Robotica (2013) volume 31, pp. 1–23. © Cambridge University Press 2012
doi:10.1017/S0263574712000021

Mapping, localization and motion planning in mobile multi-robotic systems

William Rone and Pinhas Ben-Tzvi*

Robotics and Mechatronics Laboratory, Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC, USA

(Accepted January 6, 2012. First published online: February 9, 2012)

SUMMARY
As researchers have pushed the limits of what can be accomplished by a single robot operating in a known or unknown environment, a greater emphasis has been placed on the utilization of mobile multi-robotic systems to accomplish various objectives. In transitioning from a robot-centric approach to a system-centric approach, considerations must be made for the computational and communicative aspects of the group as a whole, in addition to electromechanical considerations of individual robots. This paper reviews the state-of-the-art of mobile multi-robotic system research, with an emphasis on the confluence of mapping, localization and motion control of robotic system. Methods that compose these three topics are presented, including areas of overlap, such as integrated exploration and simultaneous localization and mapping. From these methods, an analysis of benefits, challenges and tradeoffs associated with multi-robotic system design and use are presented. Finally, specific applications of multi-robotic systems are also addressed in various contexts.

KEYWORDS: Multi-robotic systems; Mobile robots; Motion planning; Robot localization; SLAM; Path planning; Robotic exploration.

1. Introduction

As trends in robotics have pushed toward liberating robots from fixed positions in their environment and allowing them to interact with one another, research in multi-robotic systems (MRS) has grown from a novelty to a critical area for consideration and investment. This transition from an individual approach to a collective approach in designing and utilizing systems requires an augmentation of local considerations (such as electromechanical layout, on-board computational ability and robot-operator communication) with global considerations (such as inter-robot communication and task assignment).

The motivation of this work is to present the state-of-the-art for MRS, with a particular emphasis on the intersection of mapping, localization and motion planning. During operation, a *map* is necessary to enable long-term planning; it may be known beforehand or generated during movement throughout the environment. When utilizing a map, there is a need to *localize* the system within that map in terms of sensing its own and its environment's properties. In order to traverse the environment there must be some sort of *motion plan* to navigate the system from an initial pose to a desired pose while avoiding obstacles as they appear on the way.

Figure 1 shows these three requirements graphically in a Venn diagram, with the overlaps between them categorized. Simultaneous Localization and Mapping (SLAM) describes a class of algorithms that allows for the simultaneous building of a map and placement of the robotic system within it, decoupled from the motion planning required to traverse. When that motion planning is coupled to SLAM, integrated exploration is the result, providing a unified framework for the discovery of an unknown environment or verification.

Active localization and classic exploration are included in the figure for completeness, though they are not major research topics. For the former, it is not realistic to assume a perfect map of any space; therefore, any attempts at active localization would be hindered by not including protocols to validate/update the map during operation. For the latter, a mapping operation will quickly fail if the robot is not constantly being localized within the built map.

The paper is organized as follows: Section 2 addresses the methods associated with MRS organization, mapping, localization, motion planning, SLAM and exploration; Section 3 extrapolates from these methods the benefits, challenges and tradeoffs associated with the field; Section 4 addresses the practical implementation of MRS, including applications and sensor technologies used to instrument systems in real-world scenarios; and Section 5 summarizes the review and draws conclusions from it.

This paper is motivated by the difficulty new researchers have in understanding the multitude of fields composing mobile MRS. By beginning with the core subjects of mapping, localization and motion control, then analyzing their combinations in terms of SLAM and integrated exploration, the progression of technology from its earlier stages to the state-of-the-art can be better understood. Furthermore, by breaking down the analysis in terms of the methods, benefits/challenges/tradeoffs and practical implementation of the fields, a better grasp of the interconnections of the fields is achieved. This review is not designed to be all encompassing – rather, it equips new researchers with a fundamental understanding of the field and gives them the tools they will need to further explore and

* Corresponding Author. E-mail: bentzvi@gwu.edu

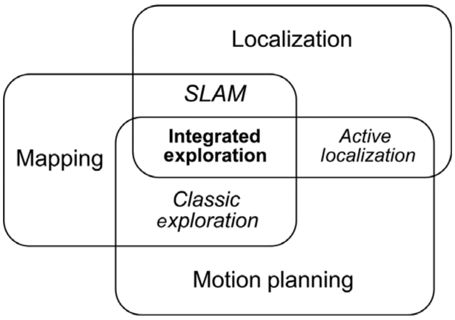


Fig. 1. Fields associated with robotic exploration (adapted from ref. [1]).

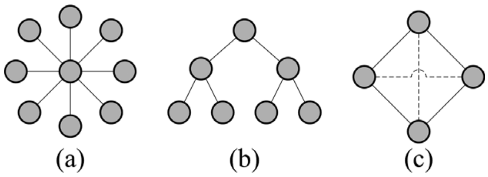


Fig. 2. Network organizations: (a) centralized, (b) hierarchical and (c) decentralized.

understand specific areas of interest. Furthermore, it provides a wealth of sources to experienced researchers in the field to better understand the state-of-the-art of the past decade, with an emphasis on the most recent five years.

2. Methods

In order to understand the challenges associated with MRS, an understanding of the current methods used in MRS is necessary. These methods include strategies for (1) organizing the communication and mechanical components of MRS, (2) storing and manipulating environmental data in maps, (3) localizing the MRS both internally and with respect to its environment, (4) planning the motions of the member robots, (5) simultaneously localizing the MRS while mapping its environment and (6) incorporating this simultaneous localization and mapping with motion planning to explore the environment.

2.1. Organization

A fundamental constraint on the operation of an MRS is its network organization or architecture. Three fundamental architectures exist: centralized, hierarchical and decentralized, as shown in Fig. 2. Centralized architectures focus all communication and computation through a single entity, either a member of the robotic team or an external controller.[2] While this architecture simplifies the control by allowing a single entity to know the entire state of the global system, it is prone to failure if this single agent or controller fails.

Hierarchical architectures borrow from centralized architectures by having a single robot control a group of robots, but the hierarchical approach is stratified, with members of one controller's group each overseeing another group of robots, and so on, until there is a strata of robots that simply perform tasks. While this method does allow for greater fault tolerance "lower" in the hierarchy, the system is still prone to failure if a robot high in the command chain fails.[3]

Decentralized architectures provide the greatest flexibility of the MRS network architectures by not having any single potential point of failure. Instead, the communication among units and computation associated with planning and execution is performed locally, with minimal communication among modules. While this architecture is immensely more fault-tolerant than the previous two, it becomes much more difficult to articulate "global" goals into local controllers because of the controllers' reduced reliance on one another.[4–6] As shown in Fig. 2(c), different types of decentralized strategies exist: the system may only be able to communicate in a "ring" structure, as shown with solid lines, or may allow for full communication, as shown by the solid and dashed lines.

A fourth category, hybrid architectures, employs elements of multiple network architectures. Simmons *et al.*[7] employed a hybrid architecture that allows for decentralized planning/execution of the robot's actions and cooperation among these planners on different systems.

Beyond the communicative aspects of organization, there may exist physical connectivity between robots, either permanent or temporary. Ben-Tzvi *et al.*[8] presented a robotic system composed of three permanently connected yet individual subsystems, with wireless communication facilitating their collective use. Kim *et al.*[9] designed several permanently coupled robotic systems, with the coupling either directly between robotic modules or through an external load.[10] O'Grady *et al.*[11] described a robot design with manipulators that would attach to other robots, creating a rigid structure, but with the ability to disengage and continue separately if necessary.

2.2. Mapping

Once a robotic network is established, there are significant benefits from understanding the environment in which it is operating. These benefits include better situational awareness and a baseline for long-term planning, compared to acting solely in response to sensed data. A robotic system's understanding of its environment is its map. While colloquial understanding of a map is a two-dimensional (2D) representation of a space, the maps utilized by MRS may (but not necessarily) take drastically different forms. The data a map contains are driven primarily by the needs of the system for its specific application. Furthermore, a necessary feature for the MRS maps is the capacity for merging maps built by individual robots to form a global representation of the space (if the map is unknown *a priori*).[12]

2.2.1. Types of maps. One of the most common and intuitive types of maps used by MRS are *occupancy-based grid maps*, first described by Elfes in 1989.[13] Occupancy-based grids decompose a space into a regular grid and assign a value to each element of the grid to denote the probability of its occupancy by an obstacle. Range-sensors, such as LIDAR, SONAR or stereovision, can be used to update this map as the robots traverse the space and collect information about

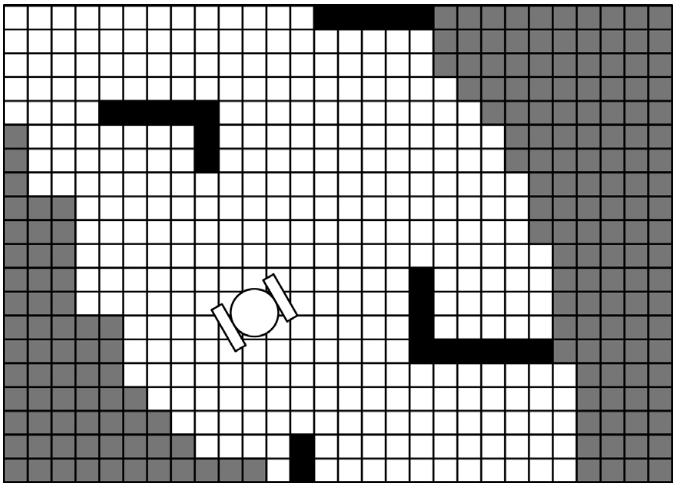


Fig. 3. Occupancy-based grid map.

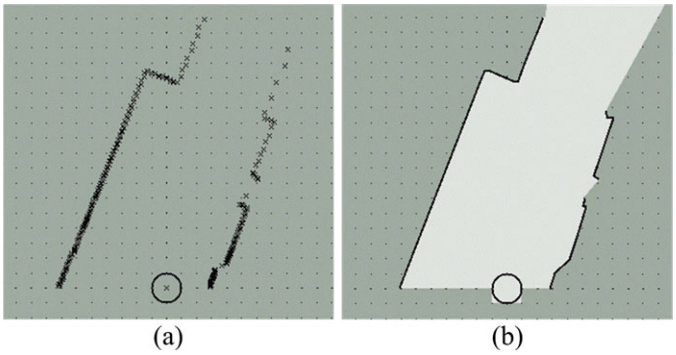


Fig. 4. (Colour online) Typical feature-based (a) point map, (b) line map.[18]

unvisited and previously visited cells. Benefits of occupancy-based grid maps are their ease of merging and flexibility in incorporating data from numerous different types of sensors. Challenges associated with occupancy-based grids are the high computation expense to maintain large grids and the difficulty in extracting optimal paths from a grid-based map without human intervention.[14,15] An example of an occupancy-based grid is shown in Fig. 3, with clear areas in white, obstacles in black and unexplored areas in gray.

Feature-based maps, also called geometric or polygonal maps, utilize geometric primitives such as points, lines and arcs to represent obstacles in a space, as shown in Fig. 4. An advantage of feature-based maps is their computational efficiency: in occupancy grids, data are stored whether or not there is an obstacle in the grid; in feature maps, only the obstacle data are stored, mapped in a local or global coordinate frame. This reduced computational load facilitates motion planning and exploring (Sections 2.4 and 2.5). A key difficulty is extracting the simplified features from the raw sensor data in an efficient and reliable way.[16–18]

Topological maps allow for a space to be broken down into a graph comprising nodes, which represent physical locations within the space, and lines, which connect the nodes and signify paths between these discrete spaces, as shown in Fig. 5. This significantly reduces the computational load of utilizing the map in applications such as motion planning, but accompanying this computational simplification is a

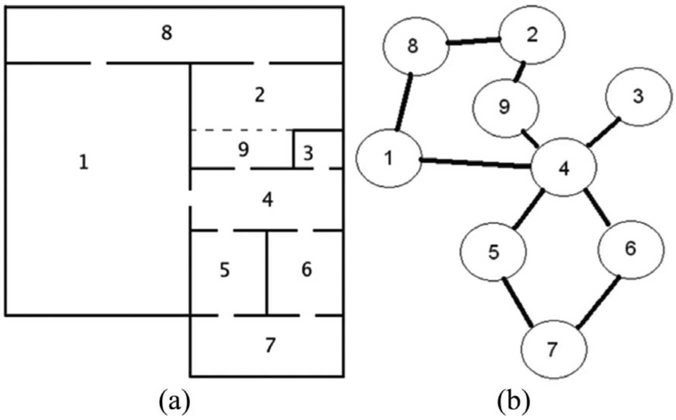


Fig. 5. (a) Physical environment, and (b) topological representation.[22]

reduction in useful information regarding the nature and structure of the actual environment.[19–21]

While 2D mapping is the most common objective in most ground robotic activities, there are cases when *volumetric maps* become necessary. Rocha *et al.*[23] demonstrated the utilization of occupancy-based grid mapping to create volumetric representation, with a grid comprising voxels (cubes) versus squares and an experimental implementation using stereo vision sensors. A typical volumetric map is shown in Fig. 6.

