Cooperative SLAM using M-Space representation of linear features

Daniele Benedettelli, Andrea Garulli, Antonio Giannitrapani*

Dipartimento di Ingegneria dell'Informazione, Università di Siena, 53100 Siena, Italy

Abstract

This paper presents a multi-robot simultaneous localization and map building (SLAM) algorithm, suitable for environments which can be represented in terms of lines and segments. Linear features are described by adopting the recently introduced M-Space representation, which provides a unified framework for the parameterization of different kinds of features. The proposed solution to the cooperative SLAM problem is split into three phases. Initially, each robot solves the SLAM problem independently. When two robots meet, their local maps are merged together using robot-to-robot relative range and bearing measurements. Then, each robot starts over with the single-robot SLAM algorithm, by exploiting the merged map.

The proposed map fusion technique is specifically tailored to the adopted feature representation, and takes into account explicitly the uncertainty affecting both the maps and the robot mutual measurements. Numerical simulations and experiments with a team composed of two robots performing SLAM in a real-world scenario, are presented to evaluate the effectiveness of the proposed approach.

Keywords: SLAM, mapping, multi-robot, M-Space

[*] Corresponding author

Email addresses: `danielebenedettelli@gmail.com` (Daniele Benedettelli), `garulli@ing.unisi.it` (Andrea Garulli), `giannitrapani@ing.unisi.it` (Antonio Giannitrapani)

1. Introduction

The self-localization of a mobile robot is widely recognized as one of the most basic problem of autonomous navigation. While such a task can be performed pretty well when the environment is a priori known, robot localization becomes much harder when a map of the environment is not available beforehand. This may be due to a lack of information on the environment the robot moves in, or to the excessive cost of manually building a map on purpose. In these circumstances, the robot must at the same time build a map of the environment and localize itself within it. This problem, known as *Simultaneous Localization and Map building* (SLAM), has been extensively studied over the last two decades (see [1–6] and references therein for a thorough review). The solutions to the SLAM problem presented so far differ mainly for the environment description adopted and for the estimation technique employed. In many applications of interest, a mobile robot has to move in indoor environments, which can be well described in terms of linear features. Typical examples are houses or offices, where there is plenty of walls and furniture [7, 8]. When it comes to represent lines and segments, several possibilities are available, each with its pros and cons. Probably the most intuitive way to represent a segment is to specify the coordinates of its endpoints. One drawback of such parameterization is that more often than not the actual extrema of the segment are not really observed in a single measurement, due to sensor limited field of view or occlusions (e.g., only a portion of a long wall is measured by a laser scan). In these circumstances what can still be measured are the distance ρ and the angle α of the supporting line of the segment. In fact, the ρ and α parameterization is indeed anther possible representation of linear features for mapping purposes. The biggest drawback of such a choice is the so called *lever arm* effect, meaning that the parameter uncertainty increases with the distance from the origin of the reference frame. A possible solution is to resort to special line representations, like the SP-Model [1] or the recently proposed M-Space representation [9], which express the feature coordinates in different local reference frames. The M-Space representation provides a unified framework for describing different kinds of 2D and 3D geometric features, thus being versatile enough to be adopted for SLAM purposes in a wide range of environments and in presence of heterogeneous sensors. One key feature of the M-Space representation is its ability to use only the partial information contained in a measurement of a given element of the map. This characteristic is especially useful for the early initialization of the estimate of a feature which has been only partially observed.

When exploring large areas, the map building procedure can be more effective if tackled by a team of robots. Besides being able to cover a given area faster than a single agent, a multi-robot system is more robust to failure and often also less expensive when compared to a single complex vehicle. Another crucial issue for a SLAM algorithm is being able to recognize places that have been already visited (see, e.g., [10–12]). Since the difficulty of the loop closure increases with the size of the environment, using a team of robots allow one to break down a large area into smaller regions to be explored by each agent. In light of these considerations, several multi-robot SLAM algorithms have been proposed in recent years, adopting different estimation techniques, like Extended Kalman Filters (EKF) [13, 14], information filters [15], particle filters [16], set-membership estimators [17], or sparse optimization techniques [18].

Besides the aforementioned advantages, cooperative localization and mapping solutions present an additional challenge too. In fact, a key issue for the effectiveness of multi-robot SLAM algorithms is the ability to merge in a common reference frame maps built by different robots in different frames. Such a task, which is particularly difficult if no information is available on the initial robot poses, is crucial to preserve the quality of the map and fully exploit the benefits of the multi-agent architecture. Wrong or inaccurate map merging can completely destroy the map consistency and eventually jeopardize the correct behavior of the system. Hence, despite the heterogeneity of techniques employed, most multi-robot SLAM algorithms have tackled this point. A fusion algorithm for maps made up of point-wise landmarks has been presented in [19]. Under the assumption that the agents can perform range and bearing mutual measurements during a rendezvous, the frame transformation is derived from geometrical arguments and a two step merging procedure has been proposed. First, the map built by one robot is incorporated into the map of the other one, according to the coordinate transformation relating the reference frames of the two robots. Then, correspondences between landmarks present in both maps are sought for, in order to improve the accuracy of the coordinate transformation. An algorithm for merging two occupancy grid maps has been proposed in [20]. In this case a set of possible reference frame transformations are computed by analyzing the cross-correlation of suitable spectra of the two maps. Each transformation is then assigned a weight representing the confidence on the corresponding merged map. In this way it is also possible to track multiple hypotheses in case of ambiguous associations. While this approach does not require robot rendezvous and mutual measurements, it can be applied only if the two maps have significant overlap. Recently, a probabilistic map merging procedure has been proposed in [21]. Analogously to what is done in [19], map fusion is carried out in two steps. An initial map alignment is performed based on range and bearing robot-to-robot measurements. Then the merged map is updated according to duplicate features present in both original maps. The novelty of this approach lies in the probabilistic method adopted for the merging procedure, which is suitable for particle filter based SLAM algorithms. A map merging algorithm for mixed topological/metric maps has been presented in [22]. A graph-like topological map is built, with vertices representing local occupancy grid maps and edges describing relative positions of adjacent local maps. In this framework, the map fusion between two robots boils down to adding an edge that connects the two topological maps, and associating to it the estimation of the relative robot pose. A similar approach is adopted in [23], where each robot builds landmark-based local maps topologically connected through an adjacency graph. In this framework, rendezvous between robots, feature correspondences in different robot maps and absolute localization measurements give rise to a cycle in the global graph which translates into constraints that allow the system to refine the estimates of the transformation between the local reference frames.

This work has been extended in [24] for dealing with heterogeneous teams of aerial and ground robots equipped with monocular cameras. Several local 3D maps containing visual landmarks and line segments are built, whereas a global connectivity graph capture their relative relationships. Another visual SLAM algorithm for a team of robots has been recently presented in [25]. In this case, a Rao-Blackwellized particle filter is proposed to collaboratively build a global map made of 3D visual landmarks detected by stereo cameras.

In this paper, a new multi-robot SLAM algorithm is presented, for line-based environment descriptions. The proposed approach builds upon the M-Space representation to describe linear features and adopts a map fusion scheme inspired by the technique proposed in [19]. The multi-robot algorithm goes through three stages. Initially, each robot runs an EKF-based single-robot SLAM algorithm for independently building local maps until a rendezvous occurs. When two robots meet, the local maps are merged by using the information coming from mutual robot measurements and feature matching. Afterward, the agents start over with the single-robot SLAM algorithm, taking advantage of the merged map. The main contribution of the paper is to present a novel map fusion algorithm tailored to environments described in terms of lines and segments. The uncertainty of the resulting map is properly updated, and the covariance matrices necessary to resume the single-robot SLAM when the M-Space representation is adopted are analytically computed. Results from numerical simulations and experimental tests involving real robots are reported, to assess the viability of the proposed approach in a real-world scenario. A preliminary version of this work has been presented in [26].

The paper is organized as follows. In Section 2, the M-Space representation of linear features is revisited. The single-robot SLAM algorithm, based on the EKF and M-Space representation is outlined in Section 3. The main contribution of the paper is presented in Section 4, where the map fusion technique, as well as the update of the resulting map uncertainty, are illustrated. In Section 5 the results of simulations and experimental tests with real robots are reported. Finally, in Section 6 some conclusions are drawn and lines of future research are outlined.

Notation. The symbol I_q denotes the identity matrix of order q. The matrix $\mathrm{diag}(a_1, \ldots, a_n)$ is the diagonal matrix having the scalars $a_1, \ldots, a_n$ on its diagonal. Similarly, the matrix $\mathrm{blkdiag}(A_1, \ldots, A_n)$ is the block diagonal matrix having matrices $A_1, \ldots, A_n$ on its diagonal. Boldface symbols denote vectors. The symbol $\hat{\mathbf{x}}$ denotes the estimate of the quantity $\mathbf{x}$, and $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is the corresponding estimation error. In the notation ${}^f\mathbf{x}_r$, the left superscript f means that the quantity $\mathbf{x}$ is expressed in the reference frame $\langle R_f \rangle$, whereas the right subscript r indicates which robot the quantity $\mathbf{x}$ refers to. Whenever found, the subscript s denotes a quantity expressed in the M-Space.

