Distributed Consensus Algorithms for Merging Feature-Based Maps with Limited Communication

R. Aragues[a,*], J. Cortes[b], C. Sagues[a]

[a] *Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain*

[b] *Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, California, 92093, USA*

Abstract

In this paper we present a solution for merging feature-based maps in a robotic network with limited communication. We consider a team of robots that explore an unknown environment and build local stochastic maps of the explored region. After the exploration has taken place, the robots communicate and build a global map of the environment. This problem has been traditionally addressed using centralized schemes or broadcasting methods. The contribution of this work is the design of a fully distributed approach which is implementable in scenarios with limited communication. Our solution does not rely on a particular communication topology and does not require any central agent, making the system robust to individual failures. Information is exchanged exclusively between neighboring robots in the communication graph. We provide distributed algorithms for solving the three main issues associated to a map merging scenario: establishing a common reference frame, solving the data association, and merging the maps. We also give worst-case performance bounds for computational complexity, memory usage, and communication load. Simulations and real experiments carried out using various vision sensors validate our results.

Keywords: Distributed consensus, Map merging, Robotic networks, Cooperative strategies, Limited communication

1. INTRODUCTION

The increasing interest in multi-robot applications is motivated by the wealth of possibilities offered by teams of robots cooperatively performing collective tasks. The efficiency and robustness of these teams goes well beyond what individual robots can do. An important issue associated to the operation of a robot team is perception. In general, each robot just observes a portion of the environment. In order to make decisions in a coordinated way, the robots must fuse their local observations into a global map. Many existent solutions for single robot perception have been extended to multi-robot scenarios under centralized schemes, full communication between the robots, or broadcasting methods. Particle filters have been generalized to multi-robot systems assuming that the robots broadcast their controls and their observations [1]. The Constrained Local Submap Filter has been extended to the multi-robot case assuming that each robot builds a local submap and broadcasts it, or transmits it to a central agent [2]. Methods based on graph maps of laser scans [3, 4, 5] make each robot build a new node and broadcast it. The same solution could be applied for many existing submap approaches [6]. However, in multi-robot scenarios, distributed approaches are often necessary because of the limited communication capabilities of the agents and the need to deal with incomplete or disconnected communication graphs, switching topologies, link failures, and

*Corresponding author.

Email addresses: raragues@unizar.es (R. Aragues), cortes@ucsd.edu (J. Cortes), csagues@unizar.es (C. Sagues)

limited bandwidth. At the same time, distributed approaches introduce an additional level of complexity on the algorithm design and analysis.

In this paper, we propose distributed algorithms for the three main issues associated to the map merging problem: establishing a common reference frame, solving the data association, and merging the maps. The first issue in a map merging scenario consists of establishing a common reference frame for the whole robot team. In general, the robots start at unknown locations and do not know their relative poses. This information can be recovered by comparing their local maps and looking for overlapping regions. This approach, known as map alignment, has been deeply investigated and interesting solutions have been presented for feature-based [7] and occupancy grid [8, 9] maps. However, it has the inconvenience that its results depend on the accumulated uncertainty in the local maps. Alternatively, the relative poses between the robots can be explicitly measured [10, 11]. These methods present the benefit that the obtained results do not depend on the uncertainties in the local maps. They also allow the robots to compute their relative poses when there is no overlapping between their maps, or even if they do not actually have a map. We propose a method that belongs to this latter class and proceeds as follows. In a first phase, a method from computer vision [12] is applied to any pair of neighboring robots. It recovers the 2D position and the orientation of three cameras with the condition that they must be nearby, and that a minimal number of landmarks must be visible from all the cameras. This method operates on three views and returns the relative positions between the cameras, up to a scale factor. In a second phase, a consensus algorithm is executed that uses this relative pose information for computing the center of mass of the robot team. The communication graph during this phase must be connected. The establishment of a common reference frame needs to be carried out only once, and since the described method does not require the robots to have a map, it can be executed at any time. For instance, it can be executed after the exploration and before merging their maps, by combining it with a rendezvous strategy. Alternatively, it can be executed at an initial stage, prior to any exploration taking place. After that, the robots explore the environment and build their local maps. We do not discuss exploration strategies or map building methods, and note that any algorithm producing feature-based maps can be employed.

When the robots finish their exploration, and before merging their maps, they must solve the data association. The data association problem consists of establishing correspondences between different measurements or estimates of a common element. Traditional data association methods, like the Nearest Neighbor and Maximum Likelihood [10, 13, 14], the Joint Compatibility Branch and Bound (JCBB) [15], or the Combined Constraint Data Association [16] are designed for single robot systems. They operate on two sets of elements, one containing the feature estimates and the other one containing the current observations. Methods based on submaps [2, 17, 18] use these data association algorithms by transforming one map into an observation of a second one. Multi-robot approaches have not fully addressed the problem of data association. Many methods rely on broadcasting controls and observations or submaps, see e.g., [1, 3, 4, 5, 14, 19], and solve the data association using a cycle-free order, thus essentially reducing the problem to that of the single robot scenario. However, in a distributed map merging scenario, the robots may fuse their maps with any other robot's map in any order and at any time. Therefore, it is not possible to force a specific order for solving the data association between the local maps. Our approach here is to let each robot solve its own local association with its neighbors in the communication graph. This scenario is more flexible but may lead to inconsistent global data associations in the presence of cycles in the communication graph. These inconsistencies are detected when chains of local associations give rise to two features from one robot being associated among them. These situations must be correctly identified and solved before merging the data. Inconsistent associations have already been discussed in the computer vision literature [20], although in centralized scenarios. Here, we present a method for detecting and solving these inconsistencies under limited communication in a decentralized fashion.

Finally, once the maps are aligned into a common reference and the data association has been solved, they need to be fused. There has been an intensive recent research in distributed implementations of the Kalman Filter that make use of its information form (IF). Measurement updates in IF are additive and therefore information coming from different sensors can be fused in any order and at any time. While optimal solutions exist for complete communication networks [21], for general communication schemes [22, 23] the delayed data problem leads to an approximate KF estimator. This problem appears when the nodes execute the state

prediction without having incorporated all the measurements taken at the current step. As a result, their estimates become suboptimal and give rise to disagreement. The effects of this delayed data problem have been studied in [24]. A solution that reduces this disagreement has been presented [25] and its convergence has been proved in the absence of observation and system noises. However, this solution does not consider system inputs, which usually model odometry measurements in typical robotic applications. Therefore, it does not solve its associated delayed data problem. Other methods have been proposed that require the previous offline computation of the gains and weights of the algorithm and that are only applicable when the variance of measurement noises is constant and a priori known [26]. Due to the limitations of these methods, the formulation of a multi-robot perception problem as a distributed estimator presents multiple obstacles. Here, instead we formulate the map merging as a sensor fusion problem, where each robot can be seen as a sensor and its local map as a measurement. Instead of maintaining a global estimator, we let each robot maintain its own local estimator, i.e., build its local map using exclusively measurements acquired by itself. The information received from other robots is introduced into its estimated global map, but not into its local map. Sensor fusion approaches [27] present the inconvenience that the successive measurements, in our case local maps, from the same robot must be independent. In a map merging scenario this does not hold, since the local map of a robot is an evolution of any of its previous maps. However, since we discuss a static map merging where the maps are fused after the exploration, our approach does not suffer from this limitation. In our solution, the local maps of the robots are expressed in IF form and they are fused in an additive fashion using a consensus filter [27, 28] to provide a distributed implementation.

This work combines ideas from two important research fields, distributed sensor networks and networked robotics. We make important contributions related to the development of specific solutions to the aspects of the map merging problem. The paper organization and the specific contributions in each section are as follows. Section 2 states the distributed map merging problem and its three aforementioned subproblems. Section 3 addresses the problem of establishing a common reference frame and proposes a solution based on the computation of the center of mass of the robot team. We combine different existing methods to give a distributed algorithm and, in addition, we provide a formal proof that the algorithm to compute the center of mass position is correct. Section 4 addresses the problem of computing the data association between the maps acquired by the robots. A preliminary version of this work appeared in [29]. Here, an improved algorithm is presented that simultaneously builds a set of labels for the features which reflects the global data association. This labeling process fulfills the connection between the distributed data association and map fusion processes. Section 5 presents the map fusion algorithm which incrementally discovers the feature labeling of any other robot in the network while simultaneously merges the maps. The relationship between the outcomes of this algorithm and the ones of a distributed sensor fusion are formally compared, concluding that they are equivalent. Finally Section 6 studies the complexity of the algorithm and Section 7 evaluates its performance for merging feature-based maps obtained with both omnidirectional and conventional cameras.

2. PROBLEM DESCRIPTION

Throughout the paper we use the following notation (Table 1):

We consider a team of $n \in \mathbb{N}$ robots with limited communication capabilities. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the undirected communication graph. The nodes are the robots, $\mathcal{V} = \{1, \ldots, n\}$. If two robots i, j can exchange information, then there is an edge $(i, j) \in \mathcal{E}$ between them. Let $\mathcal{N}_i$ be the set of neighbors of robot i,

$$\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}, j \neq i\}.$$

The robots have explored an unknown environment. Along its operation, each robot i has observed $m_i \in \mathbb{N}$ features whose true positions are unknown. Based on its own observations, each robot i has estimated its own pose together with the positions of the features. It has built a stochastic map with $\mathcal{M}_i = \mathtt{szr} + m_i\ \mathtt{szf}$ elements, composed of a mean $\hat{\mathbf{x}}_i \in \mathbb{R}^{\mathcal{M}_i}$ and covariance matrix $\Sigma_i \in \mathbb{R}^{\mathcal{M}_i \times \mathcal{M}_i}$. The constants $\mathtt{szr}$, $\mathtt{szf}$ represent the size of, respectively, a robot pose and a feature position; $\mathtt{szr} = 3$ for planar motions, where the robot position (x, y) and orientation θ are estimated; $\mathtt{szf} = 2$ or $\mathtt{szf} = 3$ for respectively 2D or 3D

Table 1: Notation.

