, b, c, f
(Jafri & Chellali, 2013)	a, d		
EIF-SLAM			
(Nettleton et al., 2000)	c, f	(Thrun & Liu, 2005)	a, c, f
PF-SLAM			
(Thrun, 2001)	c, f	(Howard, 2006)	a, c, e
(Howard et al., 2006a)	a, c, e, h	(Carlone et al., 2010)	b, d, g
(Gil et al., 2010)	c		
GraphSLAM			
(Andersson & Nygards, 2008)	a, b, c	(Takeuchi & Tsubouchi, 2008)	c, f
(Kim et al., 2010)	a, c, h	(Cunningham et al., 2010)	c, g
(Nagatani et al., 2011)	a, c	(Cunningham et al., 2012)	a, c
(Pillonetto et al., 2013)	c	(Indelman et al., 2014)	a, b
Set-based SLAM			
(Moratuwage, Wang, Rao, Senarathne, & Wang, 2014)	c, e		
Sub-map Matching			
(Williams et al., 2002b)	a, c, h	(Vidal-Calleja et al., 2011)	a, c, h
Cooperative Positioning System			
(Kurazume et al., 1994)	d	(Dellaert et al., 2003)	d
(Tobata et al., 2012)	d, c, h		
Manifold Representation			
(Howard et al., 2006b)	a, d, e		
Map Merging			
(Konolige et al., 2003)	a	(Ko et al., 2003)	a
(Zhou & Roumeliotis, 2006)	a, b, c, f	(Fox et al., 2006)	a, c
(Birk & Carpin, 2006)	a, j	(Carpin, 2008)	a
(Saeedi et al., 2011a)	a	(Saeedi et al., 2011c)	a, f
(Dinnissen et al., 2012)	a	(Michael et al., 2012)	a, h
(Blanco et al., 2013)	a, b	(Li et al., 2014)	a
(Elibol et al., 2014)	a, j	(Saeedi et al., 2014b)	a, b
Topological Approaches			
(Huang & Beevers, 2005)	a, f	(Chang et al., 2007)	a
(Erinc & Carpin, 2014)	a, j	(Saeedi et al., 2014a)	a, b, f
Communication Issues			
(Bar-Shalom, 2002)	g	(Howard et al., 2003)	d, g
(Bar-Shalom et al., 2004)	g	(Hollinger & Singh, 2012)	g
(Leung et al., 2010)	g	(Leung et al., 2012)	g
(Lazaro et al., 2013)	f, g	(Leung et al., 2011)	i

able to communicate on different wireless channels, and easy to program. Robotic simulators can also be used to develop and test the multiple-robot SLAM algorithms efficiently. Robot operating system (ROS) (Willow Garage, 2006) and Gazebo (Koenig & Howard, 2004), Microsoft Robotics Developer Studio (MRDS, 2006), virtual robot experimentation platform (V-REP) (Coppelia Robotics, 2013) are just a few examples of such simulators.

There are many data sets for single-robot SLAM (Tong, Gingras, Larose, Barfoot, & Dupuis, 2013; Smith, Baldwin, Churchill, Paul, & Newman, 2009; Bastian Steder, 2013; Lee, Achtelik, Fraundorfer, Pollefeys, & Siegwart, 2010); however, for multiple-robot SLAM, and especially for heterogeneous robots, there are only a few data sets. Leung et al. present a data set collected by a team of ground robots for cooperative localization and mapping (Leung et al., 2011). The data set includes odometry and images from monocular cameras, observing artificial landmarks. The robotics data set repository (RADISH) hosts 41 well-known single-robot and multiple-robot data sets with different sensors (Howard & Roy, 2003), but there is no data set in RADISH, which includes both flying and ground robots. Majdik et al. present a data set that consists of image data captured with a small quadrotor flying on the streets of Zurich, along a path of 2-km-long (Majdik, Albers-Schoenberg, & Scaramuzza, 2013). The data set includes aerial quadrotor images, ground-level Google Street View images, ground-truth confusion matrix, and GPS data.

8. CHALLENGES, FUTURE DIRECTIONS, AND APPLICATIONS

In this section, two topics are presented. First, challenges and future directions for multiple-robot SLAM are listed and explained, and then some general recommendations for various practical applications are presented.

8.1. Challenges

Mobile robots will have various applications in future and real-world situations. For the listed problems in Table III, the reviewed algorithms and solutions are not bulletproof. These solutions require more research to improve efficiency and reliability of the robots in daily and regular applications. In addition to the reviewed problems, other potential challenges and research directions are listed as follows:

- **Large-Scale Environments:** Most of the current research focuses on relatively small environments, such as indoor or urban environments. Extending the current research to extensive environments, such as exploring and mapping oceans and large and crowded airports, are very interesting and challenging problems. Limited processing and power resources, coordination of the robots, and communication issues are key challenges in these environments.
- **Dynamic Environments:** Performing SLAM with one or more robots in dynamic environments is one of the emerging fields in robotic perception. In these environments, SLAM is combined with detection and tracking (Wang, Thorpe, & Thrun, 2003). Mapping and filtering in such environments are performed using the RFS (Mullane et al., 2013). Very few works have been done for multiple-robot SLAM in dynamic environments. For instance, Moratuwage et al. present an algorithm that tracks both static and dynamic features (Moratuwage, Vo, & Wang, 2013), (Moratuwage et al., 2014). RFS is a new emerging field in robotic perception and has other applications, such as multiple-object detection (Granstrom, Lundquist, Gustafsson, & Orguner, 2014).
- **Human-Robot Interaction:** The assumption of static environments simplifies the multiple-robot SLAM problem; however, this simplification creates a gap between research and real-world applications. A critical issue in dynamic and real-world applications is recognizing humans, either intruders or known people. This problem is related to human interaction with robots and is far more complicated than identifying humans as targets or obstacles.
- **Semantic SLAM:** To enable robots to reason and plan their actions, new algorithms are needed by which robots can localize and map their environments using semantic concepts rather than a set of unrelated point clouds or features. To move in this direction, robots should be able to learn actively. Creating semantic databases of objects is the first step to address this problem. A team of robots can perform semantic SLAM more efficiently, because they can argue about the validity of their understanding.
- **Multisession Mapping:** Aligning partial maps of an environment, developed by a robot at different times, is referred to as multisession mapping. Multisession mapping is closely related to SLAM in dynamic environments and is used to detect various changes in the environment (McDonald et al., 2013). Another new research field in multiple-robot SLAM is multisession mapping with multiple robots, which requires dealing with the challenges in both fields.
- **Agent Scalability:** Dedicating resources to manage and coordinate large number of robots is another research direction. With a large team, missions can be done faster, but processing and making use of all information, require a robust and efficient design.
- **Dispatch Preparation:** In practical applications, preparing and dispatching robots takes a lot of time and resources. For instance, just synchronizing robots' clocks requires very careful and frequent checkups. Designing an optimal approach for robots' preparation, for instance, a proper check list, is an important step that has very high impact on the efficiency and success of the robotic missions.