2. M-Space representation of segments

Among the several possible representations, in this paper a line segment in the plane is described in a reference frame $\langle G \rangle$ either by its endpoints coordinates

$$\mathbf{x}_f = [x_A \; y_A \; x_B \; y_B]^T, \tag{1}$$

or, alternatively, by the parameters

$$\mathbf{x}_p = [\alpha, \; \rho, \; d_A, \; d_B]^T. \tag{2}$$

The parameter $\alpha \in (-\pi, \pi]$ is the angle between the x-axis of $\langle G \rangle$ and the normal $\mathbf{n}$ to the segment passing through the origin, whereas $\rho \geq 0$ is the distance of the origin of $\langle G \rangle$ to the line. To define d_A and d_B, let us introduce a second reference frame $\langle S \rangle$, whose origin is at the intersection between the segment and the normal $\mathbf{n}$, and whose x-axis lies on $\mathbf{n}$ pointing out from the origin of $\langle G \rangle$. Then, d_A and d_B are the coordinates (with proper sign) of the segment endpoints A and B along the y-axis of $\langle S \rangle$ (see Figure 1).

As pointed out in the previous section, both parameterizations suffer from some drawbacks: the segment endpoints are not always observable from a measurement, whereas the $\rho - \alpha$ representation is prone to the lever-arm effect. A possible solution, overcoming both problems, is provided by the *measurement*

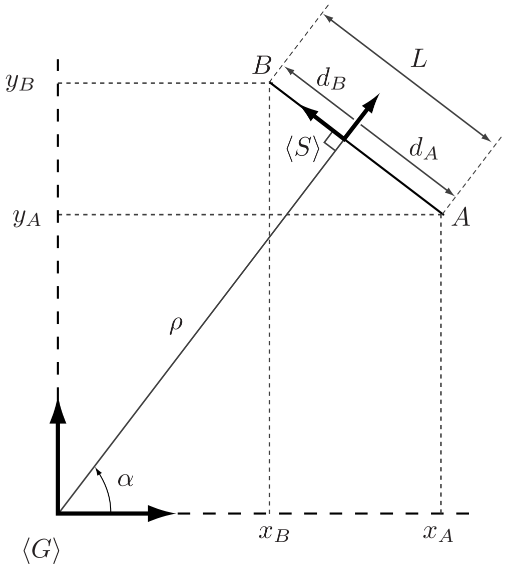


Figure 1: Two different segment parameterizations. In this case $d_A < 0$ and $d_B > 0$.

subspace representation (briefly, the M-Space representation). In this framework, a local reference frame is attached to each feature, with respect to which the feature parameters (or coordinates) are expressed. The M-Space is general enough to allow for a uniform treatment of many different kinds of features, like 2D or 3D point-wise landmarks, segments, lines, planes. In the following, we will focus on maps made up of 2D segments, referring the reader interested to the M-Space representation of generic features to [9].

The M-Space coordinates of a segment are the parameters $\mathbf{x}_p$ in (2), expressed in the reference frame of the corresponding feature. As a consequence, coordinates $\mathbf{x}_p$ relative to different features live in different reference frames. An immediate benefit of the M-Space representation is that being each feature described in a local reference frame, the effects of the lever-arm phenomenon are mitigated. Intuitively, this happens because each segment is close to the origin of the corresponding local reference frame. Another nice property of the M-Space representation is that the features are parametrized in a way that both their location and extent are fully specified, as it occurs when the segment endpoints are used. However, until a segment is not completely observed, the M-Space representation allows one to initialize the feature in a subspace corresponding to the partial information provided by the robot sensors. As a consequence, the dimension q of the M-Space coordinates $\mathbf{x}_p$ can grow from a minimum of 2 (when the robot first measures the line position and orientation) up to 4 (when it has observed both segment endpoints). Similarly, whenever a segment already included in the map is not entirely observed, i.e. only ρ and α are measured, still this information can be exploited to correct the current estimate of the feature.

The M-Space coordinates and feature space coordinates of a feature are related by two projection matrices $B_f(\mathbf{x}_f)$ and $\widetilde{B}_f(\mathbf{x}_f)$. Let $\delta\mathbf{x}_f$ denote a small change in the feature coordinates corresponding to a small change in the M-Space coordinates $\delta\mathbf{x}_p$. The relationships between $\delta\mathbf{x}_p$ and $\delta\mathbf{x}_f$ are

$$\begin{array}{rcl} \delta\mathbf{x}_p & = & B(\mathbf{x}_f)\delta\mathbf{x}_f, \\ \delta\mathbf{x}_f & = & \widetilde{B}(\mathbf{x}_f)\delta\mathbf{x}_p. \end{array} \tag{3}$$

Equations (3), relating small changes in the M-Space to small changes in the feature space, can be applied to project estimate corrections from one space to the other, as it will be shown in the next section. Notice that the projection matrices $B(\mathbf{x}_f)$ and $\widetilde{B}(\mathbf{x}_f)$ are a function of the value of the feature space coordinates $\mathbf{x}_f$, and their dimension varies with the dimensions q of $\mathbf{x}_p$. In the following treatment, for ease of notation we will always assume $q = 4$ (the case $q < 4$ requiring straightforward modifications). In this case, $B(\mathbf{x}_f)$ and $\widetilde{B}(\mathbf{x}_f)$ take on the form

$$B(\mathbf{x}_f) = \begin{pmatrix} \frac{\cos\alpha}{L\sqrt{2}} & \frac{\sin\alpha}{L\sqrt{2}} & -\frac{\cos\alpha}{L\sqrt{2}} & -\frac{\sin\alpha}{L\sqrt{2}} \\ \frac{\cos\alpha}{\sqrt{2}} & \frac{\sin\alpha}{\sqrt{2}} & \frac{\cos\alpha}{\sqrt{2}} & \frac{\sin\alpha}{\sqrt{2}} \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & -\sin\alpha & \cos\alpha \end{pmatrix},$$

$$\widetilde{B}(\mathbf{x}_f) = \begin{pmatrix} \frac{L\cos\alpha}{\sqrt{2}} & \frac{\cos\alpha}{\sqrt{2}} & -\sin\alpha & 0 \\ \frac{L\sin\alpha}{\sqrt{2}} & \frac{\sin\alpha}{\sqrt{2}} & \cos\alpha & 0 \\ -\frac{L\cos\alpha}{\sqrt{2}} & \frac{\cos\alpha}{\sqrt{2}} & 0 & -\sin\alpha \\ -\frac{L\sin\alpha}{\sqrt{2}} & \frac{\sin\alpha}{\sqrt{2}} & 0 & \cos\alpha \end{pmatrix}.$$

where L denotes the length of the segment.

3. Single-robot SLAM

In this section, first the single-robot SLAM problem will be cast as a state estimation problem for an uncertain dynamic system. Then, an EKF-based solution, suitable when an M-Space representation of linear features is adopted, will be briefly reviewed.

Let us consider an autonomous robot navigating in a 2D environment, and let $\mathbf{x}_R = [x_R \; y_R \; \theta_R]^T$ be its pose, where $[x_R \; y_R]^T$ is the position and θ_R is the orientation with respect to a global reference frame. The generic robot motion model based on linear and angular velocity commands $\mathbf{u}(k) = [v(k) \quad \omega(k)]^T$ is denoted by

$$\mathbf{x}_R(k+1) = f(\mathbf{x}_R(k), \mathbf{u}(k), \varepsilon_{\mathbf{u}}(k)), \tag{4}$$

where $\varepsilon_{\mathbf{u}}$ is a white noise affecting the velocities, with $E[\varepsilon_{\mathbf{u}}(k)] = \mathbf{0}$ and $E[\varepsilon_{\mathbf{u}}(k)\varepsilon_{\mathbf{u}}^T(k)] = Q(k)$. Assume that the surrounding environment can be described in terms of linear features, as usually happens in indoor environments presenting walls, doors, or furniture. Then, a map of the environment can be given in terms of n line segments, identified by the coordinates of their endpoints $\mathbf{x}_{f_i}$, $i = 1, \ldots, n$, defined in (1), expressed in the global reference frame. Since static features are considered, their location does not change with time, i.e.

$$\mathbf{x}_{f_i}(k+1) = \mathbf{x}_{f_i}(k), \quad i = 1, \ldots, n. \tag{5}$$

By stacking into the vector $\mathbf{x}$ both the robot pose and the segment endpoints

$$\mathbf{x}(k) = [\mathbf{x}_R^T(k) \; \mathbf{x}_{f_1}^T(k) \; \ldots \; \mathbf{x}_{f_n}^T(k)]^T, \tag{6}$$

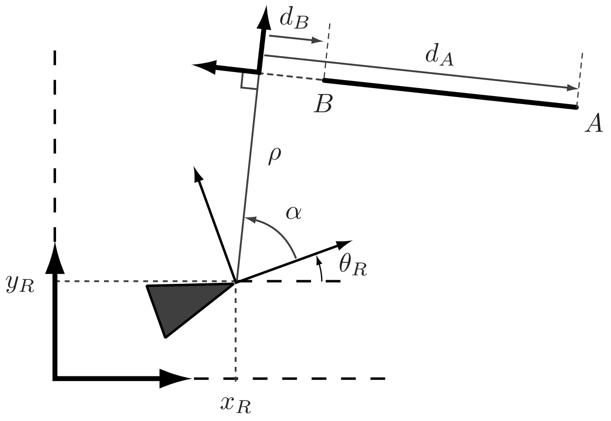


Figure 2: Measurements $\mathbf{m}_i$.

equations (4) and (5) can be rewritten in compact form as

$$\mathbf{x}(k+1) = F(\mathbf{x}(k), \mathbf{u}(k), \varepsilon_{\mathbf{u}}(k)). \tag{7}$$

Notice that features are incrementally detected as the robot explores new regions of the environment, and hence the dimension of the vector $\mathbf{x} \in \mathbb{R}^{3+4n}$ grows with time.