In some scenarios, *multistorey maps* allow for generalizations to be made when exploring multiple floors of the same building. These generalizations utilize the knowledge that different floors of the same building, in most cases, have the same general footprint and layout. While in specific cases (e.g. cubicle arrangement) the arrangement of obstacles may be different, features such as columns, stairs and elevators serve as absolute benchmarks with which correspondence can be drawn, as shown in Fig. 7.[24]

Beyond physical layouts and their topological abstractions, *image maps* allow for camera-type sensors to more reliably map their environments by storing the previously taken images. This method is an extension of topographical techniques, but instead of storing a location or a region of the space in a node, an image is stored, and the lines connecting these nodes represent correspondence between different images.[25]

Hierarchical maps allow for the integration of these different types of maps into a single framework to allow systems to utilize the minimal amount of data needed for a specific task without the loss of the entire data set. Generally, these hierarchies will consist of a topological map for generalized path planning (due to its computational efficiency) and an occupancy-based grid for its ease of creation, as shown in Fig. 8.[26] However, other topological maps have included image map layers[25] or higher level feature identities.[27]

2.2.2. Merging maps. In any system with multiple robots performing mapping of the environment, there is a need to merge local maps produced by individual robots with one another in an automated, accurate and reliable manner. The most intuitive mechanism for merging maps is their

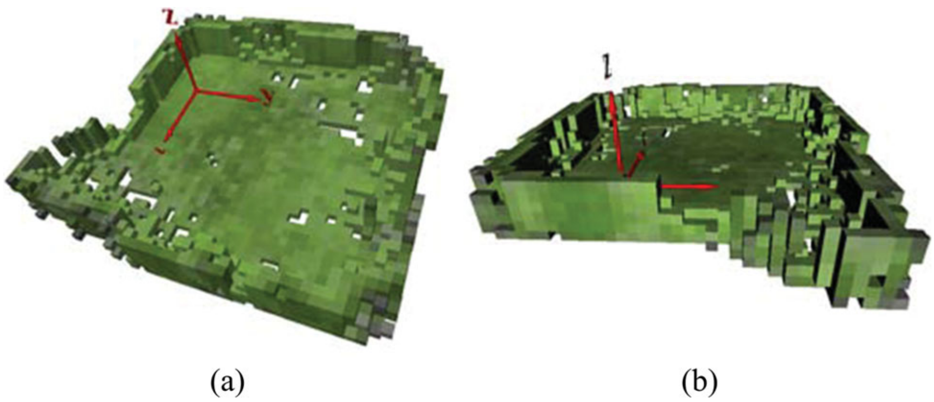


Fig. 6. (Colour online) Typical volumetric map.[23]

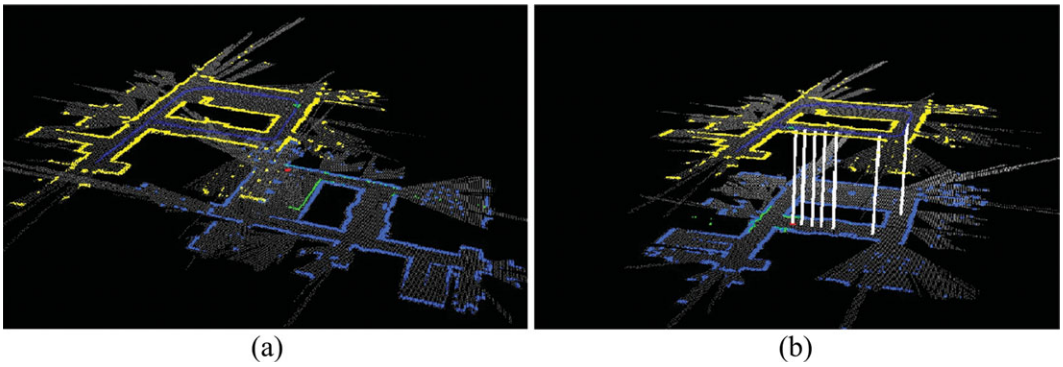


Fig. 7. (Colour online) Map of two adjacent building floors: (a) before correspondence, and (b) after correspondence.[24]

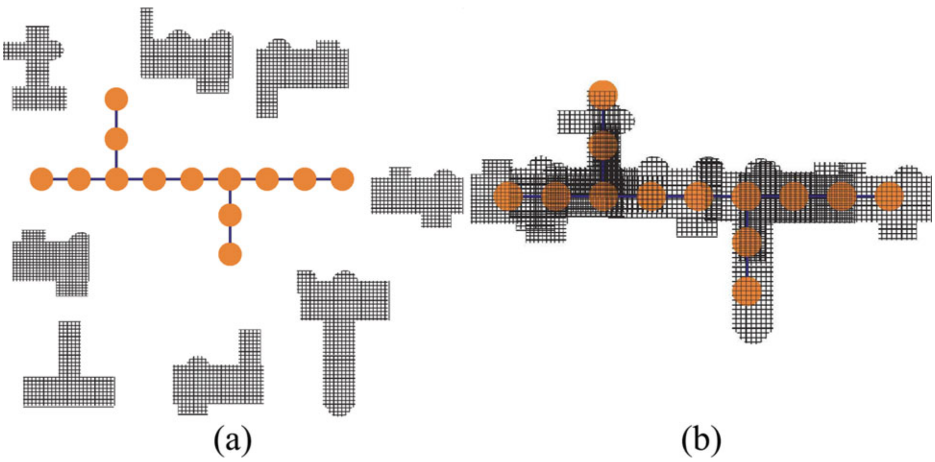


Fig. 8. (Colour online) Hierarchical maps: (a) topological and geometric sub-maps, and (b) overlain sub-maps.[28]

superposition based on the *known initial configuration* of the robots relative to one another and/or a global landmark. However, while this task seems trivial, it requires an accurate understanding of the robots' current positions relative to their original configuration; a problem addressed by localization (Section 2.3).

If the original configuration is not known, or the current positions of the robot do not possess the necessary accuracy, merging based on other metrics is possible. The first is *correspondence*, where recognizable arrangements of scan points, such as doors, junctions and corners, are isolated within different maps and analyzed for overlap. This method has been applied to both occupancy grid maps[29] and feature-based maps.[30] A key challenge associated with this method is the need for overlap of the map; the algorithm must be able to recognize when this does not occur, as opposed to joining maps at the "best" correspondence site, especially if the "best" is not good. This challenge is illustrated in Fig. 9.

Relative position estimation is another mechanism for merging maps, where two or more robots estimate the other's position relative to itself, and those two local estimates are combined to provide a better overall estimate. A key advantage of this method over correspondence is the ability to merge maps without overlap; robots do not need to have visited the same segment of the environment, but simply possess a robust method of estimating relative positions.[31]

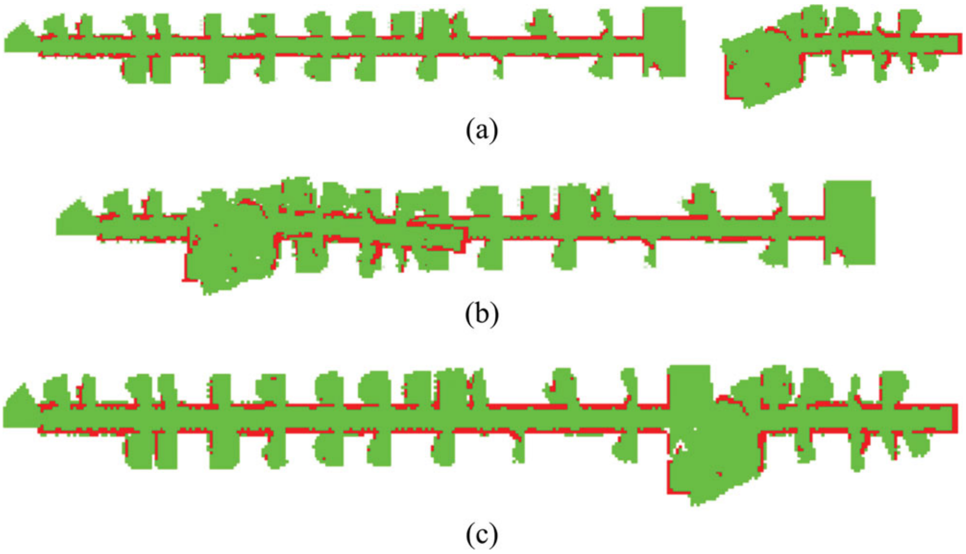


Fig. 9. (Colour online) Merging maps: (a) local sub-maps, (b) erroneous merging and (c) correct merging.[29]

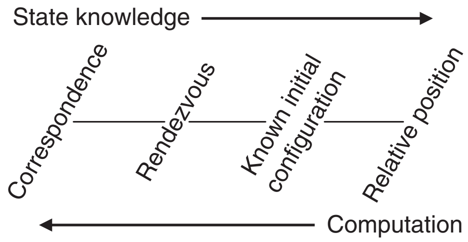


Fig. 10. State knowledge vs. computation requirements for merging maps.

Rendezvous is a fourth method of merging maps, where two robots approach and meet one another in order to generate a common point (their target location) in both of their maps, with a relative position estimate with very high accuracy (because of their closeness). This is an extension of relative position estimation, but the relative position becomes the same position due to the motion of the robots toward one another.[32]

Figure 10 compares these four methods with respect to their state knowledge and computational requirements. Because correspondence solely utilizes the map information to determine overlap, there is a heavy computational requirement in processing of maps. At the opposite end of the spectrum, relative position methods require precise knowledge of the robots' states relative to one another at each moment mapping occurs in order to successfully merge maps. Between these extrema, the rendezvous and known initial configuration methods require both state knowledge and computation; however, because the rendezvous method requires the robots to approach one another, there is a greater computational cost. It should be noted that this analysis does not consider the computational requirements for localization (which the rendezvous, known initial configuration and relative position methods require); the computation strictly relates to the effort that must be expended to merge the maps.

2.2.3. Additional mapping-specific issues. In topological maps, a pressing challenge is ensuring *cycle detection*, or *loop closure*. This is required when the topological map rejoins a previously visited node. Because of the serial nature by which nodes are created as robots move along their paths, it is not trivial for it to recognize a previously visited point when and its most recently generated node should be connected to an "older" node.[21] Figure 11 illustrates this point; if the original node was not detected, the robot would need to traverse around the generated path to return to the door.

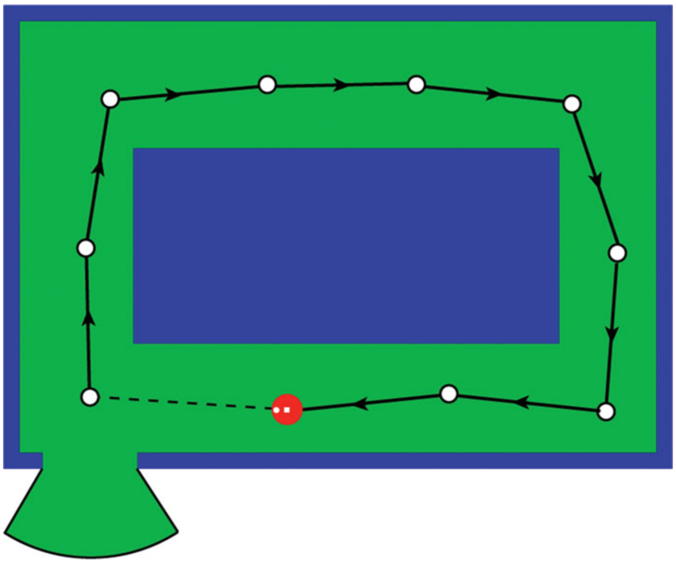
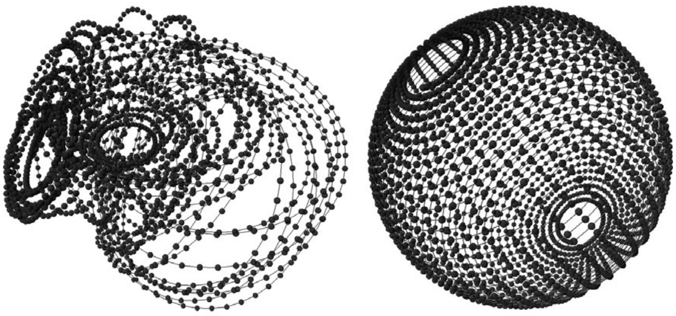


Fig. 11. (Colour online) Cycle detection/loop closure.[33]

Depending on the environment being mapped, a *spatial map correction* may be necessary, even in a 2D environment. Grisetti *et al.*[34] used simulations and experimentation to determine spatial corrections for 2D maps both numerically (with simulations of a robot mapping a sphere) and experimentally (with a robot traversing a college campus with hills), as shown in Fig. 12.

Because of the prevalence of cameras as sensors, *image extraction* is a key challenge in building various maps



(a) (b)

Fig. 12. Simulations of 3D planar mapping: (a) without 3D correction, and (b) with 3D correction.[34]

described previously. Specifically, the ability to isolate features from the images, and incorporate these features into the map being generated or to match these features to a map already known is critical in utilizing these sensors in addition to or instead of more quantitative range of finding sensors such as SONAR.[25] Figure 13 shows post-processed images to extract the planar features, with different detected planes in different colors.

While research has primarily focused on static environments, *modifying maps in dynamic environments* is a critical challenge in translating this field of research into widespread use. The key challenge lies in detecting the change in the environment – to know with confidence that an obstacle that was present at one time is no longer present, or *vice versa*. While the Bayesian update procedures are common in refining the map as robots traverse the space, considerations need to be made for radical changes in occupancy, feature-distribution or graph connectivity, depending on the map being used.[35,36]

2.3. Localization

In order to effectively utilize a map of the environment or to generate such a map *ab initio* (discussed in Section 2.5), localization is needed to understand where robots are within their own maps and/or their environment. In single-robotic systems, localization has two key components: local and global. *Local localization*, also called position tracking, refers to the accurate tracking of each robot's current location relative to its start point. *Global localization*, also called global self-actualization, refers to the accurate tracking of each robot's current location relative to its environment. When the start point of each robot within the environment is known, these problems collapse into a single localization problem, but when the start points are unknown, they remain distinct yet coupled.[37,38]

In MRS, *collective* or *mutual localization*, a third type, emerges when there is a need to know the relative positions of each of the robots with respect to one another. It is a more challenging problem than position tracking, but simpler than global localization (though it becomes trivial if a global localization for the entire system is already available). Work has also been presented to perform this collective localization without needing the identities of other robots within the MRS. This reduces the required communication between robots.[2,39]

In terms of computational complexity, the optimal solution to the multi-robot collective localization problem (assuming only local and relative observations are available) has been determined to be non-deterministic polynomial-time hard (NP-hard). As a consequence of this, the problem cannot be solved by a deterministic algorithm in polynomial time unless $P = NP$.[40] However, there have been attempts made to either re-cast the problem in a form that enables a polynomial-time solution, or to approximate the NP-hard optimal solution using a polynomial-time algorithm. For the first case (re-casting the problem), additional constraints