Symbol	Usage
i, i', j, j'	robot index
r, r', s, s'	feature index
G	index used for referring to the global reference frame and map
t	iteration number, $t \in \mathbb{N}$
f_r^i	the r^{th} feature observed by the i^{th} robot
$A_{r,s}$	the (r,s) entry of matrix A
A_{ij}	the block (i,j) of matrix A defined by blocks
$[A_{ij}]_{r,s}$	the (r,s) entry of A_{ij}
$\mathcal{M}_i$	size of the map of robot i
$\mathcal{M}_G$	size of the global map

environments. If $\mathbf{x}_i \in \mathbb{R}^{\mathcal{M}_i}$ is the vector with the true robot pose after the exploration and with the true positions of the m_i features, then

$$\hat{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{v}_i, \tag{1}$$

where $\mathbf{v}_i$ is a zero mean Gaussian noise $\mathbf{v}_i \sim N(0, \Sigma_i)$. The aim is to merge the local maps to obtain a global estimate of the map.

Let $m \in \mathbb{N}$ be the number of different features in the environment. As noted above, each robot $i \in \{1, \dots, n\}$ has observed $m_i \leq m$ of them. Let $\mathbf{x} \in \mathbb{R}^{\mathcal{M}_G}$ contain the true poses of the n robots and the true positions of the m features, where $\mathcal{M}_G = n\ \texttt{szr} + m\ \texttt{szf}$. Let $H_i \in \{0,1\}^{\mathcal{M}_i \times \mathcal{M}_G}$ be the observation matrix that relates the elements in $\mathbf{x}$ with the elements observed by robot i. It is a binary matrix, i.e., each entry is equal to 0 or 1, where there is at most a 1 per row. Since $\mathbf{x}_i = H_i\mathbf{x}$, eq. (1) becomes

$$\hat{\mathbf{x}}_i = H_i\mathbf{x} + \mathbf{v}_i,$$

where $\mathbf{v}_i \sim N(0, \Sigma_i)$. Considering all the local maps together, we have that

$$\left(\hat{\mathbf{x}}_1^T \dots \hat{\mathbf{x}}_n^T\right)^T = \left(H_1^T \dots H_n^T\right)^T \mathbf{x} + \left(\mathbf{v}_1^T \dots \mathbf{v}_n^T\right)^T.$$

We assume that the noises $\mathbf{v}_i$ are independent since every robot has constructed the map based on its own observations, and thus $\left(\mathbf{v}_1^T \dots \mathbf{v}_n^T\right)^T \sim N\left(0, \text{diag}\left(\Sigma_1, \dots, \Sigma_n\right)\right)$. The local map of each robot i is represented in IF form by its information matrix $I_i \in \mathbb{R}^{\mathcal{M}_G \times \mathcal{M}_G}$ and its information vector $\mathbf{i}_i \in \mathbb{R}^{\mathcal{M}_G}$,

$$I_i = H_i^T \Sigma_i^{-1} H_i, \quad \mathbf{i}_i = H_i^T \Sigma_i^{-1} \hat{\mathbf{x}}_i. \tag{2}$$

Given the n local maps in IF form, the operation that merges their information and produces the global map,

$$I_G = \sum_{i=1}^{n} I_i, \quad \mathbf{i}_G = \sum_{i=1}^{n} \mathbf{i}_i, \tag{3}$$

is additive, commutative, and associative. For this reason, merging the maps in IF form is a common practice [7]. Expressed by its mean and covariance matrix,

$$\hat{\mathbf{x}}_G = (I_G)^{-1}\ \mathbf{i}_G, \quad \Sigma_G = (I_G)^{-1}. \tag{4}$$

it corresponds [28] to the maximum likelihood estimate of $\mathbf{x}$ given the noisy observations $\hat{\mathbf{x}}_i$, $i \in \{1, \dots, n\}$.

The goal is for each robot to compute the global map (3)-(4) in a distributed fashion. We propose a map merging algorithm composed of four stages (see Algorithm 2.1 below). In the first two stages, the robots

agree on a common reference frame and explore and build their local maps relative to this frame. These first two stages are interchangeable and the common reference frame computation (Section 3) can be executed either before or after exploring. In this paper, we do not discuss the exploration strategies or the SLAM algorithms for obtaining the local maps. After this, robots run an algorithm to agree on a data association which is globally consistent (Section 4), and this is then used to merge their maps (Section 5).

Algorithm 2.1 Distributed map merging

1: Reach consensus on a global reference frame
2: Explore the environment and build a local map in the global reference frame
3: Reach consensus on a globally consistent data association
4: Reach consensus on the global merged map

3. CONSENSUS ON THE GLOBAL REFERENCE FRAME

In this section we address the problem of establishing a common reference frame for the whole robot team. This is a key step which is necessary to solve any map merging problem. We define the global reference as the center of mass of the robot team and we propose a distributed strategy where the robots compute their pose relative to the center of mass. We assume that the robots are placed at nearby positions, so that i) the communication network is connected, and ii) the number of common landmarks observed by each pair of neighboring robots is enough to retrieve their relative poses. In addition we assume that their orientations satisfy $0 < \theta_i < \pi$ for all $i \in \{1, \dots, n\}$, so that the average orientation is well defined [30]. The global reference frame computation can be carried out at any time, for instance at the initial stage. The strategy roughly consists of each robot computing its relative pose to its nearby robots and then using this data to compute its pose relative to the center of mass based on local interactions in a distributed way.

We use the notation $\mathbf{p}_i^G = (R_i^G, T_i^G) \in SE(3)$ to refer to the pose of robot i expressed in the global frame G, where $z = 0$ since the robots move on the plane, and the planar rotations are around the z-axis,

$$R_i^G = \begin{bmatrix} \cos\theta_i^G & -\sin\theta_i^G & 0 \\ \sin\theta_i^G & \cos\theta_i^G & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad T_i^G = \begin{bmatrix} x_i^G \\ y_i^G \\ 0 \end{bmatrix}, \tag{5}$$

and we let $\mathbf{p}_j^i = (R_j^i, T_j^i) \in SE(3)$ be the pose of a robot j relative to robot i. Along this section we use the generic world reference frame $\mathtt{w}$ as a tool for providing the theoretical results and definitions, and we let $\mathbf{p}_i^{\mathtt{w}} = (R_i^{\mathtt{w}}, T_i^{\mathtt{w}}) \in SE(3)$ be the pose of robot i expressed in frame $\mathtt{w}$. Note that this frame, which is unknown by the robots, is only used for deriving our proposal and it is not needed for executing the proposed algorithms. Instead, the proposed algorithms exclusively rely on the relative poses $\mathbf{p}_j^i$ between neighboring robots computed as explained below. We let $\mathcal{G}^0 = (\mathcal{V}^0, \mathcal{E}^0)$ be the relative poses graph during this stage, and $\mathcal{N}_i^0$ be the set of neighbors of robot i in $\mathcal{G}^0$. Finally, we let $W_{i,j}^0$ be the Metropolis weights defined in Appendix A, (A.3) associated to $\mathcal{G}^0$.

3.1. Computing relative poses

Assume that at some time instant the robots are placed at unknown nearby positions. Each robot $i \in \{1, \dots, n\}$ takes an image of the scene $\mathcal{P}_i$ from its current pose, executes a motion and measures its odometry $\Delta\mathbf{p}_i$. Then, each robot i takes a new image $\mathcal{P}_i'$ from this second pose and exchanges $\mathcal{P}_i'$ with all the robots $j \in \mathcal{N}_i^0$ within its communication range. Robot i uses its images $\mathcal{P}_i, \mathcal{P}_i'$ and its neighbor's image $\mathcal{P}_j'$ to execute the algorithm in [12] and recover the 2D position and the orientation between the three cameras, up to a scale factor. This method operates under the condition that the cameras are nearby, and that a minimal number of landmarks is visible from all the cameras. Then, robot i uses its odometry measurement $\Delta\mathbf{p}_i$ to recover the scale and it finally obtains the relative pose $\mathbf{p}_j^i$ of each of its neighbors $j \in \mathcal{N}_i^0$. If due to the lack of common landmarks, a pair of neighboring robots were unable to retrieve their relative poses, they are considered as non-neighbors and its communication link is deleted from $\mathcal{G}^0$.