The robot is equipped with sensors able to take measurements of linear features detected during the navigation. Let

$$\mathbf{m}_i(k) = [\alpha_i(k)\ \rho_i(k)\ d_{A_i}(k)\ d_{B_i}(k)]^T + \varepsilon_{\mathbf{m}_i}(k), \quad i = 1, \ldots, n \tag{8}$$

denote the measurement of the i-th line segment at time k, affected by noise $\varepsilon_{\mathbf{m}_i}(k)$ (see Figure 2). It is assumed that the measurement noise can be modeled as a white noise, with $E[\varepsilon_{\mathbf{m}_i}(k)] = \mathbf{0}$ and $E[\varepsilon_{\mathbf{m}_i}(k)\ \varepsilon^T_{\mathbf{m}_i}(k)] = R_{\mathbf{m}_i}(k)$. Measurements like those in (8) can be easily obtained in practice if the vehicle is equipped with a laser rangefinder, returning pairs of range and bearing measurements taken from planar scans of the environment. From the raw laser readings, the measurements $\mathbf{m}$ can be extracted through segmentation and line fitting algorithms. As a by-product of such extraction phase, also the covariance matrix $R_{\mathbf{m}_i}(k)$ of the measurement noise can be estimated, given the accuracy of the laser readings [8]. Notice that $d_{A_i}(k)$ and $d_{B_i}(k)$ are present in the measurement vector $\mathbf{m}_i(k)$ only if the endpoints of the i-th identified segment are detected (e.g., due to a corner). From geometrical considerations, the measurement $\mathbf{m}_i$ can be expressed as a function of the robot pose and the segment endpoints

$$\mathbf{m}_i(k) = h(\mathbf{x}_R(k), \mathbf{x}_{f_i}(k)) + \varepsilon_{\mathbf{m}_i}(k), \quad i = 1, \ldots, n. \tag{9}$$

By grouping all the measurements into a single vector

$$\mathbf{m}(k) = [\mathbf{m}_1^T(k) \ \ldots \ \mathbf{m}_n^T(k)]^T,$$

the measurement equations (9) can be rewritten as

$$\mathbf{m}(k) = H(\mathbf{x}(k)) + \varepsilon_{\mathbf{m}}(k) \tag{10}$$

where $\varepsilon_{\mathbf{m}} = [\varepsilon_{\mathbf{m}_1}^T \ \ldots \ \varepsilon_{\mathbf{m}_n}^T]^T$.

Within this framework, the SLAM problem boils down to estimating the state vector $\mathbf{x}$ given the measurements $\mathbf{m}$. More precisely, let $\hat{\mathbf{x}}(0|0)$ be an estimate of the initial robot pose and feature coordinates. Given the dynamic model (7) and the measurement equation (10), find an estimate $\hat{\mathbf{x}}(k|k)$ of the robot pose and feature coordinates $\mathbf{x}(k)$ based on the measurements $\mathbf{m}$ collected up to time k.

When adopting the M-Space representation, the SLAM problem can be tackled by introducing an auxiliary state vector $\mathbf{x}_s$, which includes the robot pose $\mathbf{x}_R$ in the global frame and the M-Space coordinates $\mathbf{x}_{p_i}$ of each feature, defined in (2)

$$\mathbf{x}_s = \left[\mathbf{x}_R^T \ \mathbf{x}_{p_1}^T \ \ldots \ \mathbf{x}_{p_n}^T\right]^T. \tag{11}$$

A standard EKF is run to compute at each time k an estimate $\hat{\mathbf{x}}_s(k|k)$ of (11), and the covariance matrix

$$P_{\mathbf{x}_s}(k|k) = E\left[\tilde{\mathbf{x}}_s(k|k) \ \tilde{\mathbf{x}}_s^T(k|k)\right]$$

of the corresponding estimation error $\tilde{\mathbf{x}}_s$. Then, the state correction based only on the measurement taken at time k

$$\delta\hat{\mathbf{x}}_s(k) = \hat{\mathbf{x}}_s(k|k) - \hat{\mathbf{x}}_s(k|k-1) \tag{12}$$

is used to update the estimate of the original state vector $\hat{\mathbf{x}}(k|k)$ in the following way. First, the previous estimate $\hat{\mathbf{x}}(k-1|k-1)$ is propagated according to the motion model (7), obtaining the predicted state

$$\hat{\mathbf{x}}(k|k-1) = F(\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k-1), \mathbf{0}).$$

Then, $\hat{\mathbf{x}}(k|k-1)$ is corrected by projecting $\delta\hat{\mathbf{x}}_s(k)$ from the M-Space to the feature space according to the relationship (3). Thus, from (3), (6) and (11) one has

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \Pi\, \delta\mathbf{x}_s(k),$$

where $\Pi = \text{blkdiag}\left(I_3,\ \widetilde{B}(\hat{\mathbf{x}}_{f_1}), \ldots, \widetilde{B}(\hat{\mathbf{x}}_{f_n})\right)$ is built by using the $\widetilde{B}(\mathbf{x}_{f_i})$ matrices in (3) evaluated at the current feature estimates $\hat{\mathbf{x}}_{f_i}(k|k-1)$.

Remark 1. The estimation scheme described so far can be efficiently implemented without explicitly computing the nominal estimates of the auxiliary vector $\mathbf{x}_s$. It can be shown that the correction term $\delta\hat{\mathbf{x}}_s$ in (12) can be expressed as a function of the current estimates in the feature space $\hat{\mathbf{x}}$, the current measurements $\mathbf{m}$ and the covariance matrix $P_{\mathbf{x}_s}$ of the estimation error in the M-Space. A detailed description of this procedure can be found in [9].

Summarizing, the single-robot SLAM algorithm produces an estimate $\hat{\mathbf{x}}(k)$ of the robot pose and the segment endpoints in the global frame, as well as the covariance matrix $P_{\mathbf{x}_s}$, that expresses robot uncertainty in the global frame and feature uncertainty in the M-Space. Notice that if one is interested in expressing the uncertainty of the map in the feature space, the matrix $P_{\mathbf{x}_s}$ can be projected to the feature space by resorting to the relationship (3), as it will be shown in Section 4.2.

3.1. Matching

A key issue for SLAM algorithms to be successful is the data association mechanism. When a robot measures a feature, it must first decide whether the measurements originate from a newly discovered item or they refer to a feature already present in the map. In the latter case, a method for selecting which feature matches the measurements taken from the sensors is needed. Several matching techniques have been proposed in the literature, with different levels of trade-off between effectiveness and complexity [27]. One of the most popular approach is the Nearest Neighbor (NN) algorithm, that associates to the measured feature the map feature with the smallest Mahalanobis distance. In this work, a modified NN algorithm, specifically tailored to linear features, has been adopted [8]. First, three validation gates on the distance, orientation and overlapping of the features are employed to determine beforehand the candidate pairings. Then, the NN algorithm is run among all the feature associations that have passed the validation gates.

Let $\mathbf{m}_i$ be the i-th measurement taken by an agent (expressed in the robot local reference frame), defined as in (8). Consider the estimate $\hat{\mathbf{x}}_{f_j}$ of the j-th feature present in the map, containing the endpoints of the estimated segment in feature space. In order to associate a measurement to a feature, the Mahalanobis distance M_{ij} is evaluated, between the actual measurement $\mathbf{m}_i$ and the predicted measurement $\hat{\mathbf{m}}_j$ from the j-th feature estimate $\hat{\mathbf{x}}_{f_j}$. The measurement prediction can be easily computed from the current robot estimate $\hat{\mathbf{x}}_R$ as

$$\hat{\mathbf{m}}_j = h(\hat{\mathbf{x}}_R, \hat{\mathbf{x}}_{f_j}). \tag{13}$$

If $\mathbf{e}_{ij} = \mathbf{m}_i - \hat{\mathbf{m}}_j$ is the prediction error, then M_{ij} takes on the form $M_{ij} = \mathbf{e}_{ij}^T(\Lambda S_{ij} \Lambda^T)^{-1}\mathbf{e}_{ij}$ where $S_{ij} = E[\mathbf{e}_{ij}\mathbf{e}_{ij}^T]$ can be computed from the sensor noise covariance $R_{\mathbf{m}_i}$ and the estimation error covariance $P_{\mathbf{x}_s}$, by linearizing the relationship (13). The matrix $\Lambda = [\Lambda_\alpha \ \Lambda_\rho]^T$, where $\Lambda_\alpha = [1 \ 0 \ 0 \ 0]^T$ and $\Lambda_\rho = [0 \ 1 \ 0 \ 0]^T$, removes from the computation of the error e_{ij} the segment endpoints d_A and d_B possibly present in the measurement $\mathbf{m}_i$. Hence, the Mahalanobis distance M_{ij} takes care only of the ρ and α parameters of the line. On the contrary, the segment endpoints are used to compute the overlapping rate τ_{ij} between the extracted segment i and the one associated to the i-th feature, normalized between 0 and 1. The latter quantity allows the matching module to discriminate between different features lying on the same line (e.g., two walls separated by a door). The features in the map which are candidate to be associated to the given measurement are selected by evaluating the following indicators:

- line orientation error: $M_{\alpha,ij} = \dfrac{(\Lambda_\alpha^T \mathbf{e}_{ij})^2}{\Lambda_\alpha^T S_{ij} \Lambda_\alpha}$;
- line distance error: $M_{\rho,ij} = \dfrac{(\Lambda_\rho^T \mathbf{e}_{ij})^2}{\Lambda_\rho^T S_{ij} \Lambda_\rho}$;
- overlapping rate: τ_{ij}.