Table VI. Application of Multiple-Robot SLAM.

Application	Description	Reference
Exploration, Intruder Detection	2D distributed particle filtering using 80 heterogeneous ground robots	(Howard et al., 2006a)
Search and Rescue	3D GraphSLAM for cooperative mapping by multiple ground robots	(Nagatani et al., 2011)
Disaster Management	A ground robot and a quadrotor cooperatively map a multifloor earthquake-damaged building	(Michael et al., 2012)
Exploration and Mapping	A team of 14 robots explore and map unknown environments with little human intervention	(Olson et al., 2013)
Underwater Mapping	Large area visual mapping in underwater environments using multiple underwater robots	(Elibol et al., 2014)

- **Practical Applications:** There are many practical multi-robot applications that require special hardware and software design. Each of these applications is a potential future research direction. A nonexhaustive list of these applications includes: mine explorations, service robotics, planetary exploration, search and rescue, military operation, border patrol, surveillance and detection, wildlife monitoring, nuclear and biohazard discovery, assembly and construction, load transport, structural health monitoring, and sensor networks.

8.2. Recommendations for Applications

In GPS-denied environments, in which multiple robotic agents are performing autonomous tasks, coordination and cooperation of the agents are critical requirements. In such applications, multiple-robot SLAM plays a key role in providing perception for the robots. In these environments, the choice of the SLAM algorithm depends on many parameters, including the cost of the hardware, the availability of the communication channels, and the importance of the task in relation to the human and environment safety. Table VI summarizes some of the applications of multiple-robot SLAM, which were reviewed in this paper. In this section, some general recommendations for various applications are presented.

Defense applications are among the most important applications in which the success of the missions rely on the accuracy and precision of the localization and mapping techniques. In these applications, there is no room for failure; thus, redundancy at all levels is required. Performing view-based, appearance-based, and feature-based SLAM in parallel by all or a group of agents will help to ensure better estimates at the cost of equipping the robots with a variety of sensors and faster processing units. One of the important bottlenecks of multiple-robot SLAM is the lack of information about the relative poses of the robots. In some of the defense applications, a solution for this problem is acquiring GPS coordinates prior to entry to the GPS-denied environment. For instance, if 10 robots are intended to enter a large building, it is a plausible solution to start the mission by acquiring GPS signals first. This simple solution provides the robots with initial estimates about the relative poses, which can be refined using the various techniques explained in this work. In addition, in defense applications, communication issues can be alleviated by setting up ad hoc networks where the wireless access points can move and relocate in such a way that all agents have access to communication channels. Flying robots, such as quadrotors, are well suited for these applications. These robots are highly maneuverable, can fly over obstacles, and have access to vantage points that ground robots do not. Accompanying the flying robots with a fleet of ground robots can help the flying agents with reliable and fast fuel or battery recharging stations. Also small flying robots with limited processing units can only record data and deliver them to well-equipped ground robots by directly contacting them, without using the communication channels. Such a team of heterogeneous robots could prove to be very efficient in search and rescue operations.

Border patrol is another application that requires high coordination among the robots to ensure a comprehensive coverage of the borderlines. Again a team of ground and aerial robots can perform the mission more efficiently. In these applications, the flying robots usually have access to the GPS signals; therefore, flying robots do not depend on SLAM. But if the flying robots do not have access to GPS, the best option is performing feature-based or appearance-based SLAM with vision sensors, combined with inertial systems. If the robots fly at low altitude, adding a laser ranger and performing view-based SLAM can provide volumetric information about the environment. Depending on the type of the environment, ground robots will face different challenges. In open areas such as deserts, there is no structured setting; therefore, view-based SLAM, which highly depends on the presence of the objects would fail. The best option for these environments is using feature-based SLAM. In structured environment or among the woods, feature-based, view-based, or appearance-based SLAM are all efficient approaches.

Another important application of multiple-robot SLAM is mining, especially in horizontal mine shafts. View-based SLAM using scanning laser rangers is the best choice

Figure 27. (left) NASA's ARES to be deployed on Mars in 2016, image courtesy of NASA.[1] (right) NASA's multiple-robot extraterrestrial exploration team, image courtesy of NASA.[2]

for these environments, as there is not sufficient light for vision-based algorithms. A very useful application in mining is mapping the distribution and leakage points of hazardous gases such carbon dioxide or methane. It is also possible to perform SLAM only using specific sensors that can detect these types of gases, with the assumption that the leakage is stationary or has known or predictable flow patterns.

Extraterrestrial and planetary explorations are other interesting applications where there is no GPS signal available. In some environments, such as the Moon, due to the lack of air, only ground robots can be deployed. In other environments, such as Mars, unmanned aerial vehicles such as gliders or quadrotors can be deployed. For these environments, feature-based or appearance-based SLAM are the best solutions. Because of lack of various features, line-of-sight observation can greatly improve the localization. CPS (Tobata et al., 2012) is a suitable solution for these environments. Figure 27 shows NASA's Aerial Regional-Scale Environmental Survey of Mars (ARES) (left). While at the moment Curiosity, NASA's ground robot, is navigating on Mars, ARES will be able to fly in Mars's atmosphere by 2016, which opens a new field in extraterrestrial heterogeneous multirobot systems. Deploying a fleet of mining robots, called Swarmie robots, to explore Mars and other planets is NASA's other multirobot extraterrestrial mission. Figure 27 (right) shows four of the Swarmie robots.

In underwater environments, feature-based SLAM using vision sensors is limited to clear waters. View-based SLAM using SONAR sensors such as side-scan sonar is the best approach for many underwater applications. A major issue in underwater environments is the limited communication bandwidth, with the maximum rate of 30 kbps in optimal conditions (Paull et al., 2014). In low-rate communication channels, it is recommended to share processed data such as the maps or poses of the robots. Compared with sharing raw data, sharing processed data has the advantage that it does not need higher bandwidth, and also, it does not require availability of the channels at all times. However, its main drawback is overconfidence, due to the frequent use of one measurement. Depending on the application, special precaution should be taken to avoid overconfidence. In general, in underwater environments, the performance of SLAM algorithms are lower compared with ground or aerial environments due to inherent limitations in these environments; thus, using line-of-sight observation, through acoustic measurement, can greatly improve the performance of multiple-robot SLAM. To the best of the authors' knowledge, in the literature, there is no record of multiple-robot appearance-based of polygon-based SLAM with sharing raw sensor data among the robots.