3.2. Computing the global reference

Here we present the distributed algorithm executed by the robots to agree on the global reference frame computed as the center of mass G of the robot team. Let us first give an explanation on what this center of mass is. Suppose we are given the true robot poses $\mathbf{p}_i^{\mathtt{w}} = (R_i^{\mathtt{w}}, T_i^{\mathtt{w}})$ in a world reference frame $\mathtt{w}$. Then the position of the center of mass $T_G^{\mathtt{w}}$ in frame $\mathtt{w}$ is

$$T_G^{\mathtt{w}} = \frac{1}{n}\sum_{j=1}^{n} T_j^{\mathtt{w}}. \tag{6}$$

The equivalent expression for the orientation of the global reference frame $R_G^{\mathtt{w}}$ requires further explanations on averages of rotations and it is discussed later in this section. Let $(R_{\mathtt{w}}^G, T_{\mathtt{w}}^G)$ be the inverse of $(R_G^{\mathtt{w}}, T_G^{\mathtt{w}})$, i.e., the pose of the world frame $\mathtt{w}$ expressed in the center of mass frame, $R_{\mathtt{w}}^G = (R_G^{\mathtt{w}})^T$, $T_{\mathtt{w}}^G = -(R_G^{\mathtt{w}})^T T_G^{\mathtt{w}}$. Then, the pose of each robot $i \in \{1, \dots, n\}$ with respect to the center of mass frame $\mathbf{p}_i^G = (R_i^G, T_i^G)$ is

$$R_i^G = R_{\mathtt{w}}^G R_i^{\mathtt{w}}, \qquad\qquad T_i^G = R_{\mathtt{w}}^G T_i^{\mathtt{w}} + T_{\mathtt{w}}^G. \tag{7}$$

The center of mass G representation does not depend on the original reference frame $\mathtt{w}$ selected. The same robot poses relative to the center of mass $\mathbf{p}_i^G$ are obtained regardless of the reference $\mathtt{w}$ used in (6)-(7). Moreover, as we show in this section, the robots do not need to know any world reference frame $\mathtt{w}$ in order to compute their poses $\mathbf{p}_i^G = (R_i^G, T_i^G)$ with respect to G (7). More specifically, we let each robot i compute the pose of the center of mass $\mathbf{p}_G^i = (R_G^i, T_G^i)$ in its own reference frame; and $\mathbf{p}_i^G$ can be obtained from $\mathbf{p}_G^i$ and vice versa as follows:

$$R_i^G = \left(R_G^i\right)^T, \qquad T_i^G = -\left(R_G^i\right)^T T_G^i, \qquad R_G^i = \left(R_i^G\right)^T, \qquad T_G^i = -\left(R_i^G\right)^T T_i^G. \tag{8}$$

We first present the distributed algorithm for the positions and then we explain the agreement on the orientations. Both algorithms exclusively rely on the relative poses $\mathbf{p}_j^i$ between neighboring robots computed in Section 3.1, here denoted by $(R_j^i, T_j^i) \in SE(3)$.

Let each robot $i \in \{1, \dots, n\}$ have an estimate of the position of the center of mass in its own reference frame $T_G^i(t) \in \mathbb{R}^3$ which is initialized as $T_G^i(0) = 0$ and updated each time step $t \in \mathbb{N}$ according to

$$T_G^i(t+1) = \sum_{j \in \mathcal{N}_i^0 \cup \{i\}} W_{i,j}^0 (R_j^i T_G^j(t) + T_j^i), \tag{9}$$

where $W_{i,j}^0$ are the Metropolis weights associated to $\mathcal{G}^0$ as defined in Appendix A, (A.3). Then, we have the following result.

Proposition 3.1. *Assume $\mathcal{G}^0$ is connected. Then, if each robot $i \in \{1, \dots, n\}$ executes the algorithm (9), as $t \to \infty$, each $T_G^i(t)$ with $i \in \{1, \dots, n\}$ tends to the position of the center of mass in robot's i reference T_G^i as defined by eqs. (6)-(8),*

$$\lim_{t \to \infty} T_G^i(t) = T_G^i. \tag{10}$$

Proof. First of all, we show that, by making a proper change of variables, the update rule (9) is actually an averaging algorithm (see Appendix A) and thus the states at the robots asymptotically reach consensus. Consider the change of variables,

$$\mathcal{T}_i(t) = R_i^{\mathtt{w}} T_G^i(t) + T_i^{\mathtt{w}}, \qquad\qquad T_G^i(t) = (R_i^{\mathtt{w}})^T \mathcal{T}_i(t) - (R_i^{\mathtt{w}})^T T_i^{\mathtt{w}}, \tag{11}$$

that expresses each $T_G^i(t)$ in a world frame $\mathtt{w}$, being $(R_i^{\mathtt{w}}, T_i^{\mathtt{w}})$ the robot poses in frame $\mathtt{w}$. We apply this change of variables to our system (9),

$$\mathcal{T}_i(t+1) = \sum_{j \in \mathcal{N}_i^0 \cup \{i\}} W_{i,j}^0 R_i^{\mathtt{w}} R_j^i R_{\mathtt{w}}^j \mathcal{T}_j(t) + \sum_{j \in \mathcal{N}_i^0 \cup \{i\}} W_{i,j}^0 (R_i^{\mathtt{w}} R_j^i T_{\mathtt{w}}^j + R_i^{\mathtt{w}} T_j^i + T_i^{\mathtt{w}}) = \sum_{j \in \mathcal{N}_i^0 \cup \{i\}} W_{i,j}^0 \mathcal{T}_j(t), \tag{12}$$

since $\sum_{j\in\mathcal{N}_i^0\cup\{i\}} W_{i,j}^0 = 1$ for all $i \in \{1,\dots,n\}$, $R_i^{\mathtt{w}} R_j^i R_{\mathtt{w}}^j = R_{\mathtt{w}}^{\mathtt{w}} = I$, and $R_i^{\mathtt{w}} R_j^i T_{\mathtt{w}}^j + R_i^{\mathtt{w}} T_j^i + T_i^{\mathtt{w}} = T_{\mathtt{w}}^{\mathtt{w}} = 0$, for all $i, j \in \{1,\dots,n\}$. Thus, (12) is an averaging algorithm like (A.1), which converges to (A.2) the average of the initial states,

$$\lim_{t\to\infty} \mathcal{T}_i(t) = \frac{1}{n}\sum_{j=1}^n \mathcal{T}_j(0) = \frac{1}{n}\sum_{j=1}^n R_j^{\mathtt{w}} T_G^j(0) + T_j^{\mathtt{w}} = \frac{1}{n}\sum_{j=1}^n T_j^{\mathtt{w}} = T_G^{\mathtt{w}}, \tag{13}$$

which is the position of the center of mass (6) in the world frame $\mathtt{w}$. Let us now reverse the change of variables (11) and use equations (7), (8) and (13),

$$\begin{aligned}\lim_{t\to\infty} T_G^i(t) &= (R_i^{\mathtt{w}})^T T_G^{\mathtt{w}} - (R_i^{\mathtt{w}})^T T_i^{\mathtt{w}} = -(R_i^{\mathtt{w}})^T \left(R_{\mathtt{w}}^G\right)^T T_{\mathtt{w}}^G - (R_i^{\mathtt{w}})^T \left(R_{\mathtt{w}}^G\right)^T R_{\mathtt{w}}^G T_i^{\mathtt{w}} \\ &= -\left(R_i^G\right)^T \left(R_{\mathtt{w}}^G T_i^{\mathtt{w}} + T_{\mathtt{w}}^G\right) = -\left(R_i^G\right)^T T_i^G = T_G^i, \end{aligned} \tag{14}$$

and the proof is complete. □

Let us now discuss the agreement on the orientation of the global reference frame. We use the Karcher mean [30] to compute the average of the robot orientations. Given the robot orientations $R_i^{\mathtt{w}}$ in some frame $\mathtt{w}$, the orientation of the center of mass $R_G^{\mathtt{w}}$ is the Karcher mean given by,

$$R_G^{\mathtt{w}} = \arg\min_{R^{\mathtt{w}}} \sum_{i=1}^n d^2(R^{\mathtt{w}}, R_i^{\mathtt{w}}), \tag{15}$$

where $d^2(R_i^{\mathtt{w}}, R)$ is the Riemannian square distance between $R^{\mathtt{w}}$ and $R_i^{\mathtt{w}}$ given by

$$d^2(R^{\mathtt{w}}, R_i^{\mathtt{w}}) = -\frac{1}{2}\mathrm{Tr}\left\{[\log((R^{\mathtt{w}})^T R_i^{\mathtt{w}})]^2\right\}. \tag{16}$$

Here log is the logarithmic map $\log : SO(3) \to so(3)$ defined by

$$\log(R) = \begin{cases} 0 & \text{if } \beta = 0, \\ \frac{\beta}{2\sin\beta}(R - R^T) & \text{if } \beta \neq 0, \end{cases} \tag{17}$$

where $\beta = \arccos\left(\frac{\mathrm{Tr}\{R\}-1}{2}\right)$. The term $\sum_{i=1}^n d^2(R_i^{\mathtt{w}}, R)$ in (15) can be seen as a cost function to be minimized. Recalling that the initial orientations θ_i are located in a geodesic ball of radius less than $\pi/2$, then this cost function restricted to the geodesic ball is convex, and the Karcher mean is well defined [30].

The orientation of the center of mass $R_G^i(t)$ relative to each robot i is then computed using a distributed consensus algorithm on $SO(3)$ [31] combined with the Metropolis weights as defined in (A.3). Robot i initializes its variable $R_G^i(0) = I$ and updates it at each $t \in \mathbb{N}$ by

$$R_G^i(t+1) = R_G^i(t)\exp(\mathbf{u}_i(t)), \quad \mathbf{u}_i(t) = \sum_{j\in\mathcal{N}_i^0\cup\{i\}} W_{i,j}^0 \log(U_{ij}(t)), \qquad U_{ij}(t) = R_G^i(t)^T R_j^i R_G^j(t), \tag{18}$$

where $W_{i,j}^0$ are the Metropolis weights (A.3) associated to the graph $\mathcal{G}^0$ and exp is the exponential map $\exp : so(3) \to SO(3)$ defined by

$$\exp(\mathbf{u}_i(t)) = \begin{cases} I & \text{if } \alpha = 0, \\ I + \frac{\sin\alpha}{\alpha}\mathbf{u}_i(t) + \frac{1-\cos\alpha}{\alpha^2}\mathbf{u}_i^2(t) & \text{if } \alpha \neq 0, \end{cases} \tag{19}$$

where $\alpha = \sqrt{\frac{1}{2}\mathrm{Tr}\{\mathbf{u}_i^T(t)\mathbf{u}_i(t)\}}$. Given that the robot orientations are planar and within a range of $\pm\frac{\pi}{2}$, and that the graph $\mathcal{G}^0$ is connected, this algorithm converges to the unique Karcher mean expressed in each robot's reference frame

$$\lim_{t\to\infty} R_i^{\mathtt{w}} R_G^i(t) = R_G^{\mathtt{w}}, \tag{20}$$

for all $i \in \{1, \dots, n\}$.