At this point, the data association mechanism can be summarized as follows. Given a measurement $\mathbf{m}_i$, a feature $\hat{\mathbf{x}}_{f_j}$ present in the map is a candidate to the matching if and only if $M_{\alpha,ij} < T_\alpha$, $M_{\rho,ij} < T_\rho$ and $\tau_{ij} > T_\tau$, where the thresholds T_α, T_ρ and T_τ are tuning knobs of the algorithm. Then, among all candidate features, the measurement is associated to the one with the smallest Mahalanobis distance M_{ij}. A measurement with zero candidate features is considered taken with respect to a new feature. Hence a new item is inserted into a tentative list of newly discovered features, waiting for to be promoted into the state of the filter when deemed reliable enough.

4. Multi-robot SLAM

Suppose that there are two robots, R_1 and R_2, exploring an unknown area. Initially, each agent runs a single-robot SLAM algorithm like that sketched in Section 3. This way, the robots have maps of the environment in different reference frames. When the robots meet, they exchange each other their local maps to produce a single global map. In order to fuse maps created by different robots, whose initial poses are unknown, the transformation between their reference frames needs to be determined. This can be efficiently done if robot-to-robot mutual measurements are available.

The overall map fusion procedure can be summarized in three steps. First, the two maps are aligned according to the estimated roto-translation relating the robot reference frames. Then, the covariance matrices of the estimation

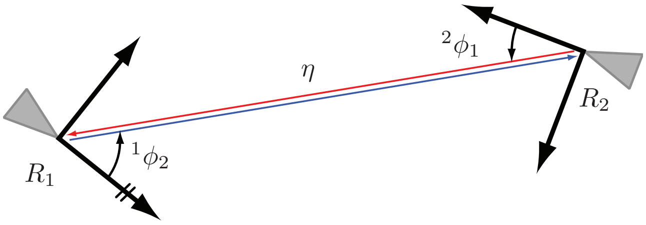


Figure 3: Robot-to-robot measurements.

error of the maps are updated according to the alignment transformation. Finally, duplicate features, due to partial overlapping between the local maps, are sought. In such a case, this information is used to impose constraints that improve the accuracy of the resulting map. The map fusion procedure is described for a team composed of two robots, but can be applied to larger teams by repeating the procedure for each pair of robots.

4.1. Map alignment

At the rendezvous, let ${}^1\mathbf{x}_1 \in \mathbb{R}^{m_1}$ and ${}^2\mathbf{x}_2 \in \mathbb{R}^{m_2}$, where $m_1 = 3 + 4n_1$ and $m_2 = 3 + 4n_2$, be the estimate of the state vector $\mathbf{x}$, built by robot R_1 and R_2, respectively. The integers n_1 and n_2 are the number of features present in each map.

The map alignment problem consists in finding the roto-translation between the reference frames $\langle R_1 \rangle$ and $\langle R_2 \rangle$, in order to express the map estimated by a robot in the frame of the other one. Without loss of generality, suppose to be interested in estimating the vector ${}^1\mathbf{x}_2$, i.e. the map of robot R_2 expressed in the frame $\langle R_1 \rangle$. This problem can be tackled by processing robot-to-robot measurements. When the agents are within sensing distance, robot i measures the range and the bearing to the vehicle j (see Figure 3):

$$ {}^i\bar{\mathbf{z}}_j = \begin{bmatrix} \eta \\ {}^i\phi_j \end{bmatrix} + \begin{bmatrix} \varepsilon_{{}^i\eta} \\ \varepsilon_{{}^i\phi_j} \end{bmatrix} \quad i, j = 1, 2, \quad i \neq j, $$

where η is the distance between the two robots, ${}^i\phi_j$ is the angle under which robot R_i sees robot R_j. Measurement errors $\varepsilon_{{}^i\eta}$ and $\varepsilon_{{}^i\phi_j}$ are modeled as zero-mean, white noise. A more accurate estimate of the distance between the two

Quantity	M-Space	Feature space
Segment coordinates	$\mathbf{x}_p$	$\mathbf{x}_f$
Single-robot state vector	$\mathbf{x}_s = [\mathbf{x}_R^T \ \mathbf{x}_{p_1}^T \ \dots \ \mathbf{x}_{p_n}^T]^T$	$\mathbf{x} = [\mathbf{x}_R^T \ \mathbf{x}_{f_1}^T \ \dots \ \mathbf{x}_{f_n}^T]^T$
Multi-robot state vector	$\mathbf{X}_s = [{}^1\mathbf{x}_{s_1}^T \ {}^2\mathbf{x}_{s_2}^T]^T$	$\mathbf{X} = [{}^1\mathbf{x}_1^T \ {}^2\mathbf{x}_2^T]^T$
Estimation error covariance	$P_s = E[\widetilde{\mathbf{X}}_s \widetilde{\mathbf{X}}_s^T]$	$P = E[\widetilde{\mathbf{X}} \widetilde{\mathbf{X}}^T]$
Aligned state vector	$\mathbf{X}_s^a = [{}^1\mathbf{x}_{s_1}^T \ {}^1\mathbf{x}_{s_2}^T]^T$	$\mathbf{X}^a = [{}^1\mathbf{x}_1^T \ {}^1\mathbf{x}_2^T]^T$
Estimation error covariance	$P_s^a = E[\widetilde{\mathbf{X}}_s^a (\widetilde{\mathbf{X}}_s^a)^T]$	$P^a = E[\widetilde{\mathbf{X}}^a (\widetilde{\mathbf{X}}^a)^T]$

Table 1: Correspondence of symbols in M-Space and in feature space.

robots can be computed as the weighted average of the two distance measurements, thus obtaining the combined measurement vector

$$\bar{\mathbf{z}} = \begin{bmatrix} \eta \\ {}^1\phi_2 \\ {}^2\phi_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_\eta \\ \varepsilon_{{}^1\phi_2} \\ \varepsilon_{{}^2\phi_1} \end{bmatrix} = \mathbf{z} + \varepsilon_{\mathbf{z}},$$

where $\mathbf{z}$ denotes the actual distance and relative bearings, and $\varepsilon_{\mathbf{z}}$ is a white measurement noise with covariance matrix

$$R_{\mathbf{z}} = E[\varepsilon_{\mathbf{z}} \ \varepsilon_{\mathbf{z}}{}^T] = \mathrm{diag}(\sigma_\eta^2, \sigma_{{}^1\phi_2}^2, \sigma_{{}^2\phi_1}^2). \tag{14}$$

Geometrical considerations on the distance and angles η, ${}^1\phi_2$, ${}^2\phi_1$ allow one to compute the exact roto-translation $\mathbf{t}$ between the reference frames $\langle R_1 \rangle$ and $\langle R_2 \rangle$ (see [19] for details):

$${}^1\mathbf{x}_2 = t({}^1\mathbf{x}_1, {}^2\mathbf{x}_2, \ \mathbf{z}). \tag{15}$$

Since the arguments of the function $t(\cdot)$ are not known exactly, the estimate ${}^1\hat{\mathbf{x}}_2$ is computed by replacing them with the corresponding estimates:

$${}^1\hat{\mathbf{x}}_2 = t({}^1\hat{\mathbf{x}}_1, {}^2\hat{\mathbf{x}}_2, \ \bar{\mathbf{z}}). \tag{16}$$

Equation (16) is a compact notation describing the overall map alignment procedure. Basically, it means that the map of robot R_2 in the reference frame $\langle R_1 \rangle$ is obtained by roto-translating all the features $\mathbf{x}_{f_i}$ present in the map according to $t(\cdot)$.

4.2. Updating map uncertainty

At the end of the map alignement procedure, each robot has a map of the overall environment explored by the two robots, in its own reference frame. However, in order for each robot to start over the navigation by running the single-robot SLAM algorithm outlined in Section 3, it is necessary to update the uncertainty associated to the aligned map. This must be done by taking into account the fact that the feature uncertainties computed so far are related to different reference frames. Hence, the covariance matrix of the aligned state vector must be modified according to the transformation performed in the map alignment stage. In the following, the map uncertainty update is described in detail.