In nuclear and biohazard discovery applications, there is another source of information that can help the robots to perform localization and mapping more efficiently. This secondary information source is the distribution or leakage sources of the nuclear or biohazard materials. Nuclear and biohazard propagation have known patterns; therefore, robots can rely on these propagation models to perform SLAM. Specially, nuclear leakage is usually very static and does not change very fast over a short period of time. For these applications, all types of the SLAM algorithms are suitable; however, if only the sources of the contamination are desired, multiple-robot feature-based SLAM using nuclear or biohazard detectors is the best choice, where the contamination sources act as landmarks. If the distribution of the contamination is desired, the best choice is developing a contamination grid map, similar to the concept of the occupancy grid map. Similar concept applies

[1] http://marsairplane.larc.nasa.gov/multimedia.html
[2] http://www.nasa.gov/content/meet-the-swarmies-robotics-answer-to-bugs

to the contamination mapping in relatively stationary underwater environments such as small lakes and lagoons.

For disaster management, due to the traversability problems, the best choice is deploying a team of heterogeneous aerial and ground robots, all performing various types of SLAM, such as view-based and feature-based SLAM. One of the challenges is fusing different types of the maps.

Domestic service robots are in direct contact with humans; therefore, very reliable perception is required. Domestic settings are structured; thus, all types of SLAM, including feature-based or polygon-based, may be used. In addition, semantic mapping in domestic settings enables robots to reason, to plan, and to react. To the best of the authors' knowledge, there is no semantic multiple-robot SLAM in the literature.

9. CONCLUSION AND FUTURE WORK

This paper provides a review of the state-of-the-art for multiple-robot systems, with a major focus on multiple-robot SLAM. Various algorithms were reviewed, and motivation, advantage, and disadvantage of each algorithm were briefly described. The paper started with a brief review of single-robot SLAM, emphasizing the algorithms presented after 2005. Other single-robot SLAM algorithms, presented prior to 2005, are introduced in an outstanding textbook in (Thrun et al., 2005). For multiple-robot SLAM, first, 10 major problems were introduced and defined. Then, the literature in relation to these problems was reviewed. For each reference, it was mentioned which problems were addressed. At the end, well-known data sets, simulation tools, future directions, and some general recommendations for various applications were presented.

Multiple-robot SLAM is not as mature as single-robot SLAM, and there is a long way to go and to explore in this field. From a general perspective, most of the current literature focuses on sharing raw data among the robots. While this method is very flexible due to the access to all past measurements, its weakness is the reliance on the availability of delay-free communication channels. In contrast, the approach of sharing processed data among robots, such as map merging, is simple, effective, and resilient to communication delays and blackouts, but may cause overconfidence, due to multiple use of a single measurement. Moreover, it lacks the flexibility of sharing raw data, because original raw data are not available for reprocessing. In addition, its efficiency in 3D SLAM has not been studied in the robotics community thoroughly. The authors believe that a combination of raw and processed data sharing, combining advantages of both methods, will provide the solution for multiple-robot SLAM.

Single-robot SLAM in 2D environments is well established, and most researchers consider it as a solved problem; however, in 3D, and especially when using RGBD cameras, such as Microsoft Kinect or Asus Xtion, the research is still ongoing. Current algorithms, such as KinectFusion, rely on very powerful processing units, which are not available on robotic platforms. Moreover, the data rate of such sensors is very high, about 92 Mbps for QVGA quality (at 30 Hz, each RGB frame is $320 \times 240 \times 3 \times 8$ bits, and each depth frame is $320 \times 240 \times 16$ bits). These limitations impose many new processing- and communication-related challenges on small-sized robotic teams, let alone large teams. For instance, at high rates, many measurements will be out of sequence, which means either the delayed measurement should be discarded or some reprocessing should be performed, as explained for the out-of-sequence measurements.

In future work, the current research will be broadened to include more specific applications and the detailed requirements and criteria for the success of these team-based robotic applications. In addition, the current paper will be extended to include other aspects of multiple-robot autonomous applications, such as path planning, exploration, active localization, and mission planning.

ACKNOWLEDGMENTS

This research is supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Foundation for Innovation (CFI).

REFERENCES

Adams, M., Vo, B.-N., Mahler, R., & Mullane, J. (2014). SLAM gets a PHD: New concepts in map estimation. IEEE Robotics Automation Magazine, 21(2), 26–37.

Andersson, L., & Nygards, J. (2008). C-SAM: Multi-robot SLAM using square root information smoothing. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 2798–2805. Pasadena, CA, USA: IEEE.

Aragues, R., Cortes, J., & Sagues, C. (2011). Distributed consensus algorithms for merging feature-based maps with limited communication. Robotics and Autonomous Systems, 59(3-4), 163–180.

Arai, T., Pagello, E., & Parker, L. (2002). Guest editorial advances in multirobot systems. IEEE Transactions on Robotics and Automation, 18(5), 655–661.

Bailey, T. (2000). Mobile Robot Localization and Mapping in Extensive Outdoor Environments. Ph.D. dissertation, University of Sydney, Sydney, NSW, Australia.

Bar-Shalom, Y. (2002). Update with out-of-sequence measurements in tracking: Exact solution. IEEE Transactions on Aerospace and Electronic Systems, 38, 769–777.

Bar-Shalom, Y., Chen, H., & Mallick, M. (2004). One-step solution for the multistep out-of-sequence-measurement problem in tracking. IEEE Transactions on Aerospace and Electronic Systems, 40, 27–37.

Barca, J. C., & Sekercioglu, Y. A. (2013). Swarm robotics reviewed. Robotica, 31(3), 345–359.

Bastian Steder (2013). Quadrotor 079 Freiburg 3D scan dataset. Retrieved July 19, 2013, from http://ais.informatik.uni-freiburg.de/projects/datasets/quadrotor079/.

Beeson, P., Jong, N. K., & Kuipers, B. (2004). Towards autonomous topological place detection using the extended Voronoi graph. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 4373–4379. New Orleans, LA, USA: IEEE.

Beeson, P., Modayil, J., & Kuipers, B. (2010). Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. The International Journal of Robotics Research, 29(4), 428–459.

Birk, A., & Carpin, S. (2006). Merging occupancy grid maps from multiple robots. Proceedings of the IEEE: Special Issue on Multi-Robot Systems, 94(7), 1384–1387.

Blanco, J. L., González-Jiménez, J., & Fernández-Madrigal, J. A. (2013). A robust, multi-hypothesis approach to matching occupancy grid maps. Robotica, 31(5), 687–701.

Boal, J., Sanchez-Miralles, A., & Arranz, A. (2014). Topological simultaneous localization and mapping: A survey. Robotica, 32, 803–821.

Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual navigation for mobile robots: A survey. Journal of Intelligent and Robotic Systems, 53(3), 263–296.

Bosse, M., & Zlot, R. (2008). Map matching and data association for large-scale two-dimensional laser scan-based SLAM. The International Journal of Robotics Research, 27(6), 667–691.

Bullo, F., Cortés, J., & Martínez, S. (2009). Distributed Control of Robotic Networks. Applied Mathematics Series. Princeton, NJ: Princeton University Press.

Bunke, H. (2000). Graph matching: Theoretical foundations, algorithms, & applications. In International Conference on Vision Interface, pp. 82–88. Montreal, QC, Canada.