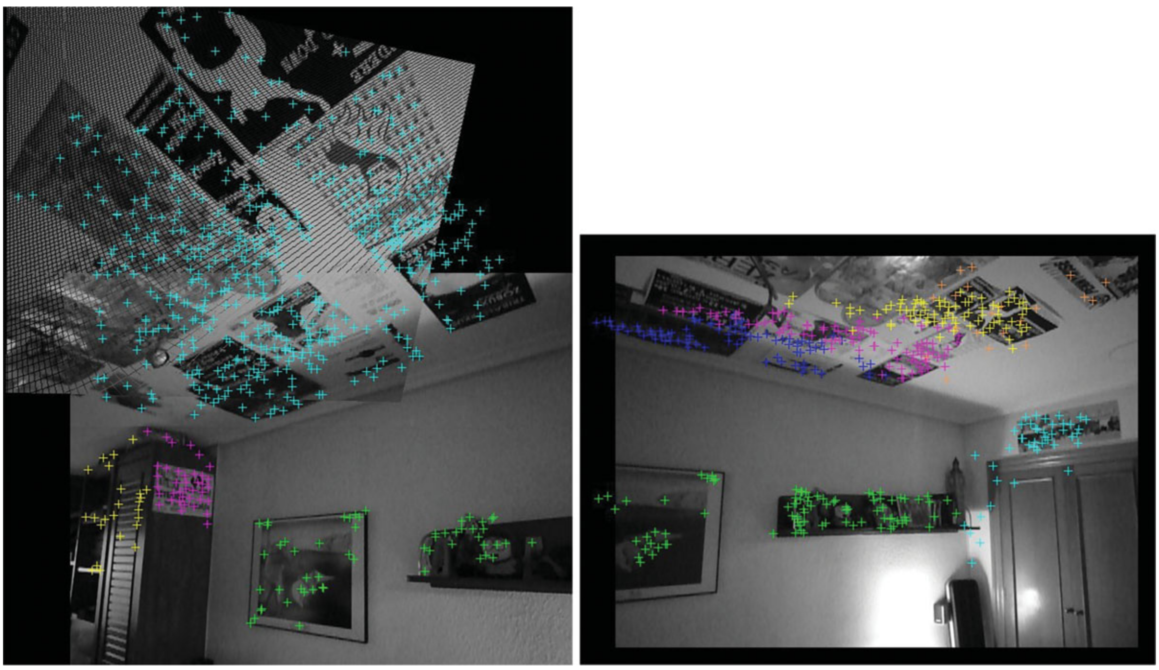


Fig. 13. (Colour online) Images with extracted planar features.[25]

on the robot's relative positions[41] and environmental data utilized[42] have successfully reduced computational complexity. Furthermore, the order of complexity has been shown to vary depending on the method/assumptions chosen: an extended Kalman filter may have a cost of $O(n^3)$, while a particle filter may have a cost of O(n).[43] For the second case (approximate solutions), algorithms to approximate the NP-hard optimal solution include works by Gerkey and Matarić,[44] Lagoudakis *et al.*[45] and Singh *et al.*[46]

These three types of localizations are facilitated by two types of sensors: interoceptive and exteroceptive. Interoceptive sensors (such as wheel encoders, gyroscopes and accelerometers) track the motion of an individual robot on which the sensor is mounted and extrapolates that data into a path followed by the robot. Exteroceptive sensors (such as SONAR and cameras) measure and represent the environment of the robot, including obstacles and other robots.[47,48]

For a single robot system, one generalized strategy for global localization utilizing these two classes of sensors consists of three steps: pose prediction, local/global map correlation and pose estimation. Pose prediction uses the interoceptive information collected by the robot to determine a change in position in the local map independent of the environment. Local and global map correlation utilizes exteroceptive sensors to relate the change in external sensing to a change in global position. Finally, pose estimation combines these two determinations to estimate the actual global localization with better accuracy than either of the two independent assessments.[49]

For MRS, different strategies exist for localization in known and unknown environments. In a known environment, global localization can be performed by a "leader" robot, and collective localization is performed on the entire team of the leader and its "followers." By simple extrapolation, the collective localization can be used to globally localize each follower in relation to its leader.[50]

In an unknown environment, different applications of SLAM are needed to create local maps, localize the robot(s) within those maps on each robot and integrate those maps and localizations into a global framework. When communication is limited during this process, the integration of maps and localizations must be performed offline after the exploration task is finished, forcing each robot to perform SLAM independent of one another (though strategies can be incorporated in the local control laws to circulate robots throughout the environment). However, if communication is not a limiting factor, Collective-SLAM can be performed with each robot providing its sensory data to a centralized processor to build the global map and localize each robot within it centrally.[50] SLAM is further discussed in Section 2.5.

2.3.1. Techniques for localization. Dead-reckoning provides the most intuitive procedure for position estimation. Wheel rotations are measured using encoders and integrated over time to determine the trajectory travelled by the robot. However, due to effects such as wheel slippage, in applications where the robot is travelling long distances, the error accumulation, if not compensated for, causes the position estimate to become useless.[51]

Non-linear *Kalman filtering* is the most common way by which these errors are reduced by coupling this measurement with others, such a gyroscopes or Global Positioning System (GPS). The goal of utilizing Kalman filters is to reduce the noise and other inaccuracies present in sensor values, and to synergize sensor values to provide the best estimate of the measured state. Specifically, two types of filters are used: extended and unscented. Extended Kalman filters propagate the state covariance associated with the measurement through a linearized model, while unscented Kalman filters use a sampling technique known as the unscented transform to better capture the nonlinearity with an accuracy of the third-order Taylor expansion.[37,43,48]

Mathematically, extended Kalman filters in discrete time require differentiable functions for the state transition and observation models, as shown in Eq. 1, where $\boldsymbol{x}_k$ is the state estimate vector, $\boldsymbol{u}_k$ is the control action vector, z_k is the output vector, $\boldsymbol{f}$ and $\boldsymbol{h}$ are differentiable functions and $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the process and observation noises with covariances of $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ (note: the subscript denotes the discrete time at which the variable represents):

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1} \\ z_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k. \end{aligned} \tag{1}$$

Using $\boldsymbol{f}$, an initial prediction of the state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ based on the state estimate $\hat{\boldsymbol{x}}_{k-1|k-1}$ and control action $\boldsymbol{u}_{k-1}$ can be constructed, as shown in Eq. 2. Using this estimate, the state transition ($\boldsymbol{F}_{k-1}$) and observation ($\boldsymbol{H}_k$) matrix can be estimated through linearization, taking the Jacobian of the $\boldsymbol{f}$ and $\boldsymbol{h}$ functions, respectively, as seen in Eq. 3:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1},\ \boldsymbol{u}_{k-1}), \tag{2}$$

$$\begin{aligned} \boldsymbol{F}_{k-1} &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1},\ \boldsymbol{u}_{k-1}} \\ \boldsymbol{H}_k &= \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}}. \end{aligned} \tag{3}$$

With the state transition matrix estimate, a prediction of the estimate of the covariance can be obtained using Eq. 4,

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_k. \tag{4}$$

After these preliminary predictions and approximations are made, the state estimate can be updated. First, the residual difference $\tilde{\boldsymbol{y}}_k$ and residual covariance $\boldsymbol{S}_k$ are calculated using the current measured values and the previous predicted state and covariance estimates, using Eq. 5. Then, a near-optimal Kalman gain $\boldsymbol{K}_k$ is calculated using Eq. 6 to create the updated state estimate $\hat{\boldsymbol{x}}_{k|k}$ and updated covariance estimate $\boldsymbol{P}_{k|k}$ using Eq. 7:

$$\begin{aligned} \tilde{\boldsymbol{y}}_k &= z_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}) \\ \boldsymbol{S}_k &= \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k, \end{aligned} \tag{5}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1}, \tag{6}$$

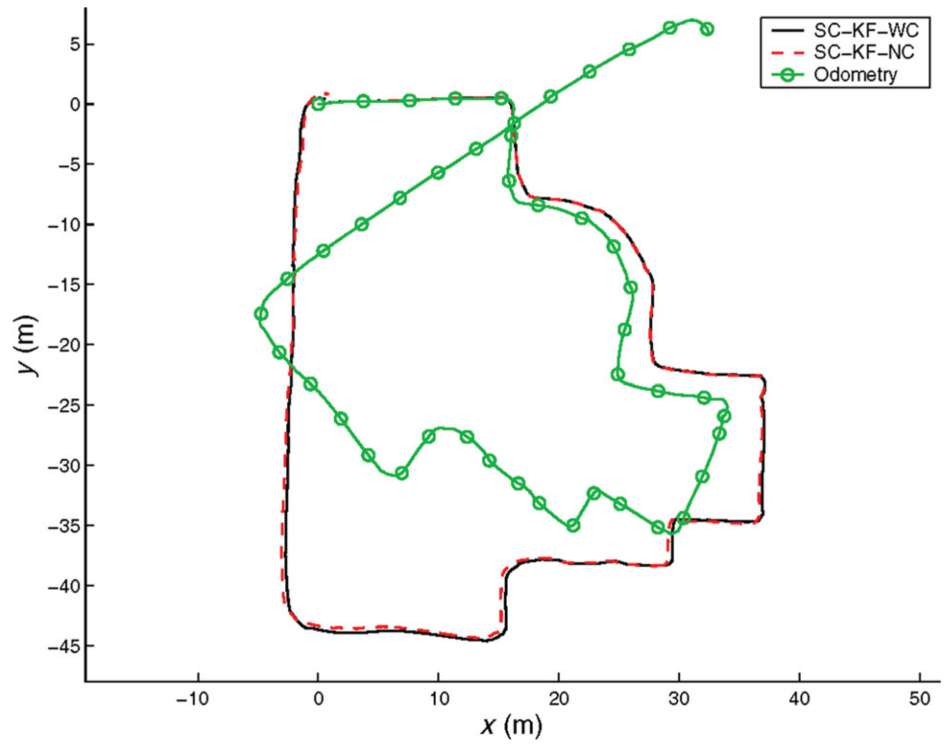


Fig. 14. (Colour online) Odometry-based vs. Kalman filter-based position estimate.[48]

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\,\tilde{\boldsymbol{y}}_k \\ \boldsymbol{P}_{k|k} &= (I - \boldsymbol{K}_k\,\boldsymbol{H}_k)\,\boldsymbol{P}_{k|k-1}. \end{aligned} \tag{7}$$

In an effort to reduce computational complexity and a reliance on real-time, high-bandwidth communication, work has been performed to decentralize the Kalman filter methods to each individual robot, necessitating communication only when the robots detect one another.[47]

Figure 14 shows the difference in position estimate for uncompensated odometry measurements (green) compared to two Kalman filter algorithms (black and red).

Kalman filtering is one of the *geometry-based localization* methods where geometric measurements are used to construct the localization scheme. A second geometric method is *least-squares estimation* where a robot's local feature-based map is analyzed relative to a global feature-based map, and the current position of the robot in the local map is extrapolated into the global map by analytically determining the placement of the local map within the global map.[49] In each geometric method, uncertainty is modeled through a covariance matrix incorporated into the analyses. In Kalman filter, linearized state transition and observation matrices are used to propagate the previous state and observation covariances forward, coupling them with estimates of the process and observation noise covariances. In least-squares estimation, this covariance originates from the propagation of uncertainties of the pose estimates and the residual error of the estimate itself to the pose estimate.

Occupancy-based maps can be used in the *grid-based localization* methods where the pose (position and orientation) of a robot within a space is needed. Specifically, x- and y-histograms are used for pose estimation, while polar histograms are used in the orientation estimation process. Each is extracted from the occupancy grid based on the robot's sensor measurements.[42]

Monte-Carlo localization is an alternate approach for robotic localization, which uses the previous state and control actions to estimate points comprising the probability distribution function of each robot's location within its environment. Because of the previously stated computational limitations of deterministic algorithms in solving the collective localization problem (the problem being NP-hard), statistical algorithms that include random variables allow for better solution convergence.[24]

Markov localization is another class of techniques used in single and MRS. It is built on the Markov assumption that future sensor readings are "conditionally independent" of past sensor readings. The current pose of the robot is stored as a probability distribution over the possible poses of the robot. Localization is performed using two models: a motion model, which utilizes interoceptive sensing to monitor the robots' movement, and an observation model, which utilizes exteroceptive sensing of the robots' environment.[38,52]

Mathematically, the Markov localization manifests in the Bayes filter. For localization, it is desired to determine the robot's pose at time t, $\boldsymbol{x}_t$, based on n known environmental landmarks $\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_n)$ and the control action specified at time t, $\boldsymbol{u}_t$. This new pose estimate will take the form of a probability distribution function, $p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)$. At each pose, sensor measurements at time t are taken as z_t and are based on a probability distribution based on the state at that time, $p(z_t|\boldsymbol{x}_t)$. The Bayes filter, shown in Eqs. 8 and 9 determines the probability distributions of the new state based on the previous control (Eq. 8) and sensor readings (Eq. 9),[16]

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{u}_t) = \int_{X_t} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{x}_t)d\boldsymbol{x}_t, \tag{8}$$

$$p(\boldsymbol{x}_{t+1}|z_t) = \frac{p(z_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t)}{p(z_t)}. \tag{9}$$

A specific localization technique incorporating both the Monte-Carlo and the Markov/Bayesian strategies used extensively in the literature for localization is *particle filtering*. Particle filters have been called "randomized adaptive grid approximations," where the values of the particle change randomly in time, but the weights of each of these particles is updated at each sequential step, capturing the deterministic movement of the robot in the environment and the random noise associated with the measurements of that motion.[31,53,54] Particle filters developed from importance sampling, which discretizes the probability distribution of a sample, then corrects the discrete distribution to adjust the bias due to over- and under-sampled regions. This bias correction is the importance weight utilized in the particle filter.[55]

The Markov assumption is a cornerstone of both Kalman and particle filters. In each, the Markov assumption manifests in the determination of the updated state: the next state is dependent only on the current state, control and noise, and not the time history propagating back from the present values.

Appearance-based or *visual localization* utilizes raw sensor readings and visual maps to compare current readings to those previously stored to localize the robot with minimal real-time image processing. In experimentation, this methodology has been shown to be more robust to environmental changes, such as lighting intensity or obstacle placement, because of the holistic approach to compare images, as opposed to the processing heavy feature-extraction or filtering operations.[56]

Ego-centric localization provides a method for determining collective localization through on-board measurements at each robot to other robots within the team. These measurements are stored as probability distributions (due to the potential for measurement error) and are updated as robots move about the space. These updates are based upon the robot's own subsequent measurements or communicated measurements from other robots.[57]

Range-based localization (also known as the *Marco Polo localization*) collectively localizes the robotic system by utilizing relative position measurements based on the delay in communication between either robots or static markers within the environment. At each iteration, or "slice," each robot or static marker will generate a signal at a known time to be measured by the robots within the system. The difference between the time at which the signal was generated and emitted correlates to the distance between them. While this is a simple and cost-effective method of localization, it requires highly reliable synchronization between the system robots' on-board clocks for precise distance measurements. Typical signals utilized include sound,[58,59] radio[60,61] and WiFi.[62]

Table I. Application of localization techniques.

Technique	Local	Global	Collective
Dead reckoning	X		
Kalman filtering	X	X	X
Least-squares estimation		X	
Grid-based		X	
Monte Carlo	X	X	
Appearance-based	X	X	
Egocentric			X
Marco Polo			X

Table I shows these localization techniques and the type(s) of local, global and/or collective localization to which they have been applied. Kalman filtering is the most flexible due to its design not simply as a localization technique but also as a method to reduce noise in measured signals.