After executing this algorithm for order $n^2 \log(\epsilon^{-1})$ iterations each robot obtains its pose relative to the center of mass with a precision ϵ (see Appendix A). Robots will then express their local maps according to this global reference frame. Therefore, when the robots finish the exploration and decide to merge their maps, all of them will be expressed in the same reference frame.

4. CONSISTENT DATA ASSOCIATION AND FEATURE LABELING

Along their exploration, the robots build their local maps using some SLAM algorithm (the specific strategy used is not relevant for the ensuing discussion). Before merging their local maps, the robots must agree on a data association which is globally consistent. Simultaneously, each robot assigns a label to each of its features in such a way that during the posterior merging process (Section 5), the entries in the information matrix and vector associated to features with the same label can be fused together. Let us begin by introducing the problem of consistent data association in robotic networks.

Let $\mathcal{S}_i = \{f_1^i, \dots, f_{m_i}^i\}$ be the set of m_i features observed by robot $i \in \{1, \dots, n\}$. Each robot i can compute the data association between its own set $\mathcal{S}_i$ and the sets of its neighbors $\mathcal{S}_j$, with $j \in \mathcal{N}_i$. However, it does not know the associations obtained by other robots in the team. Note that, in this context, two features f_r^i, f_s^j are associated when they are prone to be observations taken by different robots i, j, of a common landmark in the environment. Since the associations are not perfect, there may appear chains of local associations relating two or more features from the same robot. This is illustrated in Fig. 1. In general, these inconsistencies cannot be locally detected by the robots. If they start the merging process using only its local associations and there is any inconsistency, at some point a robot will be forced to fuse two or more of its features together. To avoid this situation, we design an algorithm for solving any inconsistent association before merging the maps.

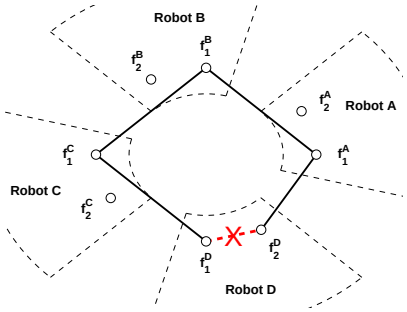


Figure 1: Robots A, B, C and D associate their features comparing their maps in a two-by-two way. Robot A associates its feature f_1^A with f_2^D and with f_1^B; robot B associates f_1^B with f_1^C; robot C associates f_1^C with f_1^D (solid lines). As a result, there is a path (dashed line) between f_1^D and f_2^D. This is an inconsistent association, and all the features connected to f_1^D are conflictive. Finding this path would require the knowledge of the whole association graph.

Let $\mathcal{F}$ be a function that computes the data association between any two sets of features, $\mathcal{S}_i$ and $\mathcal{S}_j$, and returns an association matrix $S_{ij} \in \mathbb{N}^{m_i \times m_j}$ where

$$[S_{ij}]_{r,s} = \begin{cases} 1 & \text{if } f_r^i \text{ and } f_s^j \text{ are associated,} \\ 0 & \text{otherwise,} \end{cases}$$

for $r = 1, \dots, m_i$ and $s = 1, \dots, m_j$. We assume that $\mathcal{F}$ satisfies the following conditions,

(i) When $\mathcal{S}_i = \mathcal{S}_j$, $\mathcal{F}$ returns the identity, $\mathcal{F}(\mathcal{S}_i, \mathcal{S}_i) = S_{ii} = I$;

(ii) The returned association S_{ij} has the property that the features are associated in a one-to-one way,

$$\sum_{r=1}^{m_i} [S_{ij}]_{r,s} \leq 1 \text{ and } \sum_{s=1}^{m_j} [S_{ij}]_{r,s} \leq 1,$$

for all $r = 1, \dots, m_i$ and $s = 1, \dots, m_j$;

(iii) Given two sets $\mathcal{S}_i$ and $\mathcal{S}_j$, $\mathcal{F}(\mathcal{S}_i, \mathcal{S}_j) = S_{ij} = S_{ji}^T = (\mathcal{F}(\mathcal{S}_j, \mathcal{S}_i))^T$.

Let us note that functions satisfying the above properties exist and are abundant in the literature. One example is the JCBB method [15] used in our experiments. The methods in [10, 13, 14, 16] are other examples of functions that also satisfy the requirements above.

We let $\mathcal{G}_{da} = (\mathcal{V}_{da}, \mathcal{E}_{da})$ denote the undirected association graph resulting from applying $\mathcal{F}$ to all pairs of neighboring robots. Each node in $\mathcal{V}_{da}$ is a feature f_r^i, for $i = 1, \dots, n$, $r = 1, \dots, m_i$. There is an edge between two features f_r^i, f_s^j iff $[S_{ij}]_{r,s} = 1$ and $(i, j) \in \mathcal{E}$. If the communication graph $\mathcal{G}$ were complete and $\mathcal{F}$ were providing the ground truth (gt) data association, then $\mathcal{G}_{da}^{gt}$ would exclusively contain disjoint cliques, identifying features observed by multiple robots [20]. Since $\mathcal{F}$ is not perfect, $\mathcal{G}_{da}$ is a perturbed version of $\mathcal{G}_{da}^{gt}$ that includes additional spurious edges while missing others. In addition, since the communication graph $\mathcal{G}$ will not be in general complete, only a subset of the associations are available to the robots. The goal of our algorithm is to detect and resolve inconsistencies in this graph $\mathcal{G}_{da}$ in a decentralized fashion.

Definition 4.1. An *association set* is a set of features that form a connected component in $\mathcal{G}_{da}$. Such set is an *inconsistent association* or a *conflictive set* if there exists a path in $\mathcal{G}_{da}$ between two or more features from the same robot. A feature is *inconsistent* or *conflictive* if it belongs to an inconsistent association.

We denote by m_{sum} the number of features in $\mathcal{G}_{da}$, $m_{sum} = \sum_{i=1}^{n} m_i$ and by $\text{diam}(\mathcal{G}_{da})$ the diameter of $\mathcal{G}_{da}$, the length of the longest path between any two nodes in $\mathcal{G}_{da}$. Note that the diameter satisfies $\text{diam}(\mathcal{G}_{da}) < m_{sum}$. We denote $S \in \mathbb{N}^{m_{sum} \times m_{sum}}$ the adjacency matrix of $\mathcal{G}_{da}$,

$$S = \begin{bmatrix} S_{11} & \dots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{n1} & \dots & S_{nn} \end{bmatrix} \qquad \text{where} \quad S_{ij} = \begin{cases} \mathcal{F}(\mathcal{S}_i, \mathcal{S}_j) & \text{if } j \in \mathcal{N}_i \cup \{i\}, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{21}$$

In what follows we present the data association and labeling algorithm which erases links from $\mathcal{G}_{da}$ to break any inconsistent association. Note that the adjacency matrix is unweighted and the algorithm does not manage any information of the link quality.

Simultaneously to the data association process, the robots assign labels to their features. After checking feature f_r^i is consistent, robot i assigns it a label $L_r^i = (i_\star, r_\star) \in \mathbb{N}^2$ composed of a robot identifier $i_\star$ and a feature index $r_\star$ as follows. Assume f_r^i and features $f_s^j, f_{s'}^{j'}, \dots$ form a consistent association set in $\mathcal{G}_{da}$ and thus they are observations of a common landmark in the environment taken by robots $i, j, j', \dots$. Among all the candidates $(i, r), (j, s), (j', s'), \dots$, a unique label $(i_\star, r_\star)$ is selected by the robots, e.g., the one with the lowest robot id. Then, robot i assigns this label to f_r^i, $L_r^i = (i_\star, r_\star)$; the other robots $j, j', \dots$, proceed in a similar way so that finally,

$$L_r^i = L_s^j = L_{s'}^{j'} = \dots = (i_\star, r_\star)\,.$$

We say a feature f_r^i is *exclusive* if it is isolated in $\mathcal{G}_{da}$, corresponding to a landmark observed by a single robot i; in this case, its label L_r^i is simply (i, r). Otherwise, we say f_r^i is non-exclusive and it may either be *consistent* or conflictive. Consistent features are labeled as explained above, whereas robots wait until conflicts are *resolved* for labeling its conflictive features. The data association and labeling process finishes with an association graph $\mathcal{G}_{da}$ free of any inconsistent association and with all the features labeled. When the algorithm finishes, two features f_r^i, f_s^j have the same label, $L_r^i = L_s^j$, iff they are connected by a path in the resulting conflict-free $\mathcal{G}_{da}$. The decentralized data association and labeling algorithm is summarized in Algorithm 4.1. This strategy makes use of two subroutines to detect features and resolve inconsistencies that we explain in what follows.