Let us stack together the two M-Space state vectors as $\mathbf{X}_s = [{}^1\mathbf{x}_{s_1}^T \;\; {}^2\mathbf{x}_{s_2}^T]^T$ and let

$$P_s = E[\widetilde{\mathbf{X}}_s \widetilde{\mathbf{X}}_s^T] = \text{blkdiag}(P_{\mathbf{x}_{s_1}}, P_{\mathbf{x}_{s_2}}) \tag{17}$$

be the covariance matrix of the estimation error (see Table 1 for a summary of symbols used in this Section). Notice that P_s refers to the estimation errors of the two maps *before* the fusion, hence expressed in two different reference frames. In order to properly update the uncertainty of the merged map, the covariance matrix of the filter state *after* the map alignment has to be computed. Again, without loss of generality, let us consider the map merged by robot R_1. Define the M-Space *aligned state vector* $\mathbf{X}_s^a = [{}^1\mathbf{x}_{s_1}^T \;\; {}^1\mathbf{x}_{s_2}^T]^T$, and let

$$P_s^a = E[\widetilde{\mathbf{X}}_s^a (\widetilde{\mathbf{X}}_s^a)^T] \tag{18}$$

be the covariance matrix of the corresponding estimation error $\widetilde{\mathbf{X}}_s^a$. The objective is to show how the matrix P_s^a can be computed from the map transformation $t(\cdot)$ in (15) and the covariance matrix P_s in equation (17). Since the roto-translation $t(\cdot)$ relates quantities expressed in the feature space, the map uncertainty update will be performed in the same space. To this purpose, let us introduce two auxiliary covariance matrices

$$P = E[\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T], \tag{19}$$

$$P^a = E[\widetilde{\mathbf{X}}^a(\widetilde{\mathbf{X}}^a)^T], \tag{20}$$

where $\widetilde{\mathbf{X}}$ is the estimation error of vector $\mathbf{X} = [{}^1\mathbf{x}_1^T \ {}^2\mathbf{x}_2^T]^T$, and $\widetilde{\mathbf{X}}^a$ is the estimation error of vector $\mathbf{X}^a = [{}^1\mathbf{x}_1^T \ {}^1\mathbf{x}_2^T]^T$, both expressed in feature space. Matrices P and P^a represent the feature space counterpart of matrices (17) and (18), respectively. The covariance matrix update can be broken down into three steps:

1. project the covariance matrix P_s in the M-Space to the covariance matrix P in the feature space ;
2. update the covariance matrix P in the feature space according to the map alignment, thus obtaining P^a;
3. project the covariance matrix P^a back to its counterpart P_s^a in the M-Space.

The first task can be accomplished by resorting to the projection equations (3), which relate the estimation error in the M-Space $\widetilde{\mathbf{X}}_s$ and the estimation error in the feature space $\widetilde{\mathbf{X}}$. In fact, let ${}^1\mathbf{x}_{f_i,1}$ be a feature in the map of robot R_1 expressed in the frame $\langle R_1 \rangle$, and let ${}^2\mathbf{x}_{f_j,2}$ be a feature in the map of R_2 expressed in $\langle R_2 \rangle$. The estimation errors can be projected from the M-Space to the feature space according to (3) as

$${}^1\tilde{\mathbf{x}}_{f_i,1} = \widetilde{B}({}^1\hat{\mathbf{x}}_{f_i,1}) \ {}^1\tilde{\mathbf{x}}_{p_i,1}, \tag{21}$$

$${}^2\tilde{\mathbf{x}}_{f_j,2} = \widetilde{B}({}^2\hat{\mathbf{x}}_{f_j,2}) \ {}^2\tilde{\mathbf{x}}_{p_j,2}. \tag{22}$$

By stacking all the features, equations (21)-(22) can be rewritten as

$$\widetilde{\mathbf{X}} = \widetilde{D}\widetilde{\mathbf{X}}_{\mathbf{s}}, \tag{23}$$

where $\widetilde{D}$ is a block diagonal matrix, defined as

$$\begin{aligned}
\widetilde{D} &= \mathrm{blkdiag}({}^1\widetilde{D}_1, {}^2\widetilde{D}_2), \\
{}^1\widetilde{D}_1 &= \mathrm{blkdiag}(I_3, \widetilde{B}({}^1\hat{\mathbf{x}}_{f_1,1}), \ldots, \widetilde{B}({}^1\hat{\mathbf{x}}_{f_{n_1},1})), \\
{}^2\widetilde{D}_2 &= \mathrm{blkdiag}(I_3, \widetilde{B}({}^2\hat{\mathbf{x}}_{f_1,2}), \ldots, \widetilde{B}({}^2\hat{\mathbf{x}}_{f_{n_2},2})).
\end{aligned}$$

From (23) and the definitions (17) and (19), it follows that the feature space covariance matrix P corresponding to the M-Space covariance matrix P_s is given by

$$P = \widetilde{D} P_s \widetilde{D}^T.$$

The second step consists in computing the covariance matrix P^a of the estimation error of the aligned maps, in the feature space. By recalling the definition of the aligned state vector

$$\mathbf{X}^a = \begin{bmatrix} {}^1\mathbf{x}_1 \\ {}^1\mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{x}_1 \\ t({}^1\mathbf{x}_1, {}^2\mathbf{x}_2,\ \mathbf{z}) \end{bmatrix},$$

the estimation error $\widetilde{\mathbf{X}}^a$ can be derived as a function of the estimation error $\widetilde{\mathbf{X}}$. By linearizing equation (15), one gets

$$\widetilde{\mathbf{X}}^a = \begin{bmatrix} I_{m_1} & 0_{m_1 \times m_2} \\ T_1 & T_2 \end{bmatrix} \tilde{\mathbf{X}} + \begin{bmatrix} 0_{m_1 \times 3} \\ \Gamma_2 \end{bmatrix} \varepsilon_{\mathbf{z}}, \tag{24}$$

where the matrices T_1, T_2 and Γ_2 are the Jacobians of the transformation $t(\cdot)$ with respect to ${}^1\mathbf{x}_1$, ${}^2\mathbf{x}_2$ and $\mathbf{z}$, respectively, computed at the estimates ${}^1\hat{\mathbf{x}}_1$, ${}^2\hat{\mathbf{x}}_2$ and at the measurement $\bar{\mathbf{z}}$. The interested reader is referred to [28] for the analytical expression of matrices T_1, T_2 and Γ_2. From (19), (20) and (24), the covariance of the aligned augmented vector is given by

$$P^a = TPT^T + \Gamma R_{\mathbf{z}} \Gamma^T, \tag{25}$$

where

$$T = \begin{bmatrix} I_{m_1} & 0_{m_1 \times m_2} \\ T_1 & T_2 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} 0_{m_1 \times 3} \\ \Gamma_2 \end{bmatrix},$$

and the matrix $R_{\mathbf{z}}$ is defined in (14).

The third step can be tackled similarly to what has been done in the first step. By exploiting again equation (3), the estimation error of the aligned vector in the M-Space can be written as

$$\widetilde{\mathbf{X}}^a_s = D \widetilde{\mathbf{X}}^a, \tag{26}$$

where

$$\begin{aligned} D &= \text{blkdiag}({}^1D_1, {}^1D_2), \\ {}^1D_1 &= \text{blkdiag}(I_3, B({}^1\hat{\mathbf{x}}_{f_1,1}), \ldots, B({}^1\hat{\mathbf{x}}_{f_{n_1},1})), \\ {}^1D_2 &= \text{blkdiag}(I_3, B({}^1\hat{\mathbf{x}}_{f_1,2}), \ldots, B({}^1\hat{\mathbf{x}}_{f_{n_2},2})). \end{aligned}$$

and ${}^1\hat{\mathbf{x}}_{f_i,2}$ have been obtained via (16). Finally, from (18) and (26) the covariance matrix P_s^a can be computed as

$$P_s^a = DP^aD^T,$$

where P^a is given by (25). The overall covariance matrix update procedure is summarized in Figure 4.

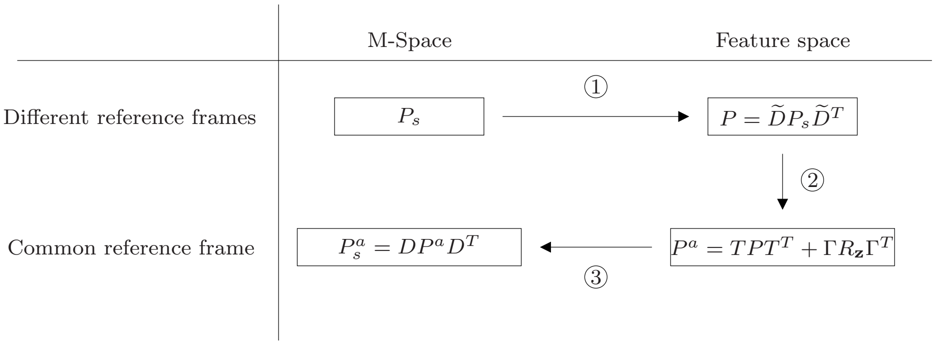

M-Space
Feature space
Different reference frames
Common reference frame
P_s
①
$P = \widetilde{D} P_s \widetilde{D}^T$
②
$P_s^a = DP^aD^T$
$P^a = TPT^T + \Gamma R_{\mathbf{z}} \Gamma^T$
③

Figure 4: Basic steps of the overall map uncertainty update procedure.

4.3. Map fusion

The matching algorithm described in Section 3.1 comes in handy for the map fusion too. As a matter of fact, it is very likely that the areas covered by the two robots before rendezvous share common regions (e.g., the neighborhood of the meeting point). This implies that a number of features will appear as duplicates in the new vector $\mathbf{X}^a$ after the map alignment. In this case, redundant features must be removed from the map, and the state of the filter, as well as the corresponding covariance matrix, properly reduced. In order to detect duplicate features, for each $\hat{\mathbf{x}}_{f_i,1}$ coming from the map of robot R_1, the feature $\hat{\mathbf{x}}_{f_j,2}$ coming from the map of R_2 with smallest Mahalanobis distance is searched for. If the two segments are close enough, and overlap significantly, then they are considered the same feature.