Burgard, W., Moors, M., Stachniss, C., & Schneider, F. (2005). Coordinated multi-robot exploration. IEEE Transactions on Robotics, 21(3), 376–386.

Burgard, W., & Schneider, F. (2002). Collaborative exploration of unknown environments with teams of mobile robots. In Dagstuhl, pp. 52–70. Dagstuhl, Germany: Springer Verlag.

Burgard, W., Stachniss, C., Grisetti, G., Steder, B., Kummerle, R., Dornhege, C., Ruhnke, M., Kleiner, A., & Tardos, J. (2009). A comparison of SLAM algorithms based on a graph of relations. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2089–2095. St. Louis, MO, USA: IEEE.

Carlone, L., Ng, M. K., Du, J., Bona, B., & Indri, M. (2010). Rao-Blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 243–249. Anchorage, AK, USA: IEEE.

Carpin, S. (2008). Fast and accurate map merging for multi-robot systems. Autonomous Robots, 25(3), 305–316.

Carpin, S., Birk, A., & Jucikas, V. (2005). On map merging. Robotics and Autonomous Systems, 53(1), 1–14.

Censi, A. (2008). An ICP variant using a point-to-line metric. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 19–25. Pasadena, CA, USA: IEEE.

Censi, A., Iocchi, L., & Grisetti, G. (2005). Scan matching in the Hough domain. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 2739–2744. Barcelona, Spain: IEEE.

Chaimowicz, L., Grocholsky, B., Keller, J., Kumar, V., & Taylor, C. (2004). Experiments in multirobot air-ground coordination. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 4, pp. 4053–4058. New Orleans, LA, USA: IEEE.

Chang, H., Lee, C. S. G., Hu, Y., & Lu, Y. H. (2007). Multi-robot SLAM with topological/metric maps. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1467–1472. San Diego, CA, USA: IEEE.

Chatterjee, A. (2009). Differential evolution tuned fuzzy supervisor adapted extended Kalman filtering for SLAM problems in mobile robots. Robotica, 27(3), 411–423.

Choset, H., & Nagatani, K. (2001). Topological simultaneous localization and mapping (SLAM), toward exact localization without explicit localization. IEEE Transactions on Robotics and Automation, 17(2), 125–137.

Choset, H., Walker, S., Eiamsa-Ard, K., & Burdick, J. (2000). Sensor-based exploration: Incremental construction of the hierarchical generalized Voronoi graph. The International Journal of Robotics Research, 19(2), 125–148.

Chunhavittayatera, S., Chitsobhuk, O., & Tongprasert, K. (2006). Image registration using Hough transform and phase correlation. In The 8th International Conference on Advanced Communication Technology, pp. 973–977. Dublin, Ireland: IEEE.

Coppelia Robotics (2013). Virtual Robot Experimentation Platform (v-rep) 3.0.0. Retrieved November 10, 2013, from http://www.coppeliarobotics.com/.

Cummins, M., & Newman, P. (2011). Aappearance-only SLAM at large scale with FAB-MAP 2.0. The International Journal of Robotics Research, 30(9), 1100–1123.

Cunningham, A., & Dellaert, F. (2012). Large-scale experimental design for decentralized SLAM. In Proceedings of SPIE, Unmanned Systems Technology XIV, volume 8387. Baltimore, MD, USA: SPIE.

Cunningham, A., Paluri, M., & Dellaert, F. (2010). DDF-SAM: Fully distributed SLAM using constrained factor graphs. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3025–3030. Taipei, Taiwan: IEEE.

Cunningham, A., Wurm, K., Burgard, W., & Dellaert, F. (2012). Fully distributed scalable smoothing and mapping with robust multi-robot data association. In International Conference on Robotics and Automation (ICRA), pp. 1093–1100. St. Paul, MN, USA: IEEE.

Curnow, R. (2014). Chrony 1.29.1. Retrieved January 31, 2013, from http://chrony.tuxfamily.org/.

Davies, E. R. (2005). Machine Vision, Theory, Algorithms, Practicalities, 3rd Edition. Amsterdam, The Netherlands: Morgan Kaufmann.

Davison, A. J.& Murray, D. W. (2002). Simultaneous localizationand map-building using active vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 865–880.

Dellaert, F., Alegre, F., & Martinson, E. (2003). Intrinsic localization and mapping with 2 applications: Diffusion mapping and Macro Polo localization. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 2, pp. 2344–2349. Taipei, Taiwan: IEEE.

Derenick, J., Speranzon, A., & Ghrist, R. (2013). Homological sensing for mobile robot localization. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 572–579. Karlsruhe, Germany: IEEE.

Dezert, J., & Bar-Shalom, Y. (1993). Joint probabilistic data association for autonomous navigation. IEEE Transactions on Aerospace and Electronic Systems, 29(4), 1275–1286.

Di Rocco, M., La Gala, F., & Ulivi, G. (2013). Testing multirobot algorithms: SAETTA: A small and cheap mobile unit. IEEE Robotics Automation Magazine, 20(2), 52–62.

Dinnissen, P., Givigi, S., & Schwartz, H. (2012). Map merging of multi-robot SLAM using reinforcement learning. In IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 53–60. Seoul, South Korea: IEEE.

Dryanovski, I., Valenti, R., & Xiao, J. (2013). Fast visual odometry and mapping from RGB-D data. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 2305–2310. Karlsruhe, Germany: IEEE.

Elfes, A. (1990). Occupancy grids: A stochastic spatial representation for active robot perception. In Sixth Conference on Uncertainty in AI, pp. 7–24. Cambridge, MA, USA: AUAI Press.

Eliazar, A., & Parr, R. (2003). DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1135–1142. Acapulco, Mexico: Morgan Kaufmann.

Elibol, A., Kim, J., Gracias, N., & Garcia, R. (2014). Efficient image mosaicing for multi-robot visual underwater mapping. Pattern Recognition Letters, 46(0), 20–26.

Engelhard, N., Endres, F., Hess, J., Sturm, J., & Burgard, W. (2011). Real-time 3D visual SLAM with a hand-held RGB-D camera. In Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum. Vasteras, Sweden.

Erinc, G., & Carpin, S. (2014). Anytime merging of appearance-based maps. Autonomous Robots, 36(3), 241–256.

Estrada, C., Neira, J., & Tardos, J. (2005). Hierarchical SLAM: Real-time accurate mapping of large environments. IEEE Transactions on Robotics, 21(4), 588–596.

Eustice, R., Singh, H., & Leonard, J. (2006). Exactly sparse delayed-state filters for view-based SLAM. IEEE Transactions on Robotics, 22(6), 1100–1114.

Fenwick, J., Newman, P., & Leonard, J. (2002). Cooperative concurrent mapping and localization. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 2, pp. 1810–1817. Washington, DC, USA: IEEE.