2.4. *Motion planning*

Once a robot is able to understand its environment through a map and place itself within that map using localization, motion planning is needed for the robots to maneuver themselves individually and collectively through the environment. Motion planning in MRS can be broken down into three major classes: formation control, interaction control and coordinated control.

2.4.1. Formation control. The most fundamental form of motion planning in MRS is *formation control*, where the motions of the robots are synchronized to move together in a desired formation through an environment. A key goal of this type of control is to minimize the need for explicit communication between robots, instead allowing local control laws and sensor readings to drive robots in formation without extensive need for global state information. Furthermore, considerations should also be made for teleoperation of the formation by facilitating simple and effective human–system interaction.[63,64]

One of the most common mechanisms for implementing formation control is *leader–follower*. In this scheme, a single robot (designated the leader) follows a predefined trajectory, generates its own path or is teleoperated. The other robots in the system (designated the "followers") are each programmed to maintain a specific distance and orientation relative to the leader, creating a formation. While this method is simple to implement and is highly scalable, its leader is a single point of failure and is poor at rejecting disturbances.[65]

A second strategy, *virtual structure*, generalizes the formation as a single entity and treats each robot as a point on that entity. The path of the entire structure is determined or provided, and the path of each robot is extrapolated from points on that "rigid-body" motion. While a benefit of this method is that it simplifies the path planning for a team of robots into a single formulation (with individual paths extrapolated from the overall solution), in order to ensure the structure is preserved as the robots move, global state knowledge is needed in all of the robots to correct for any inaccuracies that may occur during operation (slippage, environmental disturbances etc.).[66]

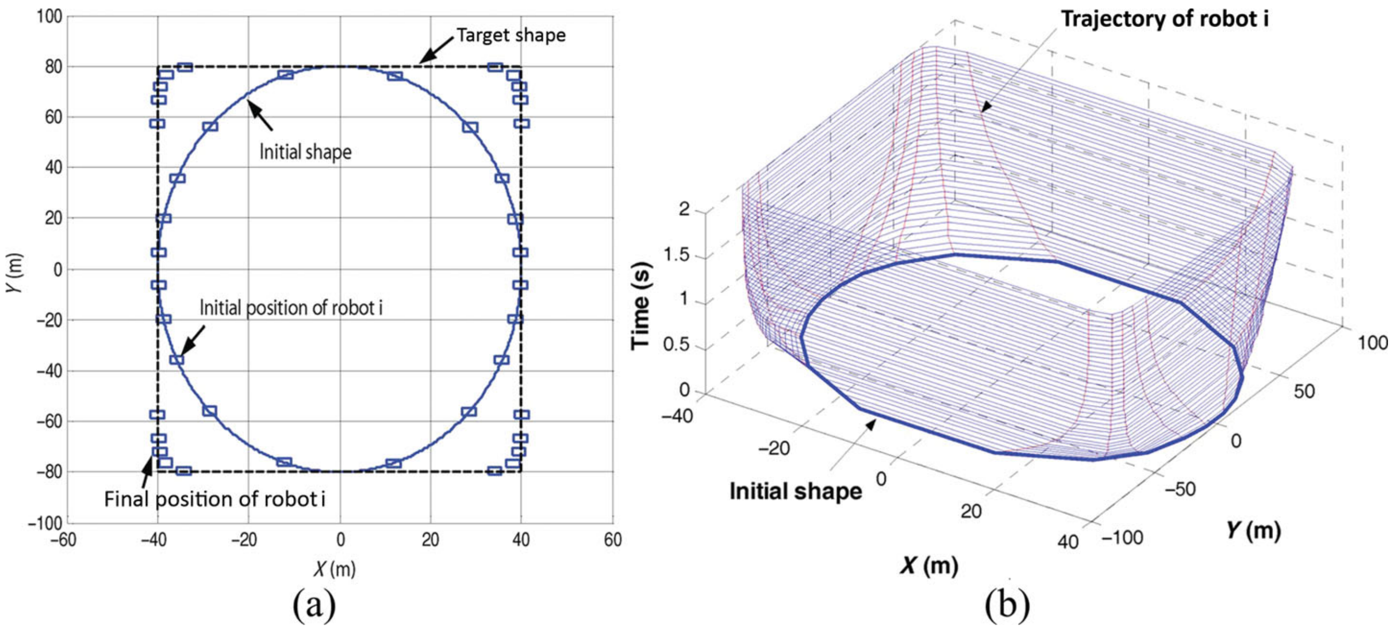


Fig. 15. (Colour online) Time-varying formation change: (a) initial and final configurations, (b) time-varying robotic trajectories.[68]

Behavior-based methods provide a third alternative, where different potential behaviors of each robot are pre-programmed into the controller, and the overall control is derived by adjusting the weight each behavior is given in the robot's controller. A benefit of behavior-based methods is the high level of decentralization associated with it versus leader–follower or virtual structure; however, it is difficult to mathematically model the overall robot behavior based on the weights, making convergence and/or stability analysis difficult.[67]

Synchronous methods, as the name implies, synchronize the relative motions of member robots as they follow individualized trajectories. As a consequence, these methods expand the system's goal beyond simply lowering each individual's trajectory error, but also minimizing an error metric associated with the accuracy of the formation.[68]

It is also important to note that within each of these methods, considerations can be made for time-varying formations. In leader–follower, this becomes a local action on each robot by changing the relative distance and orientation between the leader and each follower. In virtual structure, this can be captured by deforming (shearing and dilating) the structure as it translates and rotates through the space. In behavior-based methods, it is another behavior to add to the weighted sum. In synchronous methods, the individualized trajectories can be modified to allow for changing formations, causing the target for the error metric to change as well. Figure 15 shows an example of this operational mode, where robots arranged in an ellipse transition into a rectangular arrangement.

2.4.2. Interaction control. A second subset of motion control is *interaction control*, where robots follow individualized paths, but also interact with one another. A common situation in which this occurs is robotic *rendezvous*. Rendezvous is when two or more robots must meet at a common point, particularly in environments without a global map or a point common to each robot's local map. Algorithms solving this problem have been developed, even in cases where communication between the robots is unavailable .[69]

Recharging control, also known as *frugal feeding*, is similar to rendezvous, but instead of two or more robots meeting at a single point, one or more agents must visit (if it/they is/are mobile) or be visited by (if it/they is/are stationary) each of the other robots in the team in a prescribed order. The objective of the motion planning optimization may be to minimize travel time, energy spent or an objective function weighing the time and energy.[70]

Beyond the need for robots to physically meet at common locations within the environment, other constraints can be placed on the operation of the network, with the need for control methods to ensure these constraints are met. *Line of sight control* is one such example, where the team of robots must ensure end-to-end connectivity of MRS by sensor vision. Behavioral methods have been used to implement this strategy with varying degrees of success.[71]

2.4.3. Coordination control. A third type of motion planning is *coordinated control*, where the robots implicitly interact through communication, but do not explicitly interact in the environment. In many cases, this explicit interaction would be undesirable. One such application is *patrolling*, where members of a robotic team circulate through an environment to detect changes and/or intruders. A key trade-off in the formation of this class of methods is the desire to employ efficient methods for intruder detection, while still incorporating stochastic behavior so that the robots' behavior is not entirely predictable by the intruder. A common mechanism for modeling patrolling behavior is game theory, where the robots and intruder(s) are the participants in the game, with the strategy of the intruder to avoid detection and the strategy of the robots their control system(s).[72,73]

A second application is *complete environmental coverage control* or *region filling control*, where the team of robots is

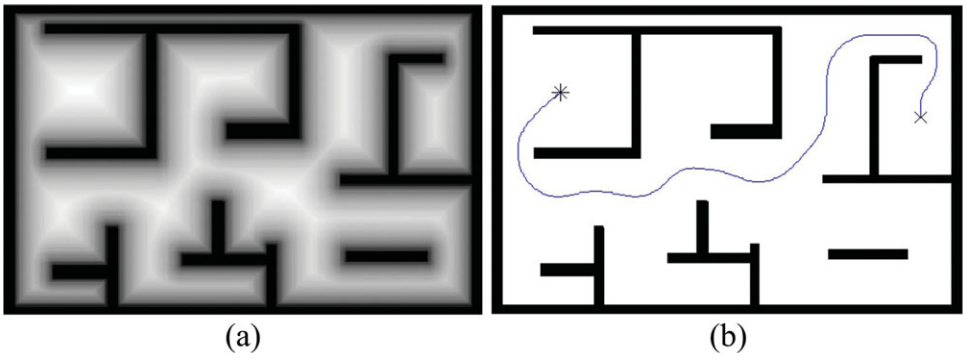


Fig. 16. (Colour online) Potential field-based path planning: (a) potential field of map known *ab initio*, (b) path generated based on potential field between two points.[78]

instructed to visit the entire environment as quickly and/or efficiently as possible. In this application, it is desirable to have minimal overlap of the robot's paths in order to cover the environment more quickly and efficiently.[74–76]

2.4.4. Obstacle avoidance. Beyond these three classifications, there are several other considerations that must be made when navigating through an environment. One such consideration is avoiding obstacles. Generally, the goal of obstacle avoidance is to create an area in the vicinity of each robot that avoids overlap with environmental obstacle and other robots to allow it the flexibility to move through the environment. For example, in non-holonomic robots, this may be to allow for the robot to effectively turn without encountering obstacles. However, other solutions for this problem, such as potential field methods and collision avoidance functions, have also been studied.[64]

Potential field methods utilize a potential function to "push" the paths of robots away from obstacles by modeling them as repulsors or sources in the field, and "pull" the robot paths toward goal points by modeling them as attractors or sinks. While this does provide a simple mechanism for ensuring obstacle avoidance *a priori* to the actual exploration, it causes sub-optimal behavior when the robot is not in danger of colliding with an obstacle due to the effect the sources and sinks can have at medium to far distances away from obstacles.[26,77] Figure 16 shows a typical potential field in a fully mapped environment and its corresponding path between two points.

Collision avoidance functions are a more local control method where obstacles are only considered when they are present within the sensing radius of the robot in danger of collision. If the corrective action is not sufficient to avoid the obstacle as soon as it is detected, it is significantly increased as the robot approaches the obstacle. This type of method is more effective for maintaining the optimal trajectory unless there is imminent danger of collision, but it requires more on-line computation and highly active sensing to detect the obstacles.[64,79]

Real-time methods allow an MRS to adapt to a dynamic environment in which obstacles may be moving along unknown trajectories. Critical to successful implementation of this method is for the robotic system to be able to sense the new obstacles and to leverage that data to optimally adapt the current motion plan for the robots in the system to avoid a collision.[80]

2.5. Simultaneous localization and mapping (SLAM)

In practice, the divisions between mapping, localization and motion planning are not as clear-cut as the previous three sections imply. Particularly in unknown environments, the tasks of localization and mapping become inextricably coupled, resulting in the need to develop SLAM algorithms for MRS. For SLAM in MRS, a team of robots is tasked with utilizing exteroceptive sensors to determine the placement of obstacles within an environment, storing those obstacles within local and/or global maps and simultaneously localizing the robots with respect to their local and/or global map, and/or one another.

Mapping with external localization, while not precisely a SLAM methodology, provides a baseline for separating decoupled mapping and localization from traditional SLAM techniques. In this method, the external localization may be provided by an overhead camera or by beacons/reflectors of known size and location placed within the environment. While this method is simplistic, its benefit within the laboratory is its capacity to decouple the mapping operation from localization, while still reaping the benefits of localization to study mapping algorithms.[15]

As in localization, *Kalman filters* can also be applied to SLAM to synergize the noisy exteroceptive and interoceptive sensor readings simultaneously. In essence, this is an extension of the localization operation, but now the map of the environment is unknown and the only obstacles for which exteroceptive sensors can be used to measure distances are those previously discovered and mapped by the robot.[1]

Sparse extended information filters are an extension of Kalman filters, where the information form of the extended Kalman filter (known as the extended information filter) is used to reduce the computational time required for calculation from a quadratic dependence on the number of features in the map to a constant time independent of feature count. This feature of quantity independence is achieved through the approximation of the environment as sparse in the extended information filter where the environmental factors that strongly affect the current SLAM estimate are considered in the estimate instead of the mandatory coupling to all factors used in the extended Kalman filter.[81]

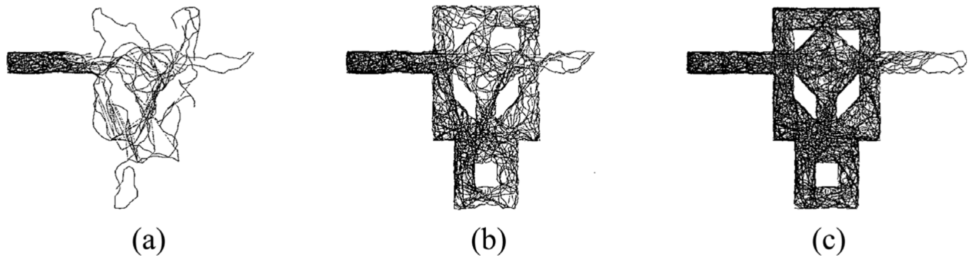


Fig. 17. Evolution of a diffusion map.[59]

Maximum-likelihood estimation, an extension of *expectation maximization*, utilizes a set of observations to compute the likelihood of each member of a set of pose estimates and obstacle locations. Mathematically, relationship is represented by Eq. 10: the conditional probability of a set of observations $\boldsymbol{M}$ given a set of measurements $\boldsymbol{X}$ is desired, with this value computed by taking the product of the conditional probabilities of obtaining an individual motion observation given an initial robot pose estimate of $\boldsymbol{x}_i^t$ and a final pose estimate of $\boldsymbol{x}_j^t$. This process is performed among the robots in an MRS, and then numerical optimization is used to determine the most likely set of poses and obstacles for the set of available observations. While this is a robust method for performing SLAM, it suffers from heavy computational load required to evaluate and optimize the pose. Moreover, it is sensitive to erroneous sensor measurements introducing permanent localization and/or mapping error with few methods to correct for the error later in the algorithm,[2,82,83]

$$P(\boldsymbol{M}|\boldsymbol{X}) = \prod_{i,j} P\left(\boldsymbol{m}_{ij}|\boldsymbol{x}_i^t, \boldsymbol{x}_j^t\right). \tag{10}$$

Topological SLAM, or *T-SLAM*, operates entirely within a topological abstraction of the environment. In Choset *et al.*,[51] a generalized Voronoi graph (GVG) is generated as the robot moves through the environment, and the robot continually localizes itself within that graph. A key assumption of this work is that the robot is able to determine without any previous state knowledge whether it is in a previously visited/mapped node through the current sensor readings. However, in many environments, such as office buildings, spaces are more homogeneous and repetitive.