It is important to notice that removing duplicate features can actually improve the alignment and the accuracy of the final map. To illustrate how this can be done, suppose that in the map merged by robot R_1 the features $\hat{\mathbf{x}}_{f_i,1}$ and $\hat{\mathbf{x}}_{f_j,2}$ match. The one imported in the map during the fusion, i.e. $\hat{\mathbf{x}}_{f_j,2}$, is used as a *pseudo-measurement* of the corresponding feature $\hat{\mathbf{x}}_{f_i,1}$. The estimate $\hat{\mathbf{x}}_{f_j,2}$ is treated like a measurement of the state component $\hat{\mathbf{x}}_{f_i,1}$. In order to fuse the information contained in both estimates, a correction step of the EKF is performed by processing the pseudo-measurement $\hat{\mathbf{x}}_{f_j,2}$. Finally, the feature $\hat{\mathbf{x}}_{f_j,2}$ is removed from the vector $\mathbf{X}^a$. This procedure is repeated for all matching features between the two aligned maps. It is interesting to note that processing the pseudo-measurement has a twofold effect. Clearly, the uncertainty of the estimate resulting from the fusion of $\hat{\mathbf{x}}_{f_i,1}$ and $\hat{\mathbf{x}}_{f_j,2}$ is smaller than the uncertainty of both original estimates. Less obvious is the improvement of the map alignment that is observed after the EKF correction step. Even just few features present in both maps before the rendezvous can significantly enhance the map registration and reduce the overall uncertainty of the final map. This is due to the correlation existing among the feature estimates in both maps before the fusion, and it is a well-known phenomenon occurring in the SLAM problem [29]. As a result, updating the estimate of a duplicate feature affects the estimates of the entire imported map, which undergoes a sort of rigid roto-translation. Analogously, the uncertainty reduction of the fused feature estimates results in an improvement of the accuracy of the overall map. An example of such a behavior is illustrated in the next section (see Figure 6).

5. Results

The multi-robot SLAM algorithm has been extensively tested both in simulations and in real-world experiments. The former allow one to quantify the performance of the estimation algorithm, in terms of map accuracy, robot localization and consistency of the estimates. The latter are used to assess the viability of the proposed approach in presence of a number of uncertainty sources and non modeled phenomena.

5.1. Simulations

Initially, the proposed technique has been tested in a simulated scenario, with the aid of a custom MATLAB simulator developed on purpose. Robots are modeled as unicycles, and they are supposed to be equipped with a laser rangefinder. In this case, the robot motion model (4) takes on the form

$$x_R(k+1) = x_R(k) + \Delta_T(v(k) + \epsilon_v(k))\cos(\theta_R(k)), \tag{27}$$

$$y_R(k+1) = y_R(k) + \Delta_T(v(k) + \epsilon_v(k))\sin(\theta_R(k)), \tag{28}$$

$$\theta_R(k+1) = \theta_R(k) + \Delta_T(\omega(k) + \epsilon_\omega(k)), \tag{29}$$

where Δ_T is the sampling time and $\epsilon_v(k)$ and $\epsilon_\omega(k)$ model the noise affecting the linear and angular speed. Raw laser data, consisting of range and bearing measurements taken from planar scans of the robot surroundings, are synthetically generated by a ray tracing algorithm applied to a CAD map of the environment. The laser field of view is limited to 180°, with an angular resolution of 1° and a maximum measurable distance of 8 m. The simulations are carried out in a simplified map resembling the "S. Niccolò" building (see gray thin line in Figure 5), which hosts the Department of Information Engineering of the University of Siena (about 3000 m^2). All the noise covariances have been tuned according to the mobile robot Pioneer 3AT and its sensory equipment. The covariance matrix Q of the robot motion noise is diagonal, and the standard deviations of the velocity errors are set to be proportional to the absolute value of the speed, i.e. $Q(k) = \mathrm{diag}(\sigma_v^2(k), \sigma_\omega^2(k))$, where

$$\sigma_v(k) = 0.03|v(k)| \ (m/s),$$

$$\sigma_\omega(k) = 0.05|\omega(k)| + 0.0017 \ (rad/s).$$

The covariance matrix of the measurement noise associated to each raw range and bearing laser reading is $R_l = \mathrm{diag}\left(\sigma_r^2,\ \sigma_b^2\right)$, where $\sigma_r = 0.003$ (m) and

$\sigma_b = 0.003$ (rad). The measurements $\mathbf{m}(k)$ in (8) are extracted from the raw laser data, together with the associated error covariance matrix $R_{\mathbf{m}}(k)$, through a segmentation procedure [8]. In order to determine whether a measurement $\mathbf{m}(k)$ refers to a line already present in the map or it is a newly detected feature, the data-association algorithm described in Section 3.1 is employed. Moreover, to avoid including spurious features in the map, new lines are first inserted in a tentative list until they are deemed reliable enough. The entries of the robot-to-robot measurement noise covariance matrix $R_{\mathbf{z}}$ in (14), are set to $\sigma_\eta = 0.02$ (m), $\sigma_{{}^1\phi_2} = 0.07$ (rad) and $\sigma_{{}^2\phi_1} = 0.07$ (rad). Notice that the robot-to-robot observations are much less accurate than the laser raw readings, as it actually occurs in real-world experiments.

Figures 5-7 report the results of a typical run. The robots first explore the area around two different courtyards (named A and B in Figure 5), and then meet to fuse the two maps somewhere in the middle between the two courtyards. Then, the experiment goes on with each robot exploring the area covered before by the other robot. The information gained during the map fusion allows each robot to stay localized quite well along the entire path.

The overall map fusion procedure is summarized in Figure 6. Figure 6(a) shows the map of robot R_1 (bottom, blue) and that of robot R_2 transformed in the frame $\langle R_1 \rangle$ (top, red) at the rendezvous. Notice that the imported map is misaligned due to the uncertainty affecting the robot-to-robot measurements. This reflects in large confidence ellipses of the segment endpoints present in the map of R_2, extracted from matrix P^a in (20). Common features present in both maps before the fusion, and corresponding to areas explored by both agents, may result in duplicate features in the merged map (see segments between the two courtyards in Figure 6). As described in Section 4.3, possible duplicate features that are detected in the merged map after the alignment can be used as constraints to improve the map fusion. The resulting map after removing duplicate features is shown in Figure 6(b). Notice how the two courtyards appear now more aligned (as they really are), and the uncertainty of the whole map is significantly reduced as shown by the smaller size of the confidence ellipses. In

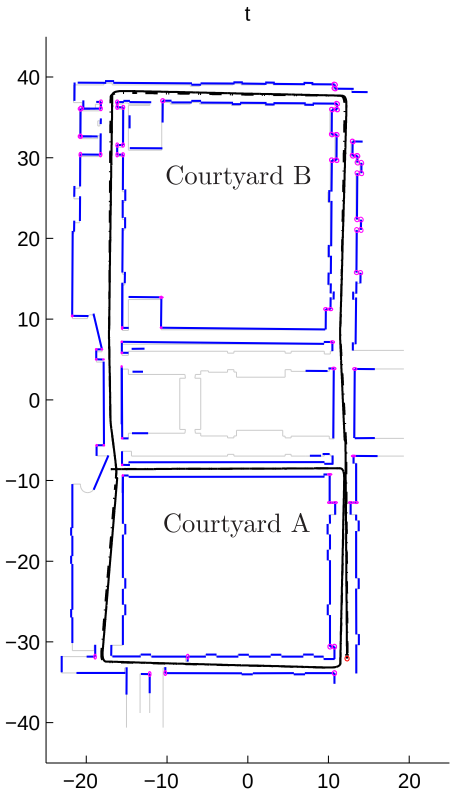
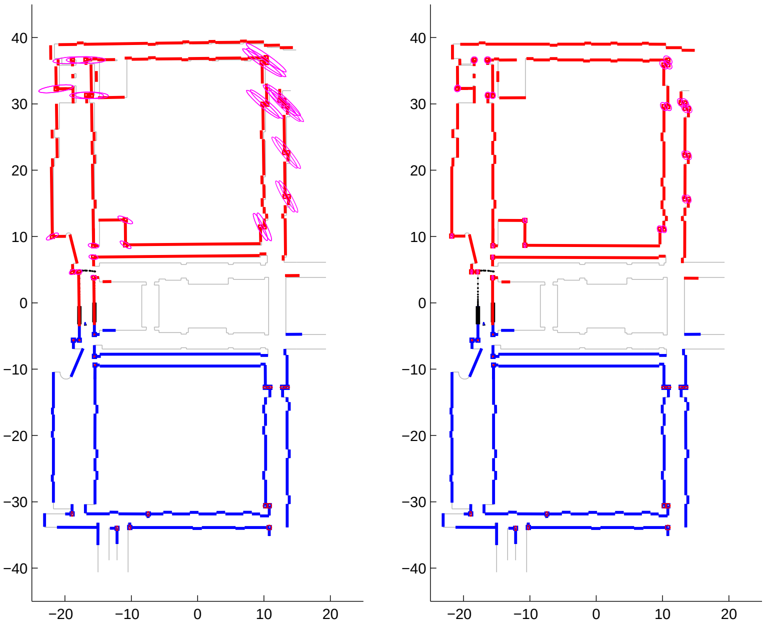


Figure 5: Simulation of the multi-robot SLAM algorithm: CAD map of the environment (thin gray line), path traveled by robot R_1 (true: dash-dotted; estimated: solid), final map built by R_1 at the end of the simulation (blue segments), 99% confidence ellipses of the the identified segment endpoints (pink ellipses).

this regard, it is worth remarking that the estimates of the features present only in the imported map (in this case, the features of the map of robot R_2) are still affected by larger uncertainty, due to the noisy robot mutual measurements. As a consequence, the merged map will be made up of estimates whose average uncertainty can be larger than that of the original single robot map. In a sense, this is the price to pay for acquiring information about unvisited regions of the environment, and hence to shorten the exploration completion time. The x, y and θ localization errors of robot R_1 are shown in Figure 7, along with the 3σ



(a) (b)

Figure 6: Incorporating the map built by robot R_2 (red) into the map of robot R_1 (blue): (a) the resulting map after the alignment procedure (Section 4.1- 4.2) and (b) the merged map after removing duplicate features (Section 4.3).