Throughout this section, we use $\tilde{\mathcal{S}}_i \subseteq \mathcal{S}_i$ for the set of unlabeled features at robot $i \in \{1, \dots, n\}$ and let $|\tilde{\mathcal{S}}_i|$ be its cardinality, i.e., the number of unlabeled features at robot i. The set of labels $\mathcal{L}_i$ consists of the labels L_r^i already assigned to the features $f_r^i \in \mathcal{S}_i \setminus \tilde{\mathcal{S}}_i$. Given a matrix A_{ij} of size $|\tilde{\mathcal{S}}_i| \times |\tilde{\mathcal{S}}_j|$, we define the function $\bar{r} = \text{row}\left(f_r^i\right)$ that takes an unlabeled feature $f_r^i \in \tilde{\mathcal{S}}_i$ and returns its associated row in A_{ij}, with $\bar{r} \in \{1, \dots, |\tilde{\mathcal{S}}_i|\}$. Equivalently, we define the function $\bar{s} = \text{col}(f_s^j)$ for features in $\tilde{\mathcal{S}}_j$.

Algorithm 4.1 Data association and labeling - Robot i

$\tilde{\mathcal{S}}_i \leftarrow \{f_1^i, \dots, f_{m_i}^i\}$, $\mathcal{L}_i \leftarrow \emptyset$
Solve the local data association
ASSIGN_LABEL($L_r^i = (i, r)$, f_r^i) to each *exclusive* feature f_r^i
while $|\tilde{\mathcal{S}}_i| > 0$ **do**
    Run the detection algorithm
    Find each *consistent* feature f_r^i and its root $f_{r_\star}^{i_\star}$
    ASSIGN_LABEL($L_r^i = (i_\star, r_\star)$, f_r^i)
    Run the resolution algorithm
    Find each *resolved* feature f_r^i and its component id $[i_\star, r_\star]$
    ASSIGN_LABEL($L_r^i = (i_\star, r_\star)$, f_r^i)
    Find each *exclusive* feature f_r^i
    ASSIGN_LABEL($L_r^i = (i, r)$, f_r^i)
end while
function ASSIGN_LABEL(L_r^i, f_r^i)
    $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \{L_r^i\}$, $\tilde{\mathcal{S}}_i \leftarrow \tilde{\mathcal{S}}_i \setminus \{f_r^i\}$
end function

4.1. Local data association and detection of exclusive features

Initially, all the features of each robot i are unlabeled,

$$\tilde{\mathcal{S}}_i = \{f_1^i, \dots, f_{m_i}^i\}, \qquad \mathcal{L}_i = \emptyset.$$

Each robot i solves a local data association with each of its neighbors $j \in \mathcal{N}_i$ and obtains the association matrix $S_{ij} \in \mathbb{N}^{m_i \times m_j}$. Then, the robot locally detects its *exclusive* features f_r^i which have not been associated to any other feature,

$$[S_{ij}]_{r,s} = 0 \qquad \text{for all } j \in \mathcal{N}_i,\ j \neq i, \text{ and all } s \in \{1, \dots, m_j\}. \tag{22}$$

Since an exclusive feature f_r^i is always consistent, robot i assigns a label L_r^i to it composed of its own robot id and feature index and removes it from the set of unlabeled features,

$$L_r^i = (i, r), \qquad \mathcal{L}_i = \mathcal{L}_i \cup L_r^i, \qquad \tilde{\mathcal{S}}_i = \tilde{\mathcal{S}}_i \setminus \{f_r^i\}. \tag{23}$$

Since its unlabeled features in $\tilde{\mathcal{S}}_i$ may be conflictive, it executes the detection algorithm on this subset.

4.2. Detection algorithm

Following Definition 4.1, the detection of inconsistent associations requires the computation of paths among the vertices of $\mathcal{G}_{da}$. This information can be extracted from the powers of the adjacency matrix. The (r, s) entry of the t^{th} power of S, $[S^t]_{r,s}$, equals the number of paths of length t (including paths with self loops) from node r to node s ([32, Lemma 1.32]). This can be done in a decentralized fashion using the following detection algorithm which relies on computing the powers of the adjacency matrix S^t of $\mathcal{G}_{da}$. Note that the algorithm is executed exclusively on the unlabeled features $\tilde{\mathcal{S}}_1, \dots, \tilde{\mathcal{S}}_n$. From now on, we let $\tilde{\mathcal{G}}_{da}$ be the subgraph associated to the unlabeled features with adjacency matrix $\tilde{S}$. Equivalently, we let $\tilde{S}_{ij} \in \mathbb{N}^{|\tilde{\mathcal{S}}_i| \times |\tilde{\mathcal{S}}_j|}$ be the elements associated to unlabeled features in the block S_{ij}.

Let each robot $i \in \{1, \dots, n\}$ maintain the blocks within $\tilde{S}^t$ associated to its own features, $X_{ij}(t) \in \mathbb{N}^{|\tilde{\mathcal{S}}_i| \times |\tilde{\mathcal{S}}_j|}$, for $j \in \{1, \dots, n\}$ and $t \geq 0$, which are initialized as

$$X_{ij}(0) = \begin{cases} I, & j = i, \\ \mathbf{0}, & j \neq i, \end{cases} \tag{24}$$

and updated, at each time step, via the following algorithm

$$X_{ij}(t+1) = \sum_{j' \in N_i \cup \{i\}} \tilde{S}_{ij'} X_{j'j}(t), \tag{25}$$

where $\tilde{S}_{ij'}$ is the association matrix between the unlabeled features at robots i and j'. Observe that the algorithm is fully distributed because the nodes only use information about its direct neighbors in the communication graph. Let $[\tilde{S}^t]_{ij} \in \mathbb{N}^{|\tilde{\mathcal{S}}_i| \times |\tilde{\mathcal{S}}_j|}$ be the block within $\tilde{S}^t$ related to the associations between robot i and robot j. The matrices $X_{ij}(t)$ computed by each robot i using the decentralized algorithm (25) are exactly the sub-matrices $[\tilde{S}^t]_{ij}$,

$$X_{ij}(t) = [\tilde{S}^t]_{ij}, \tag{26}$$

for all $i, j \in \{1, \dots, n\}$ and all $t \in \mathbb{N}$ [29, Theorem 3.2].

The detection algorithm finishes after $t_\star < \sum_{i=1}^{n} |\tilde{\mathcal{S}}_i|$ iterations, being $t_\star$ the first t for which for all $i, j \in \{1, \dots, n\}$, and all $\bar{r} \in \{1, \dots, |\tilde{\mathcal{S}}_i|\}$, $\bar{s} \in \{1, \dots, |\tilde{\mathcal{S}}_j|\}$,

$$[X_{ij}(t+1)]_{\bar{r},\bar{s}} = 0 \;\text{ iff }\; [X_{ij}(t)]_{\bar{r},\bar{s}} = 0.$$

When the algorithm finishes, robot i has the power matrices $X_{ij} \in \mathbb{N}^{|\tilde{\mathcal{S}}_i| \times |\tilde{\mathcal{S}}_j|}$, for $j \in \{1, \dots, n\}$, which contain the entries in $S^{\text{diam}(\mathcal{G}_{da})}$ associated to the features in $\tilde{\mathcal{S}}_i$ and $\tilde{\mathcal{S}}_j$. There is a path between f_r^i and f_s^j in $\mathcal{G}_{da}$ iff

$$[X_{ij}]_{\bar{r},\bar{s}} > 0, \tag{27}$$

being $\bar{r} = \text{row}(f_r^i)$ and $\bar{s} = \text{col}(f_s^j)$. These matrices give robot i the information about all the association paths of its features and the features of the rest of the robots in the network.

Then, each robot i detect its *consistent* features using two simple rules. An unlabeled feature $f_r^i \in \tilde{\mathcal{S}}_i$, with row index $\bar{r} = \text{row}(f_r^i)$, is consistent if and only if the following conditions are satisfied:

(i) There exists no path between f_r^i and any other feature of robot i. For all $\bar{r}' \in \{1, \dots, |\tilde{\mathcal{S}}_i|\} \setminus \{\bar{r}\}$,

$$[X_{ii}]_{\bar{r},\bar{r}'} = 0; \tag{28}$$

(ii) If f_r^i is connected with a feature f_s^j from a robot j, it cannot be connected with a second feature $f_{s'}^j$ from the same robot. For all $j \in \{1, \dots, n\} \setminus \{i\}$,

$$\text{if } [X_{ij}]_{\bar{r},\bar{s}} > 0 \text{ then } [X_{ij}]_{\bar{r},\bar{s}'} = 0, \tag{29}$$

for all $\bar{s}' \neq \bar{s}$, with $\bar{s}, \bar{s}' \in \{1, \dots, |\tilde{\mathcal{S}}_j|\}$.

After a feature f_r^i has been classified as consistent, its robot i proceeds to assign it a label. Here we show how robot i decides the feature label $(i_\star, r_\star)$. Let us first give a general definition of the root robot of an either consistent or conflictive association set.

Definition 4.2. The *root* robot $i_\star$ for an association set is the one that has the most features in it. In case there are multiple candidates, it is the one with the lowest identifier. Equivalently, we define the *root* features $f_{r_\star}^{i_\star}, f_{r'_\star}^{i_\star}, \dots$ as the features from the root robot that belong to the association set.