Forster, C., Lynen, S., Kneip, L., & Scaramuzza, D. (2013). Collaborative monocular SLAM with multiple micro aerial vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3962–3970. Tokyo, Japan: IEEE.

Fox, D., Burgard, W., Kruppa, H., & Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. Autonomous Robots, 8(3), 325–344.

Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., & Stewart, B. (2006). Distributed multirobot exploration and mapping. Proceedings of the IEEE, 94(7), 1325–1339.

Frank, D., & Michael, K. (2006). Square root SAM: Simultaneous location and mapping via square root information smoothing. The International Journal of Robotics Research, 25(12), 1181–1203.

Fraundorfer, F., Engels, C., & Nister, D. (2007). Topological mapping, localization and navigation using image collections. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3872–3877. San Diego, CA, USA: IEEE.

Fraundorfer, F., & Scaramuzza, D. (2012). Visual odometry: Part II. Matching, robustness, optimization, and applications. IEEE Robotics and Automation Magazine, 19(2), 78–90.

Frese, U., Larsson, P., & Duckett, T. (2005). A multilevel relaxation algorithm for simultaneous localization and mapping. IEEE Transactions on Robotics, 21(2), 196–207.

Friedman, S., Pasula, H., & Fox, D. (2007). Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI), pp. 2109–2114. Hyderabad, India: the International Joint Conferences on Artificial Intelligence (IJCAI).

Gil, A., Reinoso, O., Ballesta, M., & Miguel, J. (2010). Multirobot visual SLAM using a Rao-Blackwellized particle filter. Robotics and Autonomous Systems, 58(1), 68–80.

Gould, J. L. (1990). Honey bee cognition. Cognition, 37, 83–103.

Granstrom, K., Lundquist, C., Gustafsson, F., & Orguner, U. (2014). Random set methods: Estimation of multiple extended objects. IEEE Robotics Automation Magazine, 21(2), 73–82.

Grisetti, G., Iocchi, L., & Nardi, D. (2002). Global Hough localization for mobile robots in polygonal environments. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 1, pp. 353–358. Washington, DC, USA: IEEE.

Grisetti, G., Kümmerle, R., Stachniss, C., Frese, U., & Hertzberg, C. (2010a). Hierarchical optimization on manifolds for online 2D and 3D mapping. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA). Anchorage, AK, USA: IEEE.

Grisetti, G., Kümmerle, R., Stachniss, C., & Burgard, W. (2010b). A tutorial on graph-based SLAM. IEEE Intelligent Transportation Systems Magazine, 2(4), 31–43.

Grisetti, G., Kümmerle, R., Stachniss, C., Frese, U., & Hertzberg, C. (2010c). Hierarchical optimization on manifolds for online 2D and 3D mapping. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA). Anchorage, AK, USA: IEEE.

Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Transactions on Robotics, 23(1), 34–46.

Grisetti, G., Stachniss, C., & Burgard, W. (2009). Nonlinear constraint network optimization for efficient map learning. IEEE Transactions on Intelligent Transportation Systems, 10(3), 428–439.

Grocholsky, B., Bayraktar, S., Kumar, V., Taylor, C. J., & Pappas, G. J. (2004). Synergies in feature localization by air-ground robot teams. In Proceedings of the International Symposium on Experimental Robotics (ISER), volume 21 of Springer Tracts in Advanced Robotics, pp. 352–361. Singapore: Springer.

Guilford, T., Roberts, S., Biro, D., & Rezek, I. (2004). Positional entropy during pigeon homing II: Navigational interpretation of Bayesian latent state models. Journal of Theoretical Biology, 227(1), 25–38.

Guivant, J., & Nebot, E. (2001). Optimization of the simultaneous localization and map-building algorithm and real-time implementation. IEEE Transactions on Robotics and Automation, 17(3), 242–257.

Gutmann, J. & Konolige, K. (1999). Incremental mapping of large cyclic environments. In Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation, pp. 318–325. Monterey, CA, USA: IEEE.

Hahnel, D., Burgard, W., Fox, D., & Thrun, S. (2003a). An effcient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 206–211. Las Vegas, NV, USA: IEEE.

Hahnel, D., Burgard, W., & Thrun, S. (2003b). Learning compact 3D models of indoor and outdoor environments with a mobile robot. Robotics and Autonomous Systems, 44(1), 15–27.

Hahnel, D., Burgard, W., Wegbreit, B., & Thrun, S. (2003c). Towards lazy data association in SLAM. In Proceedings of the 11th International Symposium of Robotics Research (ISRR). Siena, Italy: Springer.

Hao, L., & Nashashibi, F. (2013). Cooperative multi-vehicle localization using split covariance intersection filter. IEEE Intelligent Transportation Systems Magazine, 5(2), 33–44.

Harati, A., & Siegwart, R. (2007). A new approach to segmentation of 2D range scans into linear regions. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2083–2088. San Diego, CA, USA: IEEE.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In Proceedings of Fourth Alvey Vision Conference, pp. 147–151. Manchester, UK: University of Sheffield Printing Unit.

Hawkins, J., & Blakeslee, S. (2004). On Intelligence. Times Books. New York City, NY, USA: Times Books.

Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2010). RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In Proceedings of the International Symposium on Experimental Robotics (ISER). New Delhi and Agra, India: Springer.

Hollinger, G., & Singh, S. (2012). Multirobot coordination with periodic connectivity: Theory and experiments. IEEE Transactions on Robotics, 28(4), 967–973.

Howard, A. (2004). Multi-robot mapping using manifold representations. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 4, pp. 4198–4203. New Orleans, LA, USA: IEEE.

Howard, A. (2006). Multi-robot simultaneous localization and mapping using particle filters. The International Journal of Robotics Research, 25(12), 1243–1256.

Howard, A., Mataric, M., & Sukhatme, G. (2001). Relaxation on a mesh: A formalism for generalized localization. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 2, pp. 1055–1060. Maui, HI, USA: IEEE.

Howard, A., Mataric, M., & Sukhatme, G. (2003). Putting the 'I' in 'team': An ego-centric approach to cooperative localization. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 868–874. Taipei, Taiwan: IEEE.

Howard, A., Parker, L., & Sukhatme, G. (2006a). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. The International Journal of Robotics Research, 25(5-6), 431–447.

Howard, A., & Roy, N. (2003). The robotics data set repository (Radish). Retrieved July 19, 2013, from http://radish.sourceforge.net/.

Howard, A., Sukhatme, G., & Mataric, M. (2006b). Multirobot simultaneous localization and mapping using manifold representations. Proceedings of the IEEE, 94(7), 1360–1369.

Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Fox, D., & Roy, N. (2011). Visual odometry and mapping for autonomous flight using an RGB-D camera. In Proceedings of the International Symposium on Robotics Research. Flagstaff, AZ, USA.

Huang, W., & Beevers, K. (2005). Topological map merging. The International Journal of Robotics Research, 24(8), 601–613.