Hierarchical SLAM extends the idea of T-SLAM by utilizing hierarchical maps with topological and geometric components. A single hierarchical map and two or more geometric maps are utilized during the SLAM algorithm, with localization in the hierarchical map used to determine the geometric maps that correspond to the robot's location within the environment. Then localization is performed in this subset of maps, reducing computational complexity and improving fidelity in areas where two or more local geometric maps overlap. Unlike other SLAM algorithms, this method requires a partial understanding of the layout of the environment *a priori*. However, it does provide for updates to the map as a part of the SLAM procedure.[29]

Evolutive localization uses a stochastic search method to determine the best estimate of the localization of the robot within the map as it is being generated. Interoceptive and exteroceptive measurements are synthesized in this process, with localization occurring after discrete motions of the robot based on the odometric measurements. While mapping is incorporated as the robot moves through the environment, it is critically important when cycles are detected and the robot returns to a previously mapped area.[78]

2.6. Exploration

While SLAM is a critical aspect of multi-robot functionality, it does not incorporate the motion planning necessary to traverse an unknown environment to ensure complete and efficient mapping, nor does it provide a metric for assessing the efficiency of mapping an environment. The combination of SLAM and motion planning is considered *integrated exploration*, or simply exploration. Exploration can be considered for both known environments, where the goal may be to search for intruders or to monitor change within the environment, and unknown environments, where the goal may be to efficiently map the obstacles or find a target location.

The three primary considerations for an exploration algorithm are its efficiency, accuracy and adaptability. Efficiency relates to the algorithm's capacity for driving an MRS through the environment as quickly as possible. Accuracy relates to the algorithm's capacity for reliably building an accurate map, and adaptability relates to the algorithm's suitability for mapping different types of unknown environments, such as wide open spaces with sparse obstacles or cramped offices with limited visibility and little space for navigating.[84]

In fully known environments, *predefined trajectories* provide a simple mechanism for robotic control where the trajectory is planned *a priori* utilizing the known map, with a goal of either verifying the map or discovering new information about the environment unrelated to the arrangement of the environment itself. While the ease of implementation is a significant benefit, using predefined trajectories does require full knowledge of a global map before exploration takes place, and if there is a possibility for additional unknown obstacles in the environment, obstacle avoidance methods (Section 2.4.4) must be incorporated into the robotic controller to reduce the potential for failure.[84]

Diffusion mapping is an exploration mechanism where large numbers of simple robots map an environment through highly redundant localizations and line of sight mappings throughout. A random-walk-type motion planner is used to allow the robots to reach every point within the environment, albeit theoretically and eventually, as shown in Fig. 17. While it is a robust and fault-tolerant method, it is highly inefficient and time-consuming, and any small, uncompensated error in localization can significantly affect the quality of the map.[59]

Greedy mapping methods drive a single robot or members of an MRS toward the nearest unexplored area within either local or global maps. While this method is simple to implement and is inherently decentralized, it does not account for potential overlap of actions within the MRS. However, it has been found that while this behavior is sub-optimal, particularly in MRS, it still accomplishes the task within a reasonable distance travelled.[85] The greedy mapping strategy has been improved by utilizing an a-optimal objective function (the sum of the covariance matrix's eigenvalues) versus the conventional d-optimal objective function (the product of the covariance matrix's eigenvalues).[86]

Frontier-based methods drive the MRS members toward the frontiers of the map between the known and unknown environmental spaces. The expectation is that if a robot moves toward the frontier, it will detect new, unexplored regions of the environment.[87,88]

Potential field methods in exploration are an implementation of the frontier-based methods and are similar to those in motion planning/obstacle avoidance. Attractors are added to the unexplored frontier of the robots' global map to encourage robots to move toward unexplored areas and repulsors are added to each robot to encourage separation of the group to maximize coverage. Coupled with the previously discussed repulsors at obstacles, this allows for an effective strategy designed to pull robots out of previously explored areas while still preserving an ability to avoid obstacles and other robots as they are detected.[77,88]

A challenge in utilizing potential field methods is their ability to create local minima in which the robot can become "trapped" either at a single point in the environment or on a loop circulating through a specific area. Several methods have been proposed to address this issue. Adding *random walk*, as discussed previously, includes a stochastic model to drive the robot off of its "prescribed" path and avoid the possibility of becoming trapped. Utilizing *artificial potential field* strategies continuously "scan" each robot's potential field map for local minima and modify the field to remove these minima as necessary. Avoiding minima by *adding repulsion* actively modifies the potential field as the robot moves through the explored area of the map, adding repulsion to previously visited points. Switching to *wall following* when the robot detects it is trapped is another easily implementable and intuitive scheme for escaping minima.[88]

A potential field can be generated using the *extended Voronoi transform* (EVT), and plan paths with this transformation using the Voronoi fast marching. EVT utilizes a map (previously generated or in progress) and creates a grayscale version of the map dark near obstacles. The log of this map creates a potential field that will drive the robot away from known obstacles and, by extension, toward unknown areas.[78]

Next-best-view methods, first investigated by Connolly in 1985,[89] directly utilize two criteria to determine the next most optimal position for creating a map: the anticipated information gain at that position and the estimated distance that point is away from the robot's current location. In order to translate this formulation into a workable algorithm, some form of optimization is needed to optimize the relative features in determining the optimal point.[18]

One such optimization scheme is *ad-hoc utility function optimization*, where the two features are combined in a weighted utility function and computed for a set of candidate points. This value, along with considerations made for its reachability from the current location, is used to determine the most optimal of the candidate solutions.[90] A second scheme for optimizing the next target point is *multi-objective optimization*, where candidate positions are generated and the various features are evaluated at each point. Then the Pareto-optimal candidates are isolated, and the rest discarded. Among the Pareto-optimal candidates, a metric is then used to determine the solution that is nearest to the ideal solution comprising minimal distance from the current location, maximum information gain and maximum overlap.[84]

Anchored wanderer methods utilize MRS with members designated as either the communication anchor or a wanderer. The anchor remains in its initial location, and the wanderers explore the space around it one at a time. As each wanderer moves, it must maintain a line-of-sight directly or indirectly (through another robot) with the communications anchor. When this line of sight breaks, an obstacle is known to be between the anchor and wanderer, and can be incorporated into the map. The wanderer is then reversed to restore the line-of-sight, and the next wanderer commences its exploration.

A specialization of this method in environments where some of its properties are known *a priori* is the *quadrant-based anchored wanderer*. In this method, the environment is known in terms of four quadrants, and there is a target point in one of the quadrants to be discovered. The operation continues as before, but bias is incorporated into the wanderer's motion toward the quadrant with the target point to hasten its discovery.

Sensor-based random tree or *graph methods* incorporate a probabilistic or randomized motion planner with a topological map (called the sensor-based random tree/graph) capable of storing information at each node such as the local safe region around that point. The connections between nodes model navigable paths within the environment, and the frontiers at the fringes of the graph are preserved at the boundaries of the local safe regions in relevant nodes. The motion planner is not entirely random; generally, it is biased to drive the robots toward the frontier and away from one another.[33,91,92]

Cellular decomposition or *segmentation methods* provide a mechanism to ensure complete coverage of an environment by decomposing it into individual cells or segments. During this operation, the members of the MRS exhibit one of the two behaviors: global exploration to create the cells or segments, or cellular exploration to fully explore and map the created cells/segments individually. An example of a decomposed environment is shown in Fig. 18. To accomplish this task with an MRS, control is quasi-decentralized, with the system broken down into sub-teams tasked with global exploration or cellular exploration, and communication occurring only within teams.[4,93]

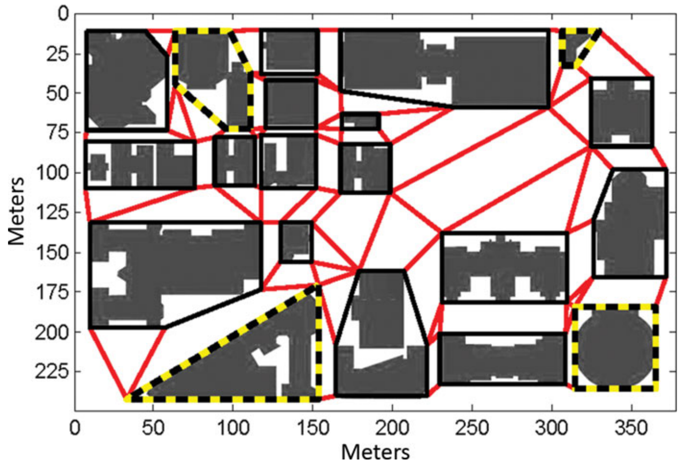


Fig. 18. (Colour online) Cellular decomposition illustration.[94]

Hill climbing methods map the unknown areas of environments by utilizing an entropy map with high entropy in unknown areas and low entropy at mapped areas. The gradient of this entropy map provides a basis for motion planning through climbing the "hills" of high entropy. A hierarchical map is used to map obstacles in an occupancy-based grid and avoid local minima in a topological map.[20]

Bidding or *free-market methods* mimic an auction or an econometric model to assign tasks to different robots. When a task is formulated and subdivided into smaller sub-tasks, the robotic members of the MRS will construct bids for each of the sub-tasks based on local cost-benefit analyses. Potential costs include estimated completion time, distance travelled and/or energy consumed. Potential benefits are generally limited to various forms of information gain. While localized "profit" maximization strategies generally provide a high-efficiency solution, it may lead to sub-optimal solutions, as seen in Fig. 19. Furthermore, the network topology is limited due to the need to communicate bids to a centralized "auctioneer" to assess them and assign tasks.[82,95–97]

Consensus-based methods strive to formulate a consistent representation of the environment across all members of an MRS to allow for localized motion planning based on the global representation. Once a consistent environment is available across the MRS, consensus then helps to coordinate the motion planning strategies to reduce overlap of information gain. While a specific area of the map may provide the greatest information gain to all members of the MRS and going there would maximize each robot's potential information gain, in the global sense this is a sub-optimal behavior. Consensus can also be used to facilitate bidding/free-market methods in providing a uniform environmental representation for more consistent bids across the MRS.[98]

3. Benefits, challenges and tradeoffs

Patterns of benefits, challenges and tradeoffs begin to emerge from the analysis of the current methods utilized in the operation of MRS. The primary trade-off relates to the use of the system's members in *parallel* or *cooperatively*. Parallel use implies that each robot performs tasks individually. However, this does not imply that tasks are performed independently: communication still ensures minimal duplication of efforts, unless duplication is preferable. Cooperative use implies direct physical interaction between within the workspace, from joining together to form a superstructure to transporting a load greater than any individual's capacity.[11]

In both situations, MRS inherently benefit from the *concurrent operation* of their members and the potential for *specialization* facilitated by a distributed system. Concurrent operation of individual robots (in parallel systems) or MRS sub-teams (in cooperative systems) speed up the execution of tasks beyond what any single robot or group of robots operating independently could perform.[87,96] Furthermore, the utilization of multiple robots in a system allows for distribution of necessary capabilities for a task among the robots, thereby creating a team of specialists that each can perform a single task exceptionally, versus a single generalist more prone to failure, more constrained in what it can accomplish and not expert in performing any single task.[97]

Furthermore, MRS address the issues of *reliability* through *redundancy* of capabilities, reducing the potential effect of *faults* on the system. Because MRS can be designed without single points of mechanical, communicative or control failure, this increases their reliability. A common method of eliminating single points of failure is to include redundant copies of each type of agent within a team.[87,96] However, with certain task redistribution and/or team reorganization strategies, systems can be scaled down below complete redundancy. When faults do occur, it is critically important that they are detected so that the system can compensate. A central challenge in this endeavor is distinguishing between faults, which are a systematic failure, and disturbances, which are unanticipated environmental effects on the system.[99]

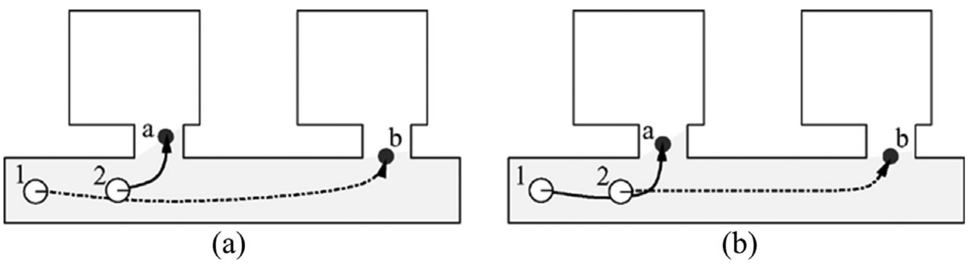


Fig. 19. Local optimality vs. global optimality: (a) locally optimal solution for robot 2 at the expense of robot 1, and (b) optimal solution to minimize overall travel time.[87]

3.1. Decentralization

Decentralization refers to the distribution of computation and control from a centralized agent or operator to the individual member of the MRS. Centralized strategies synthesize readings from robotic agents into a single framework and determine tasks to be performed by the team. In this situation, the team is thought of as a single entity with several degrees of freedom. However, producing optimal solutions for this "single entity" framework is computationally difficult and is not scalable. Fully decentralized strategies allow for the achievement of global goals without explicit communication among MRS, including efforts such as localization.[97,100]

Practically, the distinction between centralized and decentralized control is more analog than digital, with different strategies incorporating elements of both. One manner in which this manifests is in how an MRS performs *task allocation and re-allocation*. Franchi *et al.*[33] describe an exploration algorithm where the discovery of new environments is shared with other robots within a system. Because these new features are now shared globally, local algorithms for path planning can be deployed without needing to coordinate that computation with other robots. Ghaderi *et al.*[101] investigate how task reallocation must occur both locally and globally in response to minor and catastrophic failures within the system.

Research has also been performed to find distributed algorithms that can mimic a *centralized computation* with minimal external communication to each individual robot from other robots. Roumeliotis and Bekey[47] studied the distributed Kalman filter that would perform filtering on each robot and would only require inter-robot communication when localizing with respect to each other. Leung *et al.*[102] considered the problem of collective localization through global state estimation, with communication only necessary to compare and correct global state estimates.