Using the power matrices $X_{i1}, \dots, X_{in}$, robot i can find the number of features $\tilde{m}_j$ from a second robot j that belong to the same association set than f_r^i with $\bar{r} = \text{row}(f_r^i)$ as follows,

$$\tilde{m}_j = \left| \left\{ f_s^j \mid [X_{ij}]_{\bar{r},\bar{s}} > 0, \text{ with } \bar{s} = \text{col}(f_s^j) \right\} \right|. \tag{30}$$

If we let $\tilde{m}_\star$ be the maximum $\tilde{m}_j$ for $j \in \{1, \dots, n\}$, then the root robot $i_\star$ and root features $f_{r_\star}^{i_\star}, f_{r'_\star}^{i_\star}, \dots$ for the association set of f_r^i with $\bar{r} = \text{row}(f_r^i)$ are

$$i_\star = \min \{j \mid \tilde{m}_j = \tilde{m}_\star\}, \qquad \{r_\star, r'_\star, \dots\} = \left\{s \mid [X_{ii_\star}]_{\bar{r},\bar{s}} > 0 \text{ with } \bar{s} = \text{col}(f_s^{i_\star})\right\}. \tag{31}$$

When f_r^i belongs to a consistent set, the root $i_\star$ corresponds to the robot with a single feature $f_{r_\star}^{i_\star}$ in the association set that has the lowest identifier,

$$i_\star = \min \left\{j \mid [X_{ij}]_{\bar{r},\bar{s}} > 0 \text{ for some } \bar{s} \in \{1, \dots, |\tilde{\mathcal{S}}_j|\}\right\}, \quad r_\star = \left\{s \mid [X_{ii_\star}]_{\bar{r},\bar{s}} > 0 \text{ with } \bar{s} = \text{col}(f_s^{i_\star})\right\}, \tag{32}$$

where $\bar{r} = \text{row}(f_r^i)$. Robot i assigns to its feature f_r^i the label $L_r^i = (i_\star, r_\star)$ and removes it from the set of unlabeled features,

$$L_r^i = (i_\star, r_\star), \qquad \mathcal{L}_i = \mathcal{L}_i \cup L_r^i, \qquad \tilde{\mathcal{S}}_i = \tilde{\mathcal{S}}_i \setminus \{f_r^i\}. \tag{33}$$

Thus, all features in the association set are assigned the same label. The robots proceed with all its consistent features in a similar fashion. For the features classified as conflictive, the following resolution algorithm is executed to solve the inconsistencies.

4.3. Resolution algorithm

The resolution of inconsistent features deletes edges from $\mathcal{G}_{da}$ to break the paths between the conflictive features. Let $\tilde{m}_j$ be the number of features of a robot j in the inconsistency as in (31) and $\tilde{m}_\star$ be the maximum $\tilde{m}_j$ for $j \in \{1, \dots, n\}$. Note that the root robot $i_\star$ has exactly $\tilde{m}_\star$ features in the inconsistency. It is clear that the number of conflict-free components in which a conflictive set can be decomposed is lower bounded by $m_\star$ since each root feature $f_{r_\star}^{i_\star}$ must belong to a different component in order for them to be conflict-free. The resolution algorithm presented in this section constructs such conflict-free components by initially assigning each root feature to a component and incrementally adding more features to each of them as follows. Each component has associated a component id that is used later for labeling its associated features.

Initially, each robot i analyzes its power matrices X_{ij} to find out if it is the root robot for some of the conflictive sets as in eq. (31). Then, each conflictive set is managed in parallel following the process shown below. Let us consider one of the conflictive sets with root robot $i_\star$. This robot initiates the resolution process by creating a new component for each of its root features $f_{r_\star}^{i_\star}$ with component id $(i_\star, r_\star)$. Robot $i_\star$ assigns $f_{r_\star}^{i_\star}$ to this component and tries to add to each component the features directly associated to $f_{r_\star}^{i_\star}$. Whenever a robot successfully adds one of its features to a component, it also propagates the process to any feature directly associated to it.

Let us consider the case when f_s^j has been assigned to a component $(i_\star, r_\star)$. For all f_r^i such that $[\tilde{S}_{ji}]_{\bar{s},\bar{r}} = 1$, with $\bar{s} = \text{row}(f_s^j)$ and $\bar{r} = \text{col}(f_r^i)$, robot j sends a component request message to robot i. When robot i receives it, the following may happen

(a) f_r^i is already assigned to $(i_\star, r_\star)$;

(b) f_r^i is assigned to a different component;

(c) other feature $f_{r'}^i$ of robot i is already assigned to $(i_\star, r_\star)$;

(d) f_r^i is unassigned and no other $f_{r'}^i$ is assigned to $(i_\star, r_\star)$.

In case (a), f_r^i already belongs to the component and robot i does nothing. In cases (b) and (c), f_r^i cannot be added to the component and robot i deletes the association between f_s^j and f_r^i,

$$[S_{ij}]_{r,s} = 0, \tag{34}$$

and replies with a reject message to robot j; when j receives the reject message, it deletes the equivalent edge $[S_{ji}]_{s,r}$. In case (d), robot i assigns its feature f_r^i to the component $(i_\star, r_\star)$ and the process is repeated.

After n iterations the resolution algorithm finishes, with the size of each conflictive set being reduced by at least $2\tilde{m}_\star$ features [29, Theorem 4.3]. When the algorithm finishes, each feature f_r^i that has been assigned to a component $(i_\star, r_\star)$ has become consistent due to the edge removals (34). We say that such features are *resolved*. Thus, all the resolved features with the same component id form a consistent association set. Each robot i uses the component id of f_r^i as its label,

$$L_r^i = (i_\star, r_\star), \qquad \mathcal{L}_i = \mathcal{L}_i \cup L_r^i, \qquad \tilde{\mathcal{S}}_i = \tilde{\mathcal{S}}_i \setminus \{f_r^i\}. \tag{35}$$

Additionally, due to edge removal, some unlabeled features $f_r^i \in \tilde{\mathcal{S}}_i$ may have become exclusive. Robot i detects such features f_r^i by checking that

$$[\tilde{S}_{ij}]_{\bar{r},\bar{s}} = 0, \qquad \text{for all } j \in \mathcal{N}_i,\ j \neq i,\ \text{all } \bar{s} \in \{1, \ldots, |\tilde{\mathcal{S}}_j|\}, \tag{36}$$

being $\bar{r} = \text{row}(f_r^i)$, and it manages them as in (23). The remaining features may still be conflictive. Each robot i executes a new detection-resolution iteration on these still unlabeled features $\tilde{\mathcal{S}}_i$.

In a finite number of iterations, all features of all robots have been labeled, and the algorithm finishes. The resulting association graph $\mathcal{G}_{da}$ is free of any conflictive set. The interest of the presented algorithm is that it is fully decentralized and works on local information. The decisions and actions taken by each robot are based on its local data. Each robot i uses its own S_{ij} and X_{ij} to classify its features and establish their labels, and it is responsible for deleting its own local edges from S_{ij}. For further details on the detection and resolution algorithm, see [29].

5. CONSENSUS ON THE GLOBAL MAP

Once the local maps have been expressed in a common reference and the data association has been solved, we address here the map fusion problem. There is a rich literature in sensor networks, where the observations taken by a set of sensors are fused in IF form to build a better estimate of a variable. A widely used distributed sensor fusion method whose convergence conditions and properties have been deeply studied can be found in Appendix B. A brief summary of its characteristics follows. The information matrix $\hat{I}_G^i(t)$ and vector $\hat{\mathbf{i}}_G^i(t)$ estimated by each node $i \in \{1, \ldots, n\}$ executing this sensor fusion algorithm in Appendix B and the associated mean $\hat{\mathbf{x}}_G^i(t)$ and covariance $\hat{\mathbf{x}}_G^i(t)$ have the following properties:

(i) they asymptotically converge to the following values,

$$\lim_{t\to\infty} \hat{I}_G^i(t) = I_G \ / \ n, \qquad \lim_{t\to\infty} \hat{\mathbf{i}}_G^i(t) = \mathbf{i}_G \ / \ n, \qquad \lim_{t\to\infty} \hat{\mathbf{x}}_G^i(t) = \hat{\mathbf{x}}_G, \qquad \lim_{t\to\infty} \hat{\Sigma}_G^i(t) = n\ \Sigma_G,$$

being $I_G^i, \mathbf{i}_G^i, \hat{\mathbf{x}}_G, \Sigma_G$ the information matrix, information vector, mean and covariance of the global map respectively;

(ii) the mean is an unbiased estimate of the truth $\mathbf{x}$ and $\mathbf{E}\left[\hat{\mathbf{x}}_G^i(t)\right] = \mathbf{x}$ for all $t \geq 0$;

(iii) the temporal estimates are consistent for all $t \geq 0$ since the true uncertainty $Q_G^i(t)$ is smaller than the estimated uncertainty, $Q_G^i(t) \preceq \hat{\Sigma}_G^i(t)$.

The use of sensor fusion algorithms for merging stochastic maps has an important difficulty that must be addressed. Sensors usually observe a set of variables which are a priori known, e.g., temperature, humidity, etc. Specifically, during the initialization of the algorithm (Appendix B, (B.1)) each node i knows the observation matrix H_i relating its measurements with the variables to be estimated. In a map merging scenario, this matrix H_i relates the local features observed by robot i with the globally observed ones. However, in our case the robots do not know the observation matrices since they do not know the whole set of features that have been observed by the robot team. Initially each robot i exclusively knows the labels $\mathcal{L}_i$ of its local features. The goal is that each robot discovers the feature labeling of all the robots through the interaction with its neighbors. We propose an algorithm where the feature labeling discovering and the data fusion are executed simultaneously.