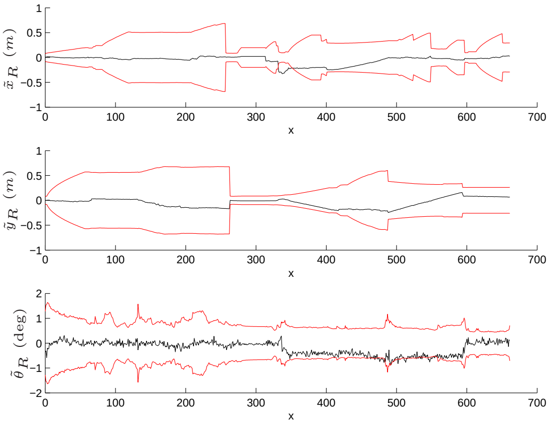


Figure 7: The robots $\tilde{x}_R$, $\tilde{y}_R$ and $\tilde{\theta}_R$ localization errors, along with the correspondent 3σ confidence intervals (robot R_1).

confidence intervals. It can be observed that most of the times the estimates are consistent.

A campaign of simulations has been carried out to evaluate the performance of the proposed multi-robot SLAM algorithm. The environment is the same depicted in Figure 5. In Tables 2-3 the results of 100 simulation runs are reported. Columns S_1 and S_2 refer to single-robot SLAM performed by robot R_1 and R_2, respectively. Columns M_1 and M_2 correspond to multi-robot SLAM by R_1 and R_2, respectively. Columns MH_1 and MH_2 in Table 3 refer to the map available to robot R_1 and R_2 at half time of the experiment (after the rendezvous and map fusion). The tables report the average absolute errors of the robot and feature estimates.

Robot pose errors				
	S_1	S_2	M_1	M_2
Position error (m)	0.24	0.19	0.24	0.23
Orientation error (deg)	0.50	0.37	0.50	0.45

Table 2: Robot pose estimation error: average value over 100 simulation runs.

Map errors						
	S_1	S_2	M_1	M_2	MH_1	MH_2
Line parameter α error (deg)	0.48	0.36	0.45	0.43	0.42	0.38
Line parameter ρ error (m)	0.09	0.07	0.11	0.09	0.10	0.08
Endpoint error (m)	0.25	0.17	0.25	0.22	0.25	0.19
Number of features	182	183	205	201	178	179
Number of endpoints	84	92	131	122	121	121

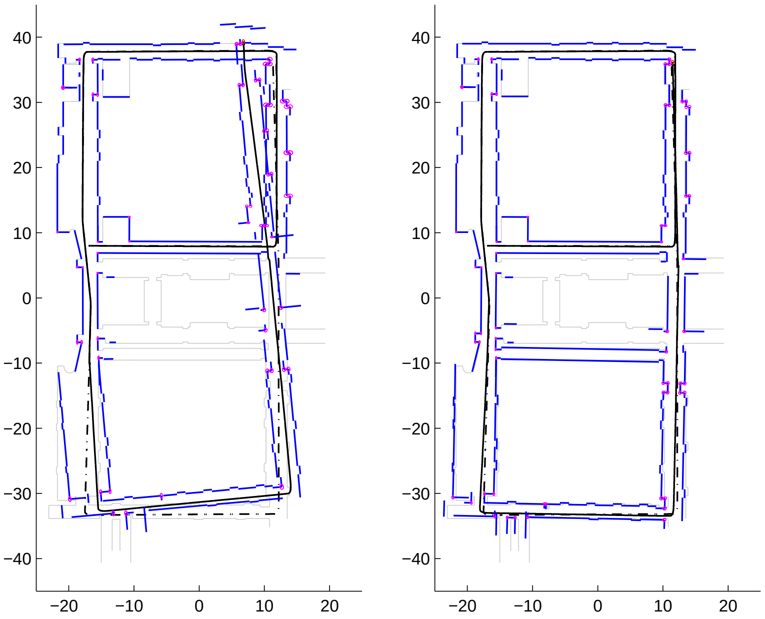
Table 3: Map estimation error and number of features: average value over 100 simulation runs.

It can be observed that the estimation errors in the single- and multi-robot case are comparable, the robot pose estimates being slightly better in the single-robot case. In particular, the map cooperatively built by the multi-robot SLAM approach shows at half time of the experiments the same level of accuracy of the map provided by the single-robot SLAM at the end of the experiment. This confirms the viability of the proposed multi-robot approach. Notice that the absolute values of the errors are significantly small if compared to the size and the complexity of the environment. Table 3 also shows that the multi-robot approach is able to detect a larger number of features (and in particular a larger number of segment endpoints), which results in a more detailed map, without worsening the average accuracy of the map itself.

The results shown in the previous tables refer to experiments in which the number of duplicate features is usually very small and the estimated map is in good agreement with the true map of the environment. This is due to the fact that no incorrect line feature matching has occurred, both in the single-robot and in the multi-robot case. However, in the simulation campaign several odd runs have been experienced, showing a much larger map estimation error in the single-robot SLAM. It has been checked that this behavior is due to wrong feature associations, which eventually lead to large map errors. An example is shown in Figure 8(a), where the remarkable misalignment in the map produced by robot R_1 is due to the fact that a newly detected feature has been wrongly associated to a different feature already present in the map. This does not occur in the multi-robot case (see Figure 8(b)) because the same feature had been already detected by robot R_2 before the rendezvous and hence has been inherited by robot R_1 during the map fusion. It is well known that correct data association is crucial for a successful map construction in feature-based SLAM. In this respect, one may claim that using multiple robots helps the matching mechanism to avoid wrong associations, thus preserving a good quality of the overall map.

5.2. Experiments

A set of experiments with real robots has also been performed, to validate the multi-robot SLAM algorithm in a real-world setup. To this purpose, two mobile robots Pioneer 3AT equipped with a laser rangefinder have been used [30]. All the experiments have been carried out in the second floor of the S. Niccolo' building, whose simplified CAD map is shown in Figure 5. The robot motion model adopted in the EKF is given by equations (27)-(29), and all the param-



(a) (b)

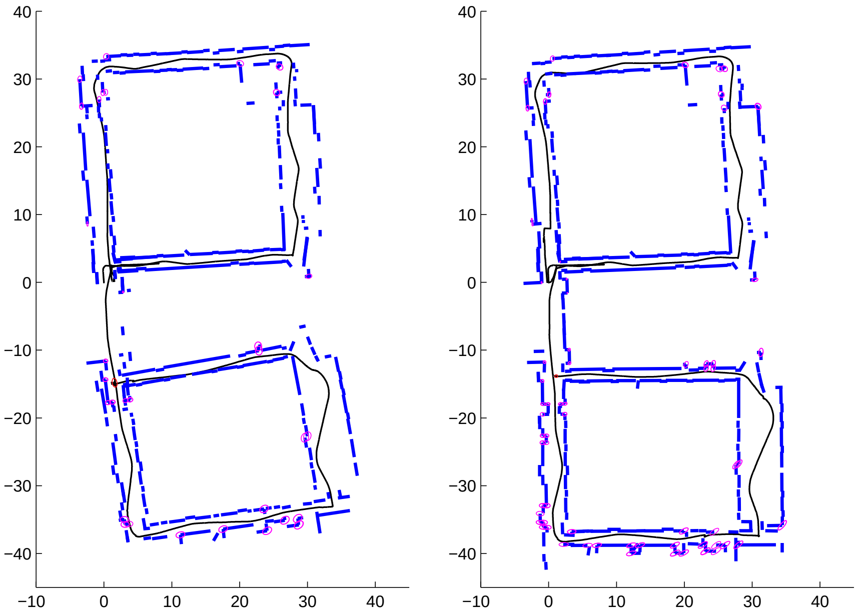
Figure 8: Effect of incorrect feature matching: (a) single-robot map, (b) multi-robot map.

eters of the filter are set to the same value used in the simulations described before. Also the path followed by each vehicle is similar to that chosen for the simulations, i.e. each robot explores a different courtyard and then merges its map with that of the other agent in proximity of the middle of the building. Figure 9 shows a comparison of the multi-robot algorithm with respect to the single-robot case, in terms of quality of the final map built. Specifically, the map depicted in Figure 9(a) results from the single-robot SLAM performed by robot R_1, whereas in Figure 9(b) the final map yielded by the multi-robot SLAM algorithm is shown. Although a detailed quantitative comparison is not possible since the ground truth of the robots and an accurate map of the building are not available, a look at Figure 9 suggests that the multi-robot algorithm is actually able to provide an improvement to the built map, also in real environments and with real robots. In fact, in spite of a number of uncertainty sources present in reality and often neglected in simulations, the employment of two coordinated robots results in a more reliable map, as is testified by the better alignment of the two courtyards. The experimental results obtained are in good agreement with the simulations previously performed. In the multi-vehicle case, the average area of the 99% confidence ellipses of the estimates of the segment endpoints is smaller than 5 cm^2. At the end of the experiment, the uncertainty affecting the estimate of the robot pose is smaller than 10 cm and 0.5 deg for the x, y and θ coordinates, respectively.