Hussein, M., Renner, M., Watanabe, M., & Iagnemma, K. (2013). Matching of ground-based LiDAR and aerial image data for mobile robot localization in densely forested environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1432–1437. Tokyo, Japan: IEEE.

Indelman, V., Nelson, E., Michael, N., & Dellaert, F. (2014). Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. In International Conference on Robotics and Automation (ICRA), pp. 593–600. Hong Kong: IEEE.

Iocchi, L., Mastrantuono, D., & Nardi, D. (2001). A probabilistic approach to Hough localization. In Proceedings of the

IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 4, pp. 4250–4255. Seoul, South Korea: IEEE.
Iocchi, L., & Nardi, D. (2002). Hough localization for mobile robots in polygonal environments. Robotics and Autonomous Systems, 40(3), 43–58.
Jafri, S., & Chellali, R. (2013). A distributed multi robot SLAM system for environment learning. In IEEE Workshop on Robotic Intelligence In Informationally Structured Space (RiiSS), pp. 82–88. Singapore: IEEE.
Jaulin, L. (2009). A nonlinear set membership approach for the localization and map building of underwater robots. IEEE Transactions on Robotics, 25(1), 88–98.
Jennings, N., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems, 1(1), 7–38.
Kaess, M., Ranganathan, A., & Dellaert, F. (2007). iSAM: Fast incremental smoothing and mapping with efficient data association. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 1670–1677. Roma, Italy: IEEE.
Kawewong, A., Tongprasit, N., Tangruamsub, S., & Hasegawa, O. (2010). Online and incremental appearance-based SLAM in highly dynamic environments. The International Journal of Robotics Research, 30(1), 33–55.
Kelly, A., Amidi, O., Herman, H., Pilarski, T., Stentz, A., Vallidis, N., & Warner, Y. (2006). Toward reliable off road autonomous vehicles operating in challenging environments. The International Journal of Robotics Research, 25, 5–6.
Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., & Teller, S. (2010). Multiple relative pose graphs for robust cooperative mapping. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 3185–3192.
Ko, J., Stewart, B., Fox, D., Konolige, K., & Limketkai, B. (2003). A practical, decision-theoretic approach to multi-robot mapping and exploration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 4, pp. 3232–3238. Las Vegas, NV, USA: IEEE.
Koenig, A., Kessler, J., & Gross, H. M. (2008). A graph matching technique for an appearance-based, visual SLAM-approach using Rao-Blackwellized particle filters. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1576–1581. Nice, France: IEEE.
Koenig, N., & Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2149–2154. Sendai, Japan: IEEE.
Kohlbrecher, S., van Stryk, O., Meyer, J., & Klingauf, U. (2011). A flexible and scalable SLAM system with full 3D motion estimation. In IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), pp. 155–160. Kyoto, Japan: IEEE.
Kohonen, T. (1987). Self-Organizing and Associative Memory. Berlin Heidelberg: Springer-Verlag.
Konolige, K., Fox, D., Limketkai, B., Ko, J., & Stewart, B. (2003). Map merging for distributed robot navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 212–217. Las Vegas, NV, USA: IEEE.
Kuipers, B. & Byun, Y. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Journal of Robotics and Autonomous Systems, 8, 47–63.
Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). g2o: A general framework for graph optimization. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 3607–3613. Shanghai, China: IEEE.
Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., & Kleiner, A. (2009). On measuring the accuracy of SLAM algorithms. Autonomous Robots, 27(4), 387–407.
Kurazume, R., Nagata, S., & Hirose, S. (1994). Cooperative positioning with multiple robots. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 1250–1257.
Lazaro, M., Paz, L., Pinies, P., Castellanos, J., & Grisetti, G. (2013). Multi-robot SLAM using condensed measurements. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1069–1076. Tokyo, Japan: IEEE.
Lee, C. S., Clark, D., & Salvi, J. (2012). SLAM with single cluster PHD filters. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 2096–2101. Saint Paul, MN, USA: IEEE.
Lee, G. H., Achtelik, M., Fraundorfer, F., Pollefeys, M., & Siegwart, R. (2010). A benchmarking tool for MAV visual pose estimation. In Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on, pp. 1541–1546. Singapore: IEEE.
Leung, K., Barfoot, T., & HT Liu, H. (2010). Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach. IEEE Transactions on Robotics, 26, 62–77.
Leung, K., Barfoot, T., & HT Liu, H. (2012). Decentralized cooperative SLAM for sparsely-communicating robot networks: A centralized-equivalent approach. Journal of Intelligent and Robotic Systems, 66, 321–342.
Leung, K., Halpern, Y., Barfoot, T., & HT Liu, H. (2011). The UTIAS multi-robot cooperative localization and mapping dataset. The International Journal of Robotics Research, 30(8), 969–974.
Leung, K., Inostroza, F., & Adams, M. (2014). Evaluating set measurement likelihoods in random-finite-set SLAM. In Proceedings of the International Conference on Information Fusion (FUSION), pp. 1–8. Salamanca, Spain: IEEE.
Leung, K. Y. K. (2012). Cooperative Localization and Mapping in Sparsely-Communicating Robot Networks. Ph.D. dissertation, University of Toronto, Toronto, ON, Canada.

Li, H., Tsukada, M., Nashashibi, F., & Parent, M. (2014). Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging. IEEE Transactions on Intelligent Transportation Systems, 15(5), 2089–2100.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. Autonomous Robots, 4(4), 333–349.

Lu, F., & Milios, E. E. (1994). Robot pose estimation in unknown environments by matching 2D range scans. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 935–938. Seattle, WA, USA: IEEE.

Madhavan, R., Fregene, K., & Parker, L. (2004). Distributed cooperative outdoor multirobot localization and mapping. Autonomous Robots, 17(1), 23–39.

Mahler, R. P. S. (2007). Statistical Multisource-Multitarget Information Fusion. Artech House Information Warfare Library. Norwood, MA, USA: Artech House Publishers.

Majdik, A., Albers-Schoenberg, Y., & Scaramuzza, D. (2013). Online dataset for air-ground matching of images. Retrieved December 1, 2013, from http://rpg.ifi.uzh.ch/data/air-ground-data.tar.gz.

McDonald, J., Kaess, M., Cadena, C., Neira, J., & Leonard, J. (2013). Real-time 6-DOF multi-session visual SLAM over large-scale environments. Robotics and Autonomous Systems, 61(10), 1144–1158.

Michael, N., Fink, J., & Kumar, V. (2008). Experimental testbed for large multirobot teams. IEEE Robotics and Automation Magazine, 15(1), 53–61.

Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., & Tadokoro, S. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. Journal of Field Robotics, 29(5), 832–841.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneouls localization and mapping problem. In Proceedings of the AAAI National Conference on Artificial Intelligence. Edmonton, AB, Canada: AAAI.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In International Joint Conference of Artificial Intelligence, pp. 1151–1156. Acapulco, Mexico: Morgan Kaufmann.