After solving the data association (Section 4), each robot $i \in \mathcal{V}$ has its label set $\mathcal{L}_i = \{L_1^i, \dots, L_{m_i}^i\}$ with the labels of its m_i features. Two features f_r^i and f_s^j from robots i, j can be fused together if and only if their labels $L_r^i = (i_\star, r_\star)$, $L_s^j = (i'_\star, r'_\star)$ have the same value,

$$i_\star = i'_\star \quad \text{and} \quad r_\star = r'_\star.$$

However, in the initialization stage in Appendix B, (B.1), the robots do not know the label sets $\mathcal{L}_j$ from the other robots, and thus cannot compute their matrix H_i.

Throughout the discussion, we use a sorted version of the label sets $\mathcal{L}_i$ that we term label vectors L_i. In a label vector L_i, the labels $L_r^i = (i_\star, r_\star)$ appear sorted following the lexicographic order (first by $i_\star$, then by $r_\star$). We assume that initially each robot i sorts its set $\mathcal{L}_i$ to create its label vector L_i, and that it arranges its local map accordingly. The robot poses which have been estimated are always placed together at the first rows and columns of the information matrices and vectors. In this case, the labels are exclusively composed of the robot id. Let $|L_i|$ denote the number of labels in L_i. Given two label vectors L_i, L_j, we say that $L_i \subseteq L_j$ if all labels of L_i are contained in L_j. We let Γ be a function that joins two label vectors L_i, L_j and returns a new one L_{ij} with all the labels in L_i and L_j, without duplicates, and sorted as prescribed above. It is not difficult to see that the function Γ satisfies the following properties:

(i) $\Gamma(L_i, L_j) = \Gamma(L_j, L_i)$;

(ii) $\Gamma(L_i, L_i) = L_i$;

(iii) $\Gamma(L_i, \Gamma(L_j, L_{j'})) = \Gamma(\Gamma(L_i, L_j), L_{j'})$;

(iv) $L_i, L_j \subseteq \Gamma(L_i, L_j)$;

for any label vectors $L_i, L_j, L_{j'}$.

We let $\mathcal{H}$ be a function that computes the observation matrix between two label vectors L_i, L_j with $L_i \subseteq L_j$. $\mathcal{H}(L_i, L_j)$ returns a matrix $H_j^i \in \{0,1\}^{|L_i|\ \mathtt{szf} \times |L_j|\ \mathtt{szf}}$ that relates the Cartesian coordinates of the features with labels L_i and the ones with labels L_j. It can be seen that $\mathcal{H}$ has the property that, if $L_i \subseteq L_j \subseteq L_{j'}$, $\mathcal{H}(L_i, L_j) = H_j^i$, and $\mathcal{H}(L_j, L_{j'}) = H_{j'}^j$, then $H_j^i H_{j'}^j = H_{j'}^i = \mathcal{H}(L_i, L_{j'})$. Additionally, $\mathcal{H}$ also satisfies $\mathcal{H}(L_i, L_i) = I$.

Given the labels of all the robots L_j, for $j = 1, \dots, n$, we let L_G be the global label vector which contains all the m *different* labels in $L_1, \dots, L_n$,

$$L_G = \Gamma(\Gamma(\dots \Gamma(L_1, L_2), \dots), L_n). \tag{37}$$

Note that the same L_G is obtained if the functions Γ are applied to the label vectors L_i in any other order, as long as each L_i, $i \in \{1, \dots, n\}$, is used at least once. Each observation matrix H_i in (B.1) is then $\mathcal{H}(L_i, L_G)$.

We let each robot i start with its own L_i and, incrementally, incorporate the new labels discovered in its neighbors data, to finally discover the full L_G. Each robot $i \in \{1, \dots, n\}$ maintains a label vector $L_i(t)$ with the different labels discovered by itself up to time t. Additionally, it maintains the variables $\mathbf{i}_G^i(t) \in \mathbb{R}^{\mathcal{M}_i(t)}$ and $I_G^i(t) \in \mathbb{R}^{\mathcal{M}_i(t) \times \mathcal{M}_i(t)}$, where $\mathcal{M}_i(t)$ is the size of its estimated global map at time t, $\mathcal{M}_i(t) = |L_i(t)|\ \mathtt{szf}$. At $t = 0$, robot i initializes these variables with its own local information,

$$L_i(0) = L_i, \qquad I_G^i(0) = \Sigma_i^{-1}, \qquad \mathbf{i}_G^i(0) = \Sigma_i^{-1} \hat{\mathbf{x}}_i. \tag{38}$$

At each iteration t, the robot incorporates into its label vector $L_i(t+1)$ the new labels discovered from its neighbors data $L_j(t)$, with $j \in \mathcal{N}_i \cup \{i\}$,

$$L_i(t+1) = \Gamma(\Gamma(\dots \Gamma(L_{j_1}(t), L_{j_2}(t)), \dots), L_{j_{\mathcal{N}_i+1}}(t)), \tag{39}$$

where $\{j_1, \dots, j_{\mathcal{N}_i+1}\} = \mathcal{N}_i \cup \{i\}$. Then, it arranges the variables $I_G^i(t)$ and $\mathbf{i}_G^i(t)$ accordingly, and computes the new values,

$$I_G^i(t+1) = \sum_{j=1}^{n} W_{i,j} (H_{i,t+1}^{j,t})^T I_G^j(t) H_{i,t+1}^{j,t}, \qquad \mathbf{i}_G^i(t+1) = \sum_{j=1}^{n} W_{i,j} (H_{i,t+1}^{j,t})^T \mathbf{i}_G^j(t), \tag{40}$$

where $H_{i,t+1}^{j,t} = \mathcal{H}(L_j(t), L_i(t+1))$.

Proposition 5.1. *The outcomes of the algorithms* (38)-(40) *and* (B.2) *are related as follows: for all* $t \geq 0$ *and all* $i \in \{1, \dots, n\}$,

$$\hat{I}^i_G(t) = (H^{i,t}_G)^T I^i_G(t) H^{i,t}_G, \qquad \hat{\mathbf{i}}^i_G(t) = (H^{i,t}_G)^T \mathbf{i}^i_G(t), \tag{41}$$

where $H^{i,t}_G = \mathcal{H}(L_i(t), L_G)$. *Furthermore, after* $\text{diam}(\mathcal{G})$ *iterations, the result of both algorithms is exactly the same, i.e., for all* $t \geq 0$ *and all* $i \in \{1, \dots, n\}$,

$$\hat{I}^i_G(\text{diag}(\mathcal{G}) + t) = I^i_G(\text{diag}(\mathcal{G}) + t), \qquad \hat{\mathbf{i}}^i_G(\text{diag}(\mathcal{G}) + t) = \mathbf{i}^i_G(\text{diag}(\mathcal{G}) + t). \tag{42}$$

Proof. We only present the proof for the information matrices $I^i_G(t)$ (the reasoning is analogous for the information vectors $\mathbf{i}^i_G(t)$). We reason by induction. At $t = 0$, equation (41) is satisfied since

$$\hat{I}^i_G(0) = H_i^T \Sigma_i^{-1} H_i = (H^{i,0}_G)^T I^i_G(0) H^{i,0}_G, \tag{43}$$

for all $i \in \{1, \dots, n\}$. Assuming that (41) is true for a given t, then for all $i \in \{1, \dots, n\}$,

$$\begin{aligned} \hat{I}^i_G(t+1) &= \sum_{j=1}^{n} W_{i,j} \hat{I}^j_G(t) = \sum_{j=1}^{n} W_{i,j} (H^{j,t}_G)^T I^j_G(t) H^{j,t}_G = (H^{i,t+1}_G)^T \left[\sum_{j=1}^{n} (H^{j,t}_{i,t+1})^T I^j_G(t) H^{j,t}_{i,t+1} \right] H^{i,t+1}_G \\ &= (H^{i,t+1}_G)^T I^i_G(t+1) H^{i,t+1}_G, \end{aligned} \tag{44}$$

where we have used the facts that for all $t \geq 0$ and all $i, j \in \{1, \dots, n\}$, $H^{j,t}_G = H^{j,t}_{i,t+1} H^{i,t+1}_G$ and $\sum_{j=1}^{n} W_{i,j} = 1$. This concludes the proof of (41). Regarding (42), note that after $\text{diam}(\mathcal{G})$ iterations of the algorithm (38)-(40), each robot $i \in \{1, \dots, n\}$ has already incorporated into its label vector information from all the initial label vectors L_j, for $j = 1, \dots, n$. Therefore, for all $t \geq 0$, $L_i(\text{diam}(\mathcal{G}) + t) = L_G$, where L_G is the global vector in (37). Then, $H^{i,\text{diam}(\mathcal{G})+t}_G = \mathcal{H}(L_G, L_G) = I$, and the result follows. □

Using the previous algorithm, the robots compute the same global map than if they were given H_i from the beginning, with the additional benefit that the information matrices $I^i_G(t)$ can be inverted for any $t \geq 0$ and $i \in \{1, \dots, n\}$, since they do not contain any non informative zero columns or rows. Note that in the basic map merging algorithm (Appendix B), the information matrices $\hat{I}^i_G(t)$ are initialized with zero rows and columns for the features which are not local to robot i. Therefore, they cannot be inverted until robot i has received informative (non zero) data for all the features. In case all the robots have observed at least one exclusive feature, the robots have to wait for $t = \text{diam}(\mathcal{G})$ iterations for inverting their information matrices and recovering a temporal estimate of the global map. However, when the robots use the algorithm (38)-(40), their information matrices $I^i_G(t)$ can be inverted for any $t \geq 0$ and $i \in \{1, \dots, n\}$ and thus the global map can be computed at any step. In the next sections, we will analyze the presented algorithm in terms of its theoretical and experimental performance. From now on, we let $\mathbf{x}^i_G(t) \in \mathbb{R}^{\mathcal{M}_i(t)}$, $\Sigma^i_G(t) \in \mathbb{R}^{\mathcal{M}_i(t) \times \mathcal{M}_i(t)}$ be the global map estimated by a robot i executing algorithm (38)-(40),

$$\mathbf{x}^i_G(t) = \left(I^i_G(t)\right)^{-1} \mathbf{i}^i_G(t), \qquad \Sigma^i_G(t) = \left(I^i_G(t)\right)^{-1}. \tag{45}$$