3.2. Coordination

Coordination refers to the communicative features of MRS operation. Two types of data are transmitted through communication: situational awareness, such as maps or sensor readings, and control commands, such as planned trajectories.[98] Furthermore, this coordination may either be (1) passive/implicit, where the robots attempt to influence one another solely through their actions (which are then sensed by the other robots), while planning is still performed independently, or (2) active/explicit, where communication is utilized during the planning process to communicate information to implement a global strategy.[22,103]

Beyond the specific data being communicated, the design of communication networks directly affects the potential architecture of an MRS.[104] This design comprises direct communication links between robots[96] or multi-hop schemes where two robots communicate through an intermediary.[105] A common but unrealistic assumption for algorithms studied in simulations or laboratory environments is unlimited communication bandwidth and perfect synchronization. However, in practice, there is an inevitable and sometimes significant delay (particularly when utilizing multi-hop communication).[98]

A major trend has been to minimize the need for explicit communication within the MRS. While it may be simply to reduce delay, there are some circumstances where explicit communication may be impossible (due to interference or environmental obstruction) or pose a risk (such as military operations).[94] Algorithms have been designed to reduce[57] or eliminate[94] this need entirely when necessary.

In an effort to improve fault tolerance, considerations must be made for communication failure. For example, if one robot is expecting a transmission from another and does not receive it, the robot should take some alternate action as opposed to waiting indefinitely. Ideally, the robot would also be able to alert the other members of the MRS and/or the system operator of the communication failure to improve system robustness and allow the fault to be addressed.[106]

3.3. Cooperation

Cooperation refers to the synergy of decentralization and coordination in the operation of MRS. This most clearly applies when addressing the localization problem, where the control of the system as a whole depends on each robot's individual measurements made in real-time. Localization improves when a variety of different sensor measurements are included in the localization scheme; MRS cooperation allows incorporation of locally measured and processed data for each robot into the localization strategies for other robots, providing a global benefit based on the distributed processing of data and internal communication of its information.[87,107]

Franchi *et al.*[108] provide an example of a strong framework to encourage cooperation between robotic pieces. In the beginning of a task, each robot within the MRS has an individual goal to work toward with understanding of the other robots' goals to minimize overlap. Once that individual goal has been achieved, the robot then cooperates locally with another robot to help it achieve its goal, and so on until the overarching task is completed.[108]

While in most circumstances an increase in the number of robots in the team increases the effectiveness of the team, there is a diminishing return associated with adding additional agents. For example, in an exploration mission, effectiveness might be reduced in crowded environments by robots having to take too many detours to avoid other robots.[93] A key challenge in designing an MRS is to find the optimal type and quantities of robots within the team where the marginal benefit of any additional piece is not less than the marginal cost associated with operating it.

Communication and control are also intertwined in the requirement for many systems that the MRS remains "connected" either by line of sight constraints or by maximum relative distances. This connectivity, often necessary for communication, requires a control-based solution that must be continuously considered during motion planning for the MRS.[105]

3.4. Localization accuracy versus area coverage

In exploration activities, there is a central tradeoff that dominates the performance of a system: the tendency for localization accuracy to decrease as the rate of area coverage increases, and *vice versa*. This inverse relationship can be classified in a tradeoff of two different types of information gain for the system: the gain of continuous information about the state of the system as estimated by its localization scheme versus the gain of discrete information about the environment in which it operates.[14] One manner in which this has been addressed is through the use of "sentry" robots that either remain stationary[109] or move with the system[59] with the sole purpose of observing the other members of the MRS to localize them more effectively.

Efforts have also been made to control the relative weight given to localization and coverage considerations within the algorithm itself. This allows the system or operator to choose when fidelity of location is more or less important than rapid deployment and coverage.[88]

3.5. Additional challenges

At a high level of abstraction, there is a need to communicate the desired behavior or task to an MRS. There are two fundamental approaches to accomplish this goal: bottom-up or top-down. Bottom-up approaches utilize local interactions between robots to create global behaviors of the system. The operator customizes a controller's methods and parameters to break down the task among robots and determine their modes of interaction. Alternatively, top-down approaches directly communicate these goals to the MRS and allow for an automatic process to determine the necessary communication protocols and feedback controllers based on the electromechanical architecture of the MRS. While the bottom-up approach is significantly more common than top-down, top-down has the potential to revolutionize the approach to address MRS control strategies through the capacity to pose more rich global problems to the system.[110]

At a lower level of abstraction, the generation of state estimates is tightly coupled to control of the MRS at local and global levels, with consideration of this coupling greatly improving both estimate and control. For example, in exploration, the most optimal path in terms of travelling distance may be through a large area of open space without obstacles. However, without external obstacles to be included in the localization estimate, the localization error will increase – hardly an optimal behavior in a broader sense. Therefore, it may be more desirable to track closer to walls or other obstacles to retain strong localization, which will ultimately assist future control via the motion planning algorithm.[22]

Furthermore, while local map overlap has been treated as a universal negative throughout various considerations of efficiency, for many MRS, a certain amount of overlap is necessary and desirable. When local maps overlap with one another or with the global map, their correspondence for merging becomes more robust than if relative measurement data were to be used alone to position the maps in the global space.[84,88]

A common issue within the literature is the focus on robust and novel algorithms solely applicable in laboratory environments with over idealized conditions. For example, in most maps there is an absolute distinction between spaced blocked by an obstacle and clear space. However, in many environments it is common for obstacles to be deformable (e.g. curtains/bushes/branches) or movable (e.g. doors/boxes/rocks). If these algorithms had a mechanism for detecting potentially deformable or movable objects (through observation or autonomous experimentation), they would provide drastically improved capabilities.[111]

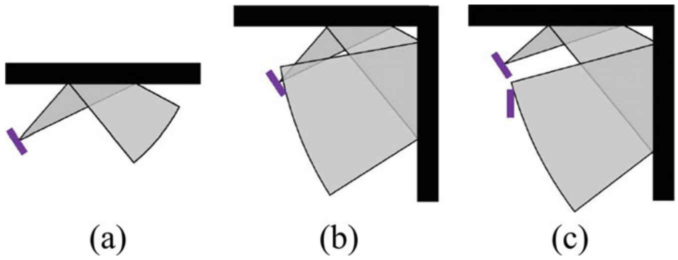


Fig. 20. (Colour online) Sensor interference: (a) oblique reflection, (b) high-order reflection and (c) cross-talk.[83]

Furthermore, within the field of MRS, there is a tendency to model robots either as purely holonomic or to use a single non-holonomic robotic architecture for the entire team. However, to fully attain the benefits of MRS, there must be a capacity to use robots within a system with drastically different architectures and locomotive capabilities.[112,113]

When utilizing MRS, the potential for interference between the robots should not be overlooked. In its simplest manifestation, this interference may be from too many robots trying to navigate a small subset of an environment with many obstacles.[87] However, in terms of sensors, interference can also occur due to unintended environmental effects on how the sensor operates. For example, Fig. 20 shows three types of sensor interferences possible with sonar sensors: in Fig 20(a), an obstacle is not detected because the sonar wave did not reflect back directly to the sensor; in Fig 20(b), the distance to the obstacle is overestimated because of the indirect path taken by the wave and in Fig 20(c), cross-talk between sensor occurs because one sensor is detecting the wave generated by the second sensor. Cross-talk can then be extrapolated beyond two sensors on the same robot to a robot in the line-of-sight of the sensor.[83,87]

Beyond any internal considerations, a significant challenge in utilizing MRS in real-world, real-time applications is the manner in which they *interface with operators*. As discussed previously, systems are often designed bottom-up with pre-programmed strategies for operation in a variety of circumstances. However, in many mobile applications of mobile robots, there is a strong desire and/or requirement to incorporate an operator "in the loop" to monitor the system's behavior and correct for undesirable action, or to supplement the autonomous function with actions that could not be pre-programmed. The key challenge is facilitating this relationship in simple and intuitive ways for the operator to reduce the potential for failure due to preventable causes.[27]

The relationship between scalability of systems and their operation is another challenge: In systems with more than two or three robots, a single operator cannot be expected to monitor each during a mission, and the use of multiple operators complicates operation. Strategies are needed to preserve the benefits of teleoperation in large-scale systems while still providing the necessary oversight over their operation.[114] Furthermore, research in bilateral control with allowances made for delay has also been attempted to better mimic real-world operating conditions and allowing for stable and robust operation when these bilateral considerations are made.[64]

4. Practical Implementation

Beyond the desire for researchers to push the boundary of scientific knowledge, there are numerous applications of MRS research in both specialized sectors and everyday life. Examples of these applications, which have seen serious consideration in the literature, are presented in Section 4.1. Section 4.2 describes sensor technologies commonly utilized to instrument MRS for tracking position and orientation.

4.1. Fields of application

Search and rescue is a commonly addressed application area of MRS due to the ease in which the problem can be posed to a robotic system, the ease with which MRS can scale to provide additional units to improve search speed and accuracy and the potential harm associated with human participation in search and rescue in some scenarios (e.g. collapsed building, inclement weather etc.). Searching has been categorized into three types: efficient, guaranteed and constrained. Efficient search attempts to locate a non-adversarial agent within the environment in the minimal time by covering maximum area as quickly as possible. Guaranteed search attempts to locate a target (adversarial or not) by ensuring evasion is impossible as MRS traverses the environment. Constrained search imposes one or more conditions on efficient or guaranteed search strategies such as a maximum relative distance between agents or line-of-sight restrictions.[22,115]

Military and police operations benefit from many of the same factors as search and rescue when applying MRS, such as relative ease of posing problems and the reduction of human harm. Furthermore, utilizing robots can reduce the need for humans to perform tedious and uninteresting tasks by automating those processes, similar to how fixed robots have revolutionized manufacturing. *Intruder detection/surveillance/patrolling* is an example of this, where robots utilize maps to plan routes to intercept intruding agents within an environment. Strategies generally incorporate optimal coverage algorithms to maximize efficiency coupled with random walk elements to increase unpredictability of the team. The adversary's motion relative to MRS itself has also been considered through techniques such as game theory to better plan the robots' strategy. Guaranteed search strategies can also be used to "trap" intruders in environments as robots patrol.[72,73,100]

Other examples of military and police operations include reconnaissance and biohazard discovery. *Reconnaissance* consists of a group of small robots covertly navigating an environment to gain sensory information from that environment. Reconnaissance benefits from utilizing MRS because of the redundancy associated with using multiple agents in parallel on the same task; if a robot fails, the entire mission is not jeopardized.[116] *Biohazard discovery* utilizes robots to explore buildings in which there are suspected biohazards and remove/neutralize any that are discovered. The reduction in potential harm to humans whom otherwise would have to execute this task is obvious, but MRS can also be outfitted with a suite of different sensors and actuators to better detect hazards than could be done by a human, even with portable instrumentation, due to automated processing and categorizing of sensor data.[117]

Planetary exploration refers to the search of unknown environments with extreme environmental conditions for specific items or features. This classification may refer not only to exploration of other planets and objects in space but also to certain areas of Earth. One such location is Antarctica, where robots have been used to search for and classify meteorites. Use of MRS to accomplish this goal promises to significantly expedite the process in terms of area covered as well as reduce the extent to which humans must be involved in the process.[118]

Mobile sensor networks allow MRS to be utilized to monitor an environment for a variety of properties, from electromagnetic fields[119] to the presence of oil in a body of water.[120] The capacity of MRS to dynamically measure these properties in multiple time-varying locations allows for dynamic correction of deficiencies. For example, sensor networks that monitor wireless signal strength could deploy mobile wireless "hotspots" to correct for any deficiencies, or sensor networks that monitor oil spill intensity can alert cleaning crews to the areas affected the most by the spill.[120]

Load transport is a natural extension of single-robotic load transport; however, the collective capacity of MRS provides greater flexibility in terms of the magnitude of load itself and its geometry. In addition, MRS systems allow higher fault tolerance during the transport operation; if one of the robots fails, the others may be able to compensate for that failure by redistributing the load. However, load transport is a highly cooperative task, with fast and efficient real-time control necessary to ensure sufficient force on the load to keep it lifted and stable as the system moves.[10,101,121]

Service robotics incorporates a broad category of applications aimed at improving society by automating or simplifying assistive actions. An example can be seen in robotic *mowing* and *floor cleaning*. In both, the member(s) of an MRS must visit each location within an environment at least once with no margin for error. To facilitate use by the greater public, this needs to be accomplished with minimal input from an operator. A challenge prevalent in these types of operations is the dynamic environment such that people may be moving around the house or yard as it is cleaned/mowed. As the robots traverse the environment, they must constantly sense for obstacles, even in previously covered/mapped areas, and in cases where previous obstacles disappear, ensure that area is cleaned or mowed.[75,122]

Large-scale assembly and *construction* tasks of structures, such as terrestrial buildings or planetary habitats, benefit from the flexibility in designing heterogeneous MRS with agents capable of performing specific tasks in the execution of the operator's goals. In order to perform these actions, the system needs both high load capacity for moving and dexterous control over materials for placing and connecting them.[7]

Mine explorations allow an MRS to explore an abandoned and/or potentially dangerous mine without risk to human life. A key challenge in exploring mines is the lack of GPS data for localization, requiring reliance on intero- and exteroceptive

Table II. Benefits and shortcomings of current sensing technologies.

Sensor	Benefits	Shortcomings
Encoders[124]	• Often already included in system for motor control. • Simple transformation to determine position/orientation from axis rotations.	• Position prone to drift due to accumulated errors in kinematic model parameters or wheel slip. • Velocity determination requires numerical differentiation that introduces additional noise.
Inertial navigation system[125,126]	• Provides both position and orientation measurements. • Integrated gravity/magnetic compass stabilize orientation calculation.	• Position determination with second-order integral highly prone to drift. • Accurate initial position needed to know absolute position in environment.
GPS[50,126]	• Provides absolute position with a known margin of error. • Not subject to error accumulation over time.	• Not available indoors/underwater/in confined spaces. • Subject to RF interference.
SONAR/ultrasonic[127,128]	• Provides a scalar distance measurement from the sensor to object. • Signal easily manipulated to provide omnidirectional sensing.	• Subject to interference if multiple sensors are used simultaneously. • Reflection of signal wave dependent on obstacle surface material/orientation.
Laser scanner/range finder[129]	• Similar to, but more accurate than, SONAR/ultrasonic sensors. • May utilize a variety of techniques to determine distance – time-of-flight, interferometry etc. • May return the distance to a single point (rangefinder) or an array of distances (scanner).	• Highly dependent on reflectivity of obstacles. • Minimum and maximum sensing distances limit operational flexibility.
Optical camera[130]	• Images store a large amount of information for use by system. • 3D maps may be extracted from multiple 2D images using stereovision.	• Image-processing and data-extraction techniques. • High computational cost to post-process images limits response time.

sensing for SLAM operation. However, unlike other indoor applications without GPS, mines are less distinct in their features for accurate localization. Collaborative localization allows for greater localization accuracy by increasing the sensor data available for the SLAM algorithm.[123]

4.2. Sensor technologies

While technologies utilized with specific techniques have been discussed throughout the review, it is helpful to consolidate the information to compare the benefits and shortcomings of various technologies. Table II summarizes the six most common methods used to instrument MRS for system localization, SLAM and integrated exploration (among other operations), and present benefits and shortcomings of each. These do not include application-specific sensors that vary depending on the desired system capabilities (e.g. radiation sensors for nuclear plant maintenance).