Moratuwage, D., Vo, B.-N., & Wang, D. (2013). Collaborative multi-vehicle SLAM with moving object tracking. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 5702–5708. Karlsruhe, Germany: IEEE.

Moratuwage, D., Wang, D., Rao, A., Senarathne, N., & Wang, H. (2014). RFS collaborative multivehicle SLAM: SLAM in dynamic high-clutter environments. IEEE Robotics and Automation Magazine, 21(2), 53–59.

Mourikis, A. I., & Roumeliotis, S. I. (2006). Predicting the performance of cooperative simultaneous localization and mapping (C-SLAM). The International Journal of Robotics Research, 25(12), 1273–1286.

MRDS (2006). Microsoft Robotics Developer Studio. Retrieved January 5, 2014, from http://msdn.microsoft.com/robotics.

Mullane, J., Vo, B.-N., Adams, M., & Vo, B.-T. (2011). A Random-Finite-Set approach to bayesian SLAM. IEEE Transactions on Robotics, 27(2), 268–282.

Mullane, J., Vo, B.-N., Adams, M., & Vo, B.-T. (2013). Random Finite Sets for Robot Mapping and SLAM - New Concepts in Autonomous Robotic Map Representations. Springer Tracts in Advanced Robotics. Berlin Heidelberg, Germany Springer-Verlag.

Murray, R. (2007). Recent research in cooperative control of multivehicle systems. ASME Journal of Dynamic Systems, Measurement, & Control, 31(129), 571–583.

Nagatani, K., Okada, Y., Tokunaga, N., Kiribayashi, S., Yoshida, K., Ohno, K., Takeuchi, E., Tadokoro, S., Akiyama, H., Noda, I., Yoshida, T., & Koyanagi, E. (2011). Multirobot exploration for search and rescue missions: A report on map building in RoboCupRescue 2009. Journal of Field Robotics, 28(3), 373–387.

Nettleton, E., Gibbens, P., & Durrant-Whyte, H. (2000). Closed form solutions to the multiple platform simultaneous localization and map building (SLAM) problem. Sensor Fusion, 4051, 428–437.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., & Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, pp. 127–136. Basel, Switzerland: IEEE.

Newman, P., Cole, D., & Ho, K. (2006). Outdoor SLAM using visual appearance and laser ranging. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 1180–1187. Orlando, FL, USA: IEEE.

Newman, P. M., Leonard, J. J., & Rikoski, R. J. (2005). Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In Proceedings of the Eleventh International Symposium on Robotics Research, pp. 409–420. Siena, Italy: Springer Verlag.

Nguyen, V., Fox, D., Martinellii, A., Tomatis, N., & Siegwart, R. (2005). A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1929–1934. Edmonton, AB, Canada: IEEE.

Ni, K., Steedly, D., & Dellaert, F. (2007). Tectonic SAM: Exact, out-of-core, submap-based SLAM. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 1678–1685. Roma, Italy: IEEE.

Nieto, J., Bailey, T., & Nebot, E. (2007). Recursive scan-matching SLAM. Robotics and Autonomous Systems, 55(1), 39–49.

Nuchter, A., & Hertzberg, J. (2008). Towards semantic maps for mobile robots. Robotics and Autonomous Systems, 56(11), 915–926.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Transactions on Automatic Control, 51(3), 401–420.

Olfati-Saber, R., Fax, J., & Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE, 95(1), 215–233.

Olson, E., Leonard, J., & Teller, S. (2006). Fast iterative alignment of pose graphs with poor initial estimates. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 2262–2269. Orlando, FL, USA: IEEE.

Olson, E., Strom, J., Goeddel, R., Morton, R., Ranganathan, P., & Richardson, A. (2013). Exploration and mapping with autonomous robot teams. Communications of the ACM, 56(3).

Pathak, K., Birk, A., Vaskevicius, N., Pfingsthorn, M., Schwertfeger, S., & Poppinga, J. (2010). Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. Journal of Field Robotics, 27(1), 52–84.

Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). AUV navigation and localization: A review. IEEE Journal of Oceanic Engineering, 39(1), 131–149.

Pillonetto, G., Bell, M. B., & Del Favero, S. (2013). Distributed Kalman smoothing in static bayesian networks. Automatica, 49(4), 1001–1011.

Radon, J. (1986). On the determination of functions from their integral values along certain manifolds. IEEE Transactions on Medical Imaging, 5(4), 170–176.

Ranganathan, A., Menegatti, E., & Dellaert, F. (2006). Bayesian inference in the space of topological maps. IEEE Transactions on Robotics, 22(1), 92–107.

Rekleitis, I., Dudek, G., & Milios, E. (2001). Multi-robot collaboration for robust exploration. Annals of Mathematics and Artificial Intelligence, 31(1-4), 7–40.

Ren, W., Beard, R., & Atkins, E. (2005). A survey of consensus problems in multi-agent coordination. In Proceedings of the American Control Conference (ACC), pp. 1859–1864. Portland, OR, USA: IEEE.

Ren, W., Beard, R., & Atkins, E. (2007). Information consensus in multivehicle cooperative control. IEEE Control Systems, 27(2), 71–82.

Richardson, T. S., Jones, C. G., Likhoded, A., Sparks, E., Jordan, A., Cowling, I., & Willcox, S. (2013). Automated vision-based recovery of a rotary wing unmanned aerial vehicle onto a moving platform. Journal of Field Robotics, 30(5), 667–684.

Rone, W., & Ben-Tzvi, P. (2013). Mapping, localization and motion planning in mobile multi-robotic systems. Robotica, 31(1), 1–23.

Roumeliotis, S., & Bekey, G. A. (2002). Distributed multirobot localization. IEEE Transactions on Robotics and Automation, 18(5), 781–795.

Saeedi, S., Paull, L., Trentini, M., & Li, H. (2011a). Multiple robot simultaneous localization and mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 853–858. San Francisco, CA, USA: IEEE.

Saeedi, S., Paull, L., Trentini, M., & Li, H. (2011b). A neural network-based multiple robot simultaneous localization and mapping. IEEE Transactions on Neural Networks, 22(12), 2376–2387.

Saeedi, S., Paull, L., Trentini, M., & Li, H. (2011c). A neural network-based multiple robot simultaneous localization and mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 880–885. San Francisco, CA, USA: IEEE.

Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2012a). Efficient map merging using a probabilistic generalized Voronoi diagram. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4419–4424. Vilamoura, Portugal: IEEE.

Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2012b). Map merging using Hough peak matching. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4683–4688. Vilamoura, Portugal: IEEE.

Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2014a). Group mapping: A topological approach to map merging for multiple robots. IEEE Robotics Automation Magazine, 21(2), 60–72.

Saeedi, S., Paull, L., Trentini, M., Seto, M., & Li, H. (2014b). Map merging for multiple robots using hough peak matching. Robotics and Autonomous Systems, 62(10), 1408–1424.