Besides the advantage provided by the multi-robot SLAM algorithm of being able to build a map of the environment much faster than its single-robot counterpart, an additional benefit has been observed during the tests. As is well known, closing large loops is a challenging task when it comes to SLAM. As matter of fact, in several experiments the single-robot SLAM algorithm proved to be unable to close the largest loop around the two courtyards, a rectangular path about 200 m long. In this respect, the multi-robot SLAM technique can benefit from the information on unexplored areas shared during the map fusion stage. When the two robots start exploring different courtyards and then merge their maps at rendezvous, the map fusion has the effect of "shortening the length of the loop", since the imported map conveys information about the remaining part of the loop. This behavior observed in the experiments is actually in agreement with what has been experienced in simulation runs like the one depicted in Figure 8.

5.3. Statistics of repeated experiments

A multi-robot experiment like that described before has been repeated 10 times to extract some statistics on the performance in a real-world setup. Both the robot localization capability and the map accuracy have been taken into account. Concerning the robot localization, the standard deviation of the estimates of the robot pose have been computed, averaged over the whole path and over all the 10 runs. Concerning the map, the following quantities have been computed: the standard deviation of the estimates of the line parameters α and ρ (see Figure 1); the area of the 3σ confidence ellipses of the estimates of the



(a)

(b)

Figure 9: Final map at the end of an experiment in the S. Niccolò building: (a) *single-robot* SLAM, (b) *multi-robot* SLAM.

segment endpoints; the number of features present in the map. All the quantities have been averaged over all the features present in the final map and over all the 10 runs. The results are summarized in Tables 4 and 5. There, for each parameter two different scenarios are considered: single-robot SLAM (columns S_1 and S_2 corresponding to robot R_1 and R_2, respectively), and multi-robot SLAM (columns M_1 and M_2 corresponding to robot R_1 and R_2, respectively). Column S and M reports the average values for the single-robot case and the multi-robot case, respectively.

Robot pose uncertainty						
	S1	S2	M1	M2	S	M
x_R standard deviation (m)	0.20	0.13	0.22	0.18	0.17	0.20
y_R standard deviation (m)	0.18	0.16	0.20	0.18	0.17	0.19
θ_R standard deviation (deg)	0.55	0.51	0.70	0.64	0.53	0.67

Table 4: Robot pose uncertainty: sample statistics from real experiments (10 runs).

Map uncertainty						
	S_1	S_2	M_1	M_2	S	M
Line parameter α STD: σ_α (deg)	0.54	0.52	0.53	0.51	0.53	0.52
Line parameter ρ STD: σ_ρ (m)	0.29	0.24	0.28	0.25	0.27	0.27
Endpoint 3σ confidence ellipses (cm^2)	4.31	2.84	3.32	5.28	3.58	4.30
Average number of features	139	132	171	178	136	174

Table 5: Map uncertainty: sample statistics from real experiments (10 runs).

By inspecting the table entries, a rough comparison between the single-robot and the multi-robot SLAM algorithm can be made. As far as the robot localization is concerned (Table 4), the two algorithms seem to perform basically the same, with the single-robot being slightly better. Again, this is in good agreement with the simulation campaign reported in Section 5.1. More interesting is the comparison on the map building process. The expected quality of the map in both cases seems to be very similar (compare the last two columns of Table 5), with a small increment of the confidence ellipses in the multi-robot case. Nonetheless, in light of the considerations made in Sections 5.1 and 5.2, this means that the many advantages brought in by the multi-robot framework are paid at little or almost no cost. Despite the absence of information on the relative initial robot poses, the map fusion technique based on robot mutual measurements is able to effectively share the whole information available at the time of rendezvous among the two robots. The effectiveness of the map fusion is also confirmed by the number of features present in the final map (fourth row of Table 5). The increase in the multi-robot scenario (174 vs. 136) is due to the fact that in general, at the end of an experiment, the total area covered with two robots is larger than that explored with a single one. At the same time, the number of features in the multi-robot case is small enough to conclude that the map imported at the rendezvous is correctly exploited when the robot reach the corresponding region of the environment, i.e. the region is recognized as already visited and no redundant features are inserted in the map.

6. Conclusions and future work

A multi-robot SLAM algorithm for linear features has been presented. The proposed solution takes advantage of the M-Space framework for efficiently parametrizing lines and segments, and exploits mutual robot measurements to merge local maps during a rendezvous. Among the benefits of the adopted representation, there is the possibility to early initialize feature estimates and to exploit measurements of partially observed features. The map merging technique, which does not require any *a priori* knowledge on the initial relative robot pose, has been suitably arranged to fit the specific needs of the chosen feature representation. In particular, the uncertainty of the merged map has been analytically computed in order to be used by the single-robot SLAM algorithm later on. Simulations and experimental tests have shown that the proposed approach enjoys the speed and robustness characteristics typical of multi-robot architectures, while at the same time preserving the quality of the final map.

Several aspects are currently under investigation. To fully exploit the versatility of the M-Space representation paradigm, the map description is going to be enriched by including new kind of features, like corners or poles. Simulations and experimental tests involving larger teams of robots are planned to evaluate the behavior of the map merging scheme after multiple map fusions. In this respect, an open issue is how to avoid multiple processing of the same information, due to repeated map fusions between the same robots, which clearly affects the consistency of the estimates. Finally, more sophisticated data association algorithms, like the joint compatibility test, could bring in a significant improvement of the merged map quality by identifying a larger number of duplicate features.

References

[1] J. A. Castellanos, J. D. Tardos, Mobile Robot Localization and Map Building: A Multisensor Fusion Approach, Kluwer Academic Publisher, Boston, 1999.

[2] S. Thrun, Robotic mapping: A survey, in: G. Lakemeyer, B. Nebel (Eds.), Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, 2002, pp. 1–35.

[3] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, MIT Press, 2005.

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

[5] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): Part II, IEEE Robotics & Automation Magazine 13 (3) (2006) 108–117.

[6] U. Frese, A discussion of simultaneous localization and mapping, Autonomous Robots 20 (1) (2006) 25–42.

[7] A. T. Pfister, S. I. Roumeliotis, J. W. Burdick, Weighted line fitting algorithms for mobile robot map building and efficient data representation, in: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taiwan, 2003, pp. 1304–1311.

[8] A. Garulli, A. Giannitrapani, A. Rossi, A. Vicino, Mobile robot SLAM for line-based environment representation, in: Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference, 2005, pp. 2041–2046.

[9] J. Folkesson, P. Jensfelt, H. Christensen, The M-space feature representation for SLAM, IEEE Transactions on Robotics 23 (5) (2007) 1024–1035.

[10] M. Bosse, P. Newman, J. Leonard, S. Teller, Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework, The International Journal of Robotics Research 23 (12) (2004) 1113.

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

[12] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, J. Tardós, A comparison of loop closing techniques in monocular SLAM, Robotics and Autonomous Systems 57 (12) (2009) 1188–1197.

[13] J. Fenwick, P. Newman, J. Leonard, Cooperative concurrent mapping and localization, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2002, pp. 1810–1817.

[14] S. Williams, G. Dissanayake, H. Durrant-Whyte, Towards multi-vehicle simultaneous localisation and mapping, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2002, pp. 2743–2748.

[15] S. Thrun, Y. Liu, Multi-robot slam with sparse extended information filers, in: P. Dario, R. Chatila (Eds.), Robotics Research, Vol. 15 of Springer Tracts in Advanced Robotics, Springer Berlin / Heidelberg, 2005, pp. 254–266.

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

[17] M. Di Marco, A. Garulli, A. Giannitrapani, A. Vicino, Simultaneous localization and map building for a team of cooperating robots: a set membership approach, IEEE Transactions on Robotics and Automation 19 (2) (2003) 238–249.

[18] R. Reid, T. Braunl, Large-scale multi-robot mapping in MAGIC 2010, in: Proceedings of the 2011 IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2011, pp. 239 –244.

[19] X. Zhou, S. 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, 2006, pp. 1785–1792.

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

[21] H. Lee, S. Lee, M. Choi, B. Lee, Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses, Robotica, Available on CJO 2011 doi:10.1017/S026357471100049X.

[22] H. Chang, C. Lee, Y. Hu, Y. Lu, Multi-robot slam with topological/metric maps, in: Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 1467–1472.

[23] T. Vidal-Calleja, C. Berger, S. Lacroix, Event-driven loop closure in multi-robot mapping, in: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 1535–1540.

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

[25] A. Gil, Ó. Reinoso, M. Ballesta, M. Juliá, Multi-robot visual SLAM using a Rao-Blackwellized particle filter, Robotics and Autonomous Systems 58 (1) (2010) 68–80.

[26] D. Benedettelli, A. Garulli, A. Giannitrapani, Multi-Robot SLAM using M-Space feature representation, in: Proceedings of the 49th IEEE Conference on Decision and Control, 2010, pp. 3826–3831.

[27] Y. Bar-Shalom, T. Fortman, Tracking and data association, Academic Press, 1988.

[28] D. Benedettelli, Multi-Robot SLAM using M-Space feature representation, Master's thesis, Faculty of Engineering, University of Siena, available at http://robotics.benedettelli.com/robots/publications/BenedettelliMSc.pdf (2009).

[29] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem, IEEE Transactions Robotics and Automation 17 (3) (2001) 229–241.

[30] Adept MobileRobots, Pioneer 3-AT, http://www.mobilerobots.com (2011).