Encoders measure shaft rotation within the robot, which is then correlated to the distance moved by each wheel/track and in turn the motion of the robot. Inertial navigation systems utilize three-axis accelerometers and rate gyroscopes to estimate a robot's position and orientation by integrating the acceleration/angular velocities. GPS utilizes signals from orbiting satellites to trilaterate the robot's position based on the calculated distance to each satellite. SONAR and ultrasonic sensors utilize sonic/ultrasonic longitudinal waves to measure the distance to obstacles or between robots by measuring the time-of-flight of the wave from emission to detection. Laser scanners/rangefinders operate similarly to SONAR/ultrasonic sensors, but utilize lasers as the signal and may utilize additional benefits of electromagnetic waves (phase shift, interference etc.) to extract additional information about the obstacle. Furthermore, laser-based sensors may return a pointwise distance (rangefinder) or an array of distances resulting in a 3D map (scanner). Optical cameras generate 2D images for processing by the robot to extract information about the environment.

5. Conclusion

This paper provided a review of the methods, benefits, challenges, tradeoffs and applications of utilizing MRS. Based on the reviewed literature, several areas of improvement for future work exist for current shortcomings in the literature.

On a high level, few considerations have been made to consider the selection of members of MRS "teams" to accomplish a task. Currently, an *ad hoc* approach is applied to either adapt a team for which architecture and control have already been established or to custom-design a team suited specifically for a task. If a more unified framework was developed to analyze tasks, it is possible that a team could self-select its own members to accomplish it. Furthermore, in order to allow the greatest flexibility when designing a team, more consideration needs to be placed on the design of control and cooperation strategies for robots with drastically different architectures and types of holonomic constraints. While research has been performed on the consideration of MRS each with the *same* non-holonomic constraint, little has been performed in integrating robots with different non-holonomic constraints with one another of other holonomic robots.

Beyond the internal considerations of communication and coordination, the current research is relatively sparse in terms of analyzing dynamic environments compared with static environments. While static environments do simplify the process of analyzing a proposed SLAM algorithm or improvements to an exploration strategy, these oversimplifications do not allow the research to be translated to real-world use. A greater focus on research in the vein of Vannoy and Xiao's work[80] on Real-Time Adaptive Motion planning is needed, but with greater emphasis on mapping and localization actions. Furthermore, a critical consideration in addressing dynamic environments is recognizing humans within either previously unmapped (so they are not included in the map being generated) or mapped (to detect intruders) environments. The concerns related to human interaction extend beyond the consideration of humans as obstacles or targets; there is also a significant need to improve research in how operators can best control MRS. While MRS are being designed and scaled with members numbering from two to 2000, minimal work has been done on the approaches for humans to interact with these systems as they scale with varying levels of autonomy and capability. Even with the most advanced and capable system, if it is not designed with human operation as a critical constraint, it will be significantly less effective.

References

1. A. A. Makarenko, S. B. Williams, F. Bourgault and H. F. Durrant-Whyte, "An Experiment in Integrated Exploration," **In**: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland (Sep.–Oct. 2002) pp. 534–539.
2. A. Howard, M. J. Matarić and G. S. Sukhatme, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation," **In**: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland (Sep.–Oct. 2002) pp. 434–439.
3. F. Amigoni, V. Caglioti and G. Fontana, "A Perceptive Multirobot System for Monitoring Electro-Magnetic Fields," *Proceedings of IEEE Symposium on Virtual Environments, Hunan-Computer Interfaces and Measurement Systems*, Boston, Massachusetts (Jul. 2004) pp. 95–100.
4. D. Latimer IV, S. Srinivasa and V. Lee-Shue, "Towards Sensor-Based Coverage with Robot Teams," **In**: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC (May 2002) pp. 961–967.
5. A. V. Savkin and H. Teimoori, "Decentralized navigation of groups of wheeled mobile robots with limited communication," *IEEE Trans. Robot.* **26**, 1099–1104 (2010).
6. Y. Mohan and S. G. Ponnambalam, "An Extensive Review of Research in Swarm Robotics," *Proceedings of World Congress on Nature and Biologically Inspired Computing*, Coimbatore, India (Dec. 2009) pp. 140–145.
7. R. Simmons, S. Singh, D. Hershberger, J. Ramos and T. Smith, "First Results in the Coordination of Heterogeneous Robots for Large-Scale Assembly," **In**: *Proceedings of the International Symposium on Experimental Robotics*, Honolulu, Hawaii (Dec. 2000) pp. 323–322.
8. P. Ben-Tzvi, A. A. Goldenberg and J. W. Zu, "Design and analysis of a hybrid mobile robot mechanism with compounded locomotion and manipulation capability," *J. Mech. Des.* **130**, 1–13 (2008).
9. Y. Kim and M. A. Minor, "Distributed kinematic motion control of multi-robot coordination subject to physical constraints," *Int. J. Robot. Res.* **29**, 92–109 (2010).
10. Y. Kim and M. A. Minor, "Coordinated kinematic control of compliantly coupled multi-robot systems in an array format," *IEEE Trans. Robot.* **26**, 173–180 (2010).
11. R. O'Grady, A. L. Christensen and M. Dorigo, "SWARMORPH: multi-robot morphogenesis using directional self-assembly," *IEEE Trans. Robot.* **25**, 738–743 (2009).
12. S. J. Moorehead, R. Simmons and W. L. Whittaker, "Autonomous exploration using multiple sources of information," **In**: *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea (May 2001) pp. 3098–3103.
13. A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer* **22**, 46–57 (1989).
14. F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky and H. F. Durrant-Whyte, "Information-Based Adaptive Robotic Exploration," **In**: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland (Sep.–Oct. 2002) pp. 540–545.
15. A. A. Makarenko, S. B. Williams, F. Bourgault and H. F. Durrant-Whyte, "Decentralized Certainty Grid Maps," **In**: *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada (Oct. 2003) pp. 3258–3263.
16. E. Brunskill and N. Roy, "SLAM using Incremental Probabilistic PCA and Dimensionality Reduction," **In**: *Proceedings of the IEEE/RSJ International Conference of Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 342–347.
17. S. J. Lee, D. W. Cho, W. K. Chung, J. H. Lim and C. U. Kang, "Feature-Based Map Building Using Sparse Sonar Data," **In**: *Proceedings of the 2005 International Conference on Intelligent Robots and Systems*, Edmonton, Canada (Aug. 2005) pp. 1648–1652.
18. H. González-Baños, E. Mao, J. C. Latombe, T. M. Murali and A. Efrat, "Planning Robot Motion Strategies for Efficient Model Construction," **In**: *Proceedings of the 9th International Symposium of Robotics Research*, Snowbird, Utah (Oct. 2000) pp. 345–352.
19. S. Albers and M. R. Henzinger, "Exploring Unknown Environments," **In**: *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, El Paso, Texas (May 1997) pp. 416–425.
20. R. Rocha, F. Ferreira and J. Dias, "Multi-Robot Complete Exploration Using Hill Climbing and Topological Recovery," **In**: *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France (Sep. 2008) pp. 1884–1889.
21. Y. Tarutoko, K. Kobayashi and K. Watanabe, "Topological Map Generation Based on Delaunay Triangulation for Mobile Robot," *Proceedings of the International Joint Conference of SICE-ICASE*, Busan, Korea (Oct. 2006) pp. 492–496.
22. G. Hollinger, S. Singh, J. Djugash and A. Kehagias, "Efficient multi-robot search for a moving target," *Int. J. Robot. Res.* **28**, 201–219 (2009).
23. R. Rocha, J. Dias and A. Carvaho, "Exploring Information Theory for Vision-Based Volumetric Mapping," **In**: *Proceedings of the 2005 International Conference on Intelligent Robots and Systems*, Edmonton, Canada (Aug. 2005) pp. 1023–1028.
24. M. Karg, K. M. Wurm, C. Stachniss, K. Dietmayer and W. Burgard, "Consistent Mapping of Multistory Buildings by Introducing Global Constraints to Graph-Based SLAM," **In**: *Proceedings of IEEE International Conference on Robotics and Automation*, Anchorage, Alaska (May 2010) pp. 5383–5388.

25. E. Montijano and C. Sagues, "Topological Maps Based on Graphs of Planar Regions," **In**: *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, Missouri (Oct. 2009) pp. 1661–1666.
26. A. Bandera, C. Urdiales and F. Sandoval, "An Hierarchical Approach to Grid-Based and Topological Maps Integration for Autonomous Indoor Navigation," **In**: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii (Oct.–Nov. 2001) pp. 883–888.
27. T. Kaupp, B. Douillard, B. Upcroft and A. Makarenko, "Hierarchical Environment Model for Fusing Information from Human Operators and Robots," **In**: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (Oct. 2006) pp. 5837–5842.
28. F. Ferreira, I. Amorim, R. Rocha and J. Dias, "T-SLAM: Registering Topological and Geometric Maps for Robot Localization in Large Environments," **In**: *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea (2008) pp. 392–398.
29. A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proc. IEEE* **94**, 1384–1397 (2006).
30. K. Konolige, D. Fox, B. Limketkai, J. Ko and B. Stewart, "Map Merging for Distributed Robot Navigation," **In**: *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada (Oct. 2003) pp. 212–217.
31. J. Ko, B. Steward, D. Fox, K. Konolige and B. Limketkai, "A Practical, Decision-Theoretic Approach to Multi-Robot Mapping and Exploration," **In**: *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada (Oct. 2003) pp. 3232–3238.
32. X. S. Zhou and S. I. Roumeliotis, "Multi-Robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case," **In**: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (Oct. 2006) pp. 1785–1792.
33. A. Franchi, L. Freda, G. Oriolo and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatron.* **14**, 163–175 (2009).
34. G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff and W. Burgard, "Efficient Estimation of Accurate Maximum Likelihood Maps in 3D," **In**: *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, California (Oct.–Nov. 2007) pp. 3472–3478.
35. P. Drews Jr., P. Núñez, R. Rocha, M. Campos and J. Dias, "Novelty Detection and 3D Shape Retrieval Using Superquadrics and Multi-Scale Sampling for Autonomous Mobile Robots," **In**: *Proceedings of IEEE International Conference on Robotics and Automation*, Anchorage, Alaska (May 2010) pp. 3635–3640.
36. P. Núñez, P. Drews Jr., R. Rocha, M. Campos and J. Dias, "Novelty Detection and 3D Shape Retrieval Based on Gaussian Mixture Models for Autonomous Surveillance Robotics," **In**: *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, Missouri (Oct. 2009) pp. 4724–4730.
37. E. Ivanjko, M. Vašak and I. Petrović, "Kalman Filter Theory-Based Mobile Robot Pose Tracking Using Occupancy Grid Maps," *International Conference on Control and Automation*, Budapest, Hungary (Jun. 2005) pp. 869–874.
38. D. Fox, W. Burgard, H. Kruppa and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autom. Robot.* **8**, 325–344 (2000).
39. A. Franchi, G. Oriolo and P. Stegagno, "On the Solvability of the Mutual Localization Problem with Autonomous Position Measures," **In**: *Proceedings of IEEE International Conference on Robotics and Automation*, Anchorage, Alaska (May 2010) pp. 3193–3199.
40. Y. Dieudonné, O. Labbani-Igbida and F. Petit, "Deterministic robot-network localization is hard," *IEEE Trans. Robot.* **26**, 331–339 (2010).
41. N. Ayanian and V. Kumar, "Abstractions and Controllers for Groups of Robots in Environments with Obstacles," **In**: *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska (May 2010) pp. 3537–3542.
42. L. Banjanović-Mehmedović, I. Petrović and E. Ivanjko, "Mobile Robot Localization Using Local Occupancy Grid Maps Transformations," **In**: *Proceedings of 12th International Power Electronics and Motion Control Conference*, Portoroz, Slovenia (Aug.–Sep. 2006) pp. 1307–1312.
43. L. Armesto and J. Tornero, "Robust and Efficient Mobile Robot Self-Localization Using Laser Scanner and Geometrical Maps" **In**: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (Oct. 2006) pp. 3080–3085.
44. B. Gerkey and M. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. Robot. Res.* 23, 939–954 (2004).
45. M. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak and A. Kleywegt, "Simple Auctions with Performance Guarantees for Multi-Robot Task Allocation," **In**: *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan (Sep.–Oct. 2004) pp. 698–705.
46. A. Singh, A. Krause, C. Gustrin and W. Kaiser, "Efficient Informative Sensing Using Multiple Robots," *J. Artif. Intell. Res.* 34, 707–755 (2009).
47. S. I. Roumeliotis and G. A. Bekey, "Distributed Multirobot Localization," *IEEE Trans. Robot. Autom.* **18**, 781–795 (2002).
48. A. I. Mourikis and S. I. Roumeliotis, "On the treatment of relative-pose measurements for mobile robot localization," **In**: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida (May 2006) pp. 2277–2284.
49. G. A. Borges and M. J. Aldon, "Optimal mobile robot pose estimation using geometrical maps," *IEEE Trans. Robot. Autom.* **18**, 87–94 (2002).
50. H. Chen, D. Sun, J. Yang and J. Chen, "Localization for multirobot formations in indoor environment," *IEEE/ASME Trans. Mechatron.* **15**, 561–574 (2010).
51. H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.* **17**, 125–137 (2001).
52. V. Odakura and A. H. R. Costa, "Cooperative Multi-Robot Localization: Using Communication to Reduce Localization Error," *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, Barcelona, Spain (Sep. 2005).
53. A. Doucet, N. J. Gordon and Y. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Trans. Signal Process.* **49**, 613–624 (2001).
54. A. Brooks, A. Makarenko and B. Upcroft, "Gaussian Process Models for Sensor-Centric Robot Localization," **In**: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida (May 2006) pp. 56–61.
55. V. Cevher, "Importance Sampling," *STAT 631/ELEC 639: Graphical Models*, Rice University, Houston, Texas (Nov. 17, 2008). Available at: www.ece.rice.edu/∼vc3/elec633/ImportanceSampling.pdf
56. J. M. Porta, J. J. Verbeek and B. J. A. Kröse, "Active appearance-based robot localization using stereo vision," *Autom. Robot.* **18**, 59–80 (2005).
57. A. Howard, M. J. Matarić and G. S. Sukhatme, "Putting the 'I' in 'Team': An Ego-Centric Approach to Cooperative Localization," **In**: *Proceedings of the 2003*