Salas-Moreno, R., Newcombe, R., Strasdat, H., Kelly, P., & Davison, A. (2013). SLAM++: Simultaneous localisation and mapping at the level of objects. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1352–1359. Portland, OR, USA: IEEE.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry [tutorial]. IEEE Robotics and Automation Magazine, 18(4), 80–92.

Schneider, F. E., & Wildermuth, D. (2012). Influences of the robot group size on cooperative multi-robot localisation-analysis and experimental validation. Robotics and Autonomous Systems, 60, 1421–1428.

Schwertfeger, S. & Birk, A. (2013). Evaluation of map quality by matching and scoring high-level, topological map structures. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 2221–2226. Karlsruhe, Germany: IEEE.

Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized ICP. In Proceedings of Robotics: Science and Systems (RSS), Seattle, WA, USA. Cambridge, MA, USA: MIT Press.

Sheng, W., Yang, Q., Tan, J., & Xi, N. (2006). Distributed multi-robot coordination in area exploration. Robotics and Autonomous Systems, 54(12), 945–955.

Shi, J., & Tomasi, C. (1994). Good features to track. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 593–600. Seattle, WA, USA: IEEE.

Silver, D., Ferguson, D., Moms, A., & Thayer, S. (2004). Feature extraction for topological mine maps. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 773–779. Sendai, Japan: IEEE.

Smith, M., Baldwin, I., Churchill, W., Paul, R., & Newman, P. (2009). The new college vision and laser data set. The International Journal of Robotics Research, 28(5), 595–599.

Smith, R., Self, M., & Cheeseman, P. (1987). A stochastic map for uncertain spatial relationships. In Fourth International Symposium of Robotics Research, pp. 467–474. Cambridge, MA, USA: MIT Press.

Strasdat, H., Davison, A., Montiel, J., & Konolige, K. (2011). Double window optimisation for constant time visual SLAM. In IEEE International Conference on Computer Vision (ICCV), pp. 2352–2359. Barcelona, Spain: IEEE.

Sturm, J., Bylow, E., Kahl, F., & Cremers, D. (2013). Dense Tracking And Mapping With A Quadrocopter. In Unmanned Aerial Vehicle in Geomatics (UAV-g), Rostock, Germany.

Sukhatme, G. S., Montgomery, J. F., & Vaughan, R. T. (2001). Experiments with cooperative aerial-ground robots. In Robot Teams: From Diversity to Polymorphism. AK Peters, pp. 345–367. Natick, MA, USA: A. K. Peters, Ltd.

Takeuchi, E., & Tsubouchi, T. (2008). Multi sensor map building based on sparse linear equations solver. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2511–2518.

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

Thrun, S. (2003). Learning occupancy grids with forward sensor models. Autonomous Robots, 15, 111–127.

Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. Cambridge, MA: The MIT Press.

Thrun, S. & Liu, Y. (2005). Multi-robot SLAM with sparse extended information filers. Springer Tracts in Advanced Robotics, 15, 254–266.

Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., & Durrant-Whyte, H. (2004a). Simultaneous localization and mapping with sparse extended information filters. The International Journal of Robotics Research, 23(7-8), 693–716.

Thrun, S., Martin, C., Liu, Y., Hahnel, D., Emery-Montemerlo, R., Chakrabarti, D., & Burgard, W. (2004b). A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. IEEE Transactions on Robotics and Automation, 20(3), 433–443.

Tobata, Y., Kurazume, R., Noda, Y., Lingemann, K., Iwashita, Y., & Hasegawa, T. (2012). Laser-based geometrical modeling of large-scale architectural structures using co-operative multiple robots. Autonomous Robots, 32(1), 49–62.

Tomatis, N., Nourbakhsh, I., & Siegwart, R. (2003). Hybrid simultaneous localization and map building: A natural integration of topological and metric. Robotics and Autonomous Systems, 44(1), 3–14.

Tong, C. H., Gingras, D., Larose, K., Barfoot, T. D., & Dupuis, É. (2013). The Canadian planetary emulation terrain 3D mapping dataset. The International Journal of Robotics Research, 32(4), 389–395.

Ulrich, I., & Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp. 1023–1029. San Francisco, CA, USA: IEEE.

Vidal-Calleja, T. A., Berger, C., Sola, J., & Lacroix, S. (2011). Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain. Robotics and Autonomous Systems, 59(9), 654–674.

Virginia Vassilevska Williams. Williams, V. V. (2012). Multiplying matrices faster than Coppersmith-Winograd. In Proceedings of the 44th Symposium on Theory of Computing Conference (STOC), pp. 887–898. New York, NY, USA: ACM.

Wallgrum, J. O. (2010). Voronoi graph matching for robot localization and mapping. Transactions on computational science IX, 6290, 76–108.

Wang, C.-C., Thorpe, C., & Thrun, S. (2003). Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA). Taipei, Taiwan: IEEE.

Weingarten, J., & Siegwart, R. (2006). 3D SLAM using planar segments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3062–3067. Beijing, China: IEEE.

Whyte, H. D., & Bailey, T. (2006). Simultaneous localization and mapping (SLAM), Part I the essential algorithms. IEEE Robotics and Automation Magazine, 13(3), 108–117.

Williams, S., Dissanayake, G., & Durrant-Whyte, H. (2002a). An efficient approach to the simultaneous localisation and mapping problem. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 1, pp. 406–411.

Williams, S., Dissanayake, G., & Durrant-Whyte, H. (2002b). Towards multi-vehicle simultaneous localisation and mapping. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 3, pp. 2743–2748. Washington, DC, USA: IEEE.

Willow Garage (2006). Robot Operating System (ROS). Retrieved January 5, 2014, from http://www.ros.org/.

Wurm, K., Stachniss, C., & Burgard, W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1160–1165. Nice, France: IEEE.

Wurm, K. M., Stachniss, C., & Grisetti, G. (2010). Bridging the gap between feature- and grid-based SLAM. Robotics and Autonomous Systems, 58(2), 140–148.

Wyeth, G., & Milford, M. (2009). Spatial cognition for robots. IEEE Robotics and Automation Magazine, 16(3), 24–32.

Yun, X., Latt, K., & Glennon, J. (1998). Mobile robot localization using the Hough transform and neural networks. In Proceedings of the Intelligent Control (ISIC), Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), pp. 393–400. Gaithersburg, MD, USA: IEEE.

Zhou, X., & Roumeliotis, S. (2008). Robot-to-robot relative pose estimation from range measurements. IEEE Transactions on Robotics, 24(6), 1379–1393.

Zhou, X. S., & Roumeliotis, S. I. (2006). Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1785–1792. Beijing, China: IEEE.

Zlot, R., Stentz, A., Dias, M., & Thayer, S. (2002). Multi-robot exploration controlled by a market economy. In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), volume 3, pp. 3016–3023. Washington, DC, USA: IEEE.