IEEE International Conference on Robotics and Automation, Taipei, Taiwan (Sep. 2003) pp. 868–874.
58. E. B. Martinson and F. Dellaert, "Marco Polo localization," **In**: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sep. 2003) pp. 1960–1965.
59. F. Dellaert, F. Alegre and E. B. Martinson, "Intrinsic Localization and Mapping with 2 Applications: Diffusion Mapping and Marco Polo Localization," **In**: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sep. 2003) pp. 2344–2349.
60. J. Djugash and S. Singh, "A Robust Method of Localization and Mapping Using Only Range," *Proceedings of the 11th International Symposium on Experimental Robotics*, Athens, Greece (Jul. 2008).
61. G. Hollinger, J. Djugash and S. Singh, "Coordinated Search in Cluttered Environment using Range from Mobile Robots," **In**: *Proceedings of the 6th Inernational Conference on Field and Service Robotics*, Chamonix, France (Jul. 2007) pp. 443–442.
62. B. Ferris, D. Fox and N. Lawrence, "WiFi-SLAM Using Gaussian Process Latent Variable Models," **In**: *Proceedings of the 2007 International Joint Conference on Artificial Intelligence*, Hyderabad, India (2007) pp. 2480–2485.
63. N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *Int. J. Robot. Res.* **28**, 962–975 (2009).
64. E. J. Rodríguez-Seda, J. J. Troy, C. A. Erignac, P. Murray, D. M. Stipanović and M. W. Spong, "Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance," *IEEE Trans. Control Syst. Technol.* **18**, 984–992 (2010).
65. G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. V. Valk, N. Michael, G. Pappas and K. Daniilidis, "Vision-based localization for leader-follower formation control," *IEEE Trans. Robot.* **25**, 1431–1438 (2009).
66. M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robot. Autom.* **17**, 947–951 (2001).
67. J. R. T. Lawton, R. W. Beard and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.* **19**, 933–941 (2003).
68. D. Sun, C. Wang, W. Shang and G. Feng, "A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations," *IEEE Trans. Robot.* **25**, 1074–1086 (2009).
69. A. Ganguli, J. Cortés and F. Bullo, "Multirobot rendezvous with visibility sensors in nonconvex environments," *IEEE Trans. Robot.* **25**, 340–352 (2009).
70. Y. Litus, P. Zebroski and R. T. Vaughan, "A distributed heuristic for energy-efficient multirobot multiplace rendezvous," *IEEE Trans. Robot.* **25**, 130–135 (2009).
71. R. C. Arkin and J. Diaz, "Line-of-Sight Constrained Exploration for Reactive Multiagent Robotic Teams," **In**: *Proceedings of the 7th International Workshop on Advanced Motion Control*, Maribor, Slovenia (Jul. 2002) pp. 455–461.
72. F. Amigoni, N. Basilico and N. Gatti, "Finding the Optimal Strategies for Robotic Patrolling with Adversaries in Topologically Represented Environments," **In**: *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan (May 2009) pp. 3850–3855.
73. N. Basilico, N. Gatti, T. Rossi, S. Ceppi and F. Amigoni, "Extending Algorithms for Mobile Robot Patrolling in the Presence of Adversaries to more Realistic Settings," **In**: *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, Milan, Italy (Sep. 2009) pp. 557–564.
74. H. Choset, "Coverage for robotics – a survey of recent results," *Ann. Math. Artif. Intell.* **31**, 113–126 (2001).
75. C. Luo and S. X. Yang, "A Real-Time Cooperative Sweeping Strategy or Multiple Cleaning Robots," **In**: *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Vancouver, Canada (Oct. 2002) pp. 660–665.
76. X. Zheng, S. Koenig, D. Kempe and S. Jain, "Multirobot Forest Coverage for Weighted and Unweighted Terrain," *IEEE Trans. Robot.* **26**, 1018–1031 (2010).
77. M. Juliá, A. Gil, A. Payá and Ó. Reinoso, "Local minima detection in potential field-based cooperative multi-robot exploration," *Int. J. Factor. Autom. Robot. Soft Comput.* **3** (2008).
78. S. Garrido, L. Moreno and D. Blanco, "Exploration and mapping using the VFM motion planner," *IEEE Trans. Instrum. Meas.* **58**, 2880–2892 (2009).
79. D. M. Stipanović, P. F. Hokayem, M. W. Spong and D. D. Šiljak, "Cooperative avoidance control for multiagent systems," *J. Dyn. Syst. Meas. Control* **129**, 699–707 (2007).
80. J. Vannoy and J. Xiao, "Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes," *IEEE Tran. Robot.* **24**, 1199–1212 (2008).
81. S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte and A. Y. Ng, "Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results," *Proceedings of the 5th International Workshop on Algorithmic Foundations of Robotics*, Nice, France (Dec. 2002).
82. R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and H. Younes, "Coordination for Multi-Robot Exploration and Mapping," **In**: *Proceedings of the AAAI Conference on Artificial Intelligence*, Austin, Texas (Jul.–Aug. 2000) pp. 852–858.
83. K. Lee and W. K. Chung, "Effective maximum likelihood grid map with conflict evaluation filter using sonar sensors," *IEEE Trans. Robot.* **25**, 887–901 (2009).
84. F. Amigoni and A. Gallo, "A Multi-Objective Exploration Strategy for Mobile Robots," **In**: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 3850–3855.
85. S. Koenig, C. Tovey and W. Halliburton, "Greedy Mapping of Terrain," **In**: *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea (May 2001) pp. 3594–3599.
86. R. Sim and N. Roy, "Global A-Optimal Robot Exploration in SLAM," **In**: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 662–666.
87. W. Burgard, M. Moors, C. Stachniss and F. E. Schneider, "Coordinated Multi-Robot Exploration," *IEEE Trans. Robot.* **21**, 376–386 (2005).
88. H. Lau, "Behavioural Approach for Multi-Robot Exploration," *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane, Australia (Dec. 2003).
89. C. Connolly, "The Determination of Next Best Views," **In**: *Proceedings of the 1985 IEEE International Conference on Robotics and Automation* (Mar. 1985) pp. 432–435.
90. H. H. González-Baños and J. C. Latombe, "Navigation Strategies for Exploring Indoor Environments," *Int. J. Robot. Res.* **21**, 829–848 (2002).
91. G. Oriolo, M. Vendittelli, L. Freda and G. Troso, "The SRT Method: Randomized Strategies for Exploration," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana (Apr.–May 2004) pp. 4688–4694.
92. L. Freda and G. Oriolo, "Frontier-Based Probabilistic Strategies for Sensor-Based Exploration," **In**: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 3881–3887.
93. K. M. Wurm, C. Stachniss and W. Burgard, "Coordinated Multi-Robot Exploration using a Segmentation of the Environment," **In**: *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France (Sep. 2008) pp. 1160–1165.

94. S. Berman, Á. Halász, M. Ani-Hsieh and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Trans. Robot.* **25**, 927–937 (2009).
95. M. Dias, R. Zlot, N. Kalra and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proc. IEEE* **94**, 1257–1270 (Jul. 2006).
96. W. Sheng, Q. Yang, J. Tan and N. Xi, "Distributed multi-robot coordination in area exploration," *Robot. Autom. Syst.* **54**, 945–955 (2006).
97. M. B. Dias and A. Stentz, "A Free Market Architecture for Distributed Control of a Multirobot System," *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*, Venice, Italy (July 2000).
98. H. L. Choi, L. Brunet and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.* **25**, 912–926 (2009).
99. N. Meskin, K. Khorasani and C. A. Rabbath, "A hybrid fault detection and isolation strategy for a network of a unmanned vehicles in presence of large environmental disturbances," *IEEE Trans. Control Syst. Technol.* **18**, 1422–1429 (2010).
100. J. W. Durham, A. Franchi and F. Bullo, "Distributed Pursuit-Evasion with Limited-Visibility Sensors via Frontier-Based Exploration," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska (May 2010) pp. 3562–3568.
101. F. Ghaderi and M. N. Ahamadabadi, "A Cooperative Fault Tolerance Strategy for Distributed Object Lifting Robots," **In**: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland (Sep.–Oct. 2002) pp. 2721–2727.
102. K. Y. K. Leung, T. D. Barfoot and H. H. T. Liu, "Decentralized localization of sparsely communicating robot networks: A centralized-equivalent approach," *IEEE Trans. Robot.* **26**, 62–77 (2010).
103. N. Kalra, D. Ferguson and A. Stentz, "Hoplites: A Market-Based Framework for Planned Tight Coordination in Multirobot Teams," **In**: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 1170–1177.
104. R. Rocha, J. Dias and A. Carvaho, "Cooperative Multi-Robot Systems: A Study Of Vision-Based 3-D Mapping Using Information Theory," **In**: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005) pp. 384–389.
105. D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel and L. Montano, "Enforcing network connectivity in robot team mission," *Int. J. Robot. Res.* **29**, 460–480 (2010).
106. R. Zlot, A. Stentz, M. B. Dias and S. Thayer, "Multi-Robot Exploration Controlled by a Market Economy," **In**: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC (May 2002) pp. 3016–3023.
107. I. Rekleitis, R. Sim, G. Dudek and E. Milios, "Collaborative Exploration for the Construction of Visual Maps," **In**: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii (Oct.–Nov. 2001) pp. 1269–1274.
108. A. Franchi, L. Freda, G. Oriolo and M. Vendittelli, "A Randomized Strategy for Cooperative Robot Exploration," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy (Apr. 2007) pp. 768–774.
109. W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun, "Collaborative Multi-Robot Exploration," **In**: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, California (Apr. 2000) pp. 476–481.
110. M. Kloetzer and C. Belta, "Automatic deployment of distributed teams of robots from temporal logic motion specifications," *IEEE Trans. Robot.* **26**, 48–61 (2010).
111. B. Frank, C. Stachniss, R. Schmedding, M. Teschner and W. Burgard, "Real-World Robot Navigation Amongst Deformable Obstacles," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan (May 2009) pp. 1649–1654.
112. H. Yamaguchi, "A Distributed Motion Coordination Strategy for Multiple Nonholonomic Mobile Robots in Cooperative Hunting Operations," **In**: *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada (Dec. 2002) pp. 2984–2991.
113. N. Moshtagh, N. Michael, A. Jadbabaie and K. Daniilidis, "Vision-based, distributed control laws for motion coordination of non-holonomic robots," *IEEE Trans. Robot.* **25**, 851–860 (2009).
114. A. Makarenko, T. Kaupp, B. Grocholsky and H. Durrant-Whyte, "Human-Robot Interactions in Active Sensor Networks," **In**: *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Kobe, Japan (July 2003) pp. 247–252.
115. R. R. Murphy, "Human-Robot Interaction in Rescue Robotics," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **34**, 138–153 (2004).
116. D. F. Hougen, S. Benjaafar, J. C. Bonney, J. R. Budenske, M. Dvorak, M. Gini, H. French, D. G. Krantz, P. Y. Li, F. Malver, B. Nelson, N. Papanikolopoulos, P. E. Rybski, S. A. Stoeter, R. Voyles and K. B. Yesin, "A Miniature Robotic System for Reconnaissance and Surveillance," **In**: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, California (Apr. 2000) pp. 501–507.
117. J. F. Diaz, A. Stoytchev and R. C. Arkin, "Exploring Unknown Structured Environments," *Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference*, Key West, Florida (May 2001).
118. D. S. Apostolopoulos, L. Pedersen, B. N. Shamah, K. Shillcutt, M. D. Wagner and W. L. Whittaker, "Robotic Antarctic Meteorite Search: Outcomes," **In**: *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea (May 2001) pp. 4174–4179.
119. F. Amigoni, S. Scadonici, V. Vaglioti and G. Fontana, "Experimenting with a Robotic System for Localizing Magnetic Field Sources," **In**: *Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems*, Giardini Naxos, Italy (July 2005) pp. 44–49.
120. M. Schwager, D. Rus and J. J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Robot. Res.* **28**, 357–375 (2009).
121. T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.* **18**, 94–103 (2002).
122. K. Suzuki, S. Makami, J. Akita and E. Osawa, "Development of Cooperative Small Mowing Robots," **In**: *Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Kobe, Japan (July 2003) pp. 1498–1502.
123. S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer and W. Whittaker, "A System for Volumetric Robotic Mapping of Abandoned Mines," **In**: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sept. 2003) pp. 4270–4275.
124. S. Lee and J. Song, "Robust Mobile Robot Localization Using Optical Flow Sensors and Encoders," **In**: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana (Apr. 2004) pp. 1039–1044.
125. D. Kroetsch and C. Clark, "Towards Gaussian Multi-Robot SLAM for Underwater Robotics," *Proceedings of the 2005 International Symposium on Unmanned Untethered Submersible Technology* (Jul. 2005).

126. A. El-Osery, "EE 570: Location and Navigation. INS/GPS Integration," Technical Report, Electrical Engineering Department, New Mexico Tech, Socorro, New Mexico, USA (April 29, 2011), available at: www.ee.nmt.edu/∼elosery/spring_2011/ee570/lectures/ins-gps.pres.pdf, Accessed 26 October 2011.
127. F. Rivard, J. Bisson, F. Michaud and D. Létourneau, "Ultrasonic Relative Positioning for Multi-Robot Systems," **In**: *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, Pasadena, California (May 2008) pp. 323–328.
128. N. Cen, K. Cheng and B. Fidan, "Formation Control of Robotic Swarms Based on Sonar Sensing," **In**: *Proceedings of the 2009 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, Australia (Dec. 2009) pp. 31–36.
129. S. Nejad and S. Olyaee, "Low-Noise High-Accuracy TOF Laser Range Finder," *Am. J. Appl. Sci.* **5**, 755–762 (2008).
130. M. Agrawal and K. Konolige, "Real-Time Localization in Outdoor Environments Using Stereo Vision and Inexpensive GPS," *Proceedings of the 18th Inernational Conference on Pattern Recognition*, Hong Kong (Aug. 2006).