6. COMPLEXITY ANALYSIS

In the previous sections, we have presented the parts that compose the distributed map merging algorithm. Here, we analyze the algorithm complexity regarding execution time, amount of communication, and memory space required. Let $\mathcal{M}_{\max}$ be the highest size of the local map of any robot,

$$\mathcal{M}_{\max} = \max_{i \in \{1, \dots, n\}} \mathcal{M}_i,$$

and $d_{\max}$ be the highest number of neighbors of any robot,

$$d_{\max} = \max_{i \in \{1,\dots,n\}} |\mathcal{N}_i|.$$

For time-varying topologies $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, $d_{\max}$ is defined as

$$d_{\max} = \max_{i \in \{1,\dots,n\}, t \geq 0} |\mathcal{N}_i(t)|.$$

Computational complexity per iteration and robot

The consensus initialization operations in the map merging algorithm (Section 5) have a computational complexity per iteration and robot of $O(\mathcal{M}_{\max}^3)$ that exclusively depends on the local map size. During the general iterations, the cost is $O(d_{\max}\mathcal{M}_G^2)$ to perform the matrix additions.

Communication complexity per iteration and robot

Along the consensus iterations, each robot i exchanges its information matrix $I_G^i(t)$ with its neighbors, with a communication cost of $O(d_{\max}\mathcal{M}_G^2)$. Note that each $I_G^i(t)$s is a sparse information matrix, where the significant coefficients are grouped around the main diagonal. Therefore, if a compression algorithm is used, the cost of exchanging $I_G^i(t)$ can be expressed as $O(n\mathcal{M}_{\max}^2)$. Alternatively, it can be expressed as $O(n+m)$ if we consider the local map sizes as constants, giving rise to a total communication cost per robot of, respectively, $O(d_{\max}n\mathcal{M}_{\max}^2)$ or $O(d_{\max}(n+m))$. In addition, the robots exchange the label vectors of the features, with a total cost of $O(d_{\max}m)$.

Space complexity per iteration and robot

Along the consensus algorithm, the maximal space complexity for each robot is associated to the matrix $I_G^i(t)$ that, as discussed above, can be considered $O(n\mathcal{M}_{\max}^2)$ or $O(n+m)$. Additionally this algorithm requires extra storage for the label vectors, although it does not have influence in the worst-case space complexity measure.

Time complexity until completion

As we mentioned before, the consensus is asymptotically reached, which means that the time until completion is infinite. However, the convergence speed of the averaging algorithm presents a geometric rate for fixed graphs [33], [34] which depends on the second eigenvalue with the largest absolute value $|\lambda_2(W)|$ in the Metropolis weight matrix (A.3). If we denote $\gamma = |\lambda_2(W)|$, it can be shown that each entry $[I_G^i(t)]_{r,s}$, $[\mathbf{i}_G^i(t)]_r$ in the information matrices and vectors estimated by the robots evolve according to

$$\begin{aligned} |[I_G^i(t)]_{r,s} - [I_G]_{r,s}| &\leq (\gamma)^t \sqrt{n} \max_j \left\{ \left| [I_G^j(0)]_{r,s} - [I_G]_{r,s} \right| \right\}, \\ |[\mathbf{i}_G^i(t)]_r - [\mathbf{i}_G]_r| &\leq (\gamma)^t \sqrt{n} \max_j \left\{ \left| [\mathbf{i}_G^j(0)]_r - [\mathbf{i}_G]_r \right| \right\}, \end{aligned} \tag{46}$$

for all $i \in \{1,\dots,n\}$, all $r,s \in \{1,\dots,\mathcal{M}_i(t)\}$, and all $t \geq 0$.

For graphs with switching topology $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, the convergence speed is geometric if the graph has an interval of joint connectivity τ such that every subsequence

$$\{G(t_0+1), \dots\ G(t_0+\tau)\}$$

of length τ is jointly connected for all t_0 [34]. In these graphs, the τ-index of joint contractivity $\delta < 1$ is given by

$$\delta = \max_{\mathcal{W} \in \mathbb{W}_\tau} \{|\lambda_2(\mathcal{W})| \mid \mathcal{W} \text{ primitive paracontractive}\} \tag{47}$$

where $\mathbb{W}_\tau$ is the set of all products of, at most, τ Metropolis matrices $W(t)$ that can be obtained in the communication graph. The convergence speed of each entry $[I_G^i(t)]_{r,s}$, $[\mathbf{i}_G^i(t)]_r$ in the information matrices and vectors estimated by the robots depends on the τ-index of joint contractivity,

$$|[I_G^i(t)]_{r,s} - [I_G]_{r,s}| \leq (\delta)^{\llcorner \frac{t}{\tau} \lrcorner} \sqrt{n} \max_j \left\{ \left| [I_G^j(0)]_{r,s} - [I_G]_{r,s} \right| \right\},$$
$$|[\mathbf{i}_G^i(t)]_r - [\mathbf{i}_G]_r| \leq (\delta)^{\llcorner \frac{t}{\tau} \lrcorner} \sqrt{n} \max_j \left\{ \left| [\mathbf{i}_G^j(0)]_r - [\mathbf{i}_G]_r \right| \right\}, \tag{48}$$

where $\llcorner \frac{t}{\tau} \lrcorner$ is the largest integer less than or equal to $\frac{t}{\tau}$.

Therefore, the convergence speed depends on the topology of the communication graph. For a complete graph the convergence is reached in one iteration, whereas for networks with lower connectivity like string and circular graphs it is necessary to execute order $n^2 \log(\epsilon^{-1})$ iterations for achieving a precision (see Appendix A).

7. EXPERIMENTS

7.1. Simulations

In order to show the performance of the algorithm, we have carried out several simulations with a team composed by 7 robots exploring an environment of 20×20 m^2 with 300 features, see Fig. 2. Each robot executes 70 motion steps along a path of approximately 30 m. The robots estimate their motion based on odometry information that is corrupted with a noise of standard deviation $\sigma_x, \sigma_y = 0.4$ cm for the translations and $\sigma_\theta = 1$ degree for the orientations. They sense the environment using an omnidirectional camera that gives bearing measurement to features within 360 degrees around the robot (to better reproduce real situations, marks at distances greater than 6 m are not observed by the robots). The measurements are corrupted with a noise of 0.5 degrees standard deviation.

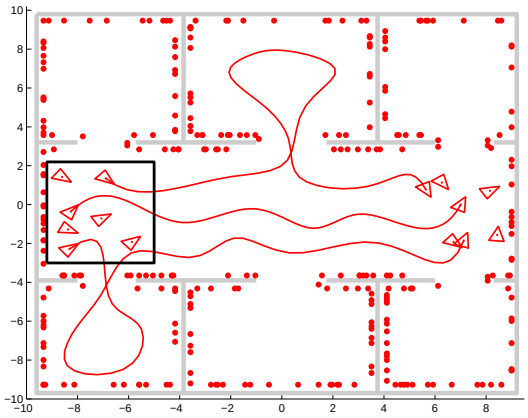
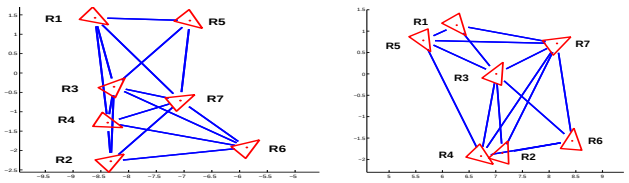


Figure 2: A team of 7 robots explore an environment of 20×20 m^2. Gray areas are walls and red dots are the ground-truth location of landmarks. Initially, the robots are placed in the black box region. From these initial poses, they reach consensus on the global reference frame. After that, they explore the environment and build their maps according this global reference frame. We display the trajectories followed by robots 2, 3, and 5, together with the final poses of the 7 robots.

Initially, the robots are placed within the black rectangle in Fig. 2. We assume that a pair of robots can exchange information and compute their relative poses if they are within a distance of 3 m. The robots execute the consensus on the global reference frame algorithm (Section 3) using the initial communication graph $\mathcal{G}^0$ (Fig. 3 (a)). Based on local interactions with its neighbors, each robot i computes the position and orientation of the center of mass (Fig. 4) of the robot team. The estimates $(T_G^i(t), R_G^i(t))$ of each robot (gray triangles) converge very fast to the correct center of mass $(T_G^{\mathbf{w}}, R_G^{\mathbf{w}})$ (red triangle). The robots execute $5n^2 = 245$ iterations to reach estimation errors smaller than a one percent of the initial error. Note that this



(a) Comm. graph $\mathcal{G}^0$ **(b) Comm. graph $\mathcal{G}$**

Figure 3: Communication graphs associated to the initial (a) and final (b) robot poses in Fig. 2. There is a link (blue solid line) between any pair of robot poses (red triangles) that are within a distance of 3 m.

is a worst-case bound and for this experiment, after the first 10 iterations the robots have already reached this precision (see Fig. 5). The errors in the $x-$ and $y-$coordinates (Fig. 5 (a), (b)) are less than 0.11 cm and 2.12 cm respectively, and the error in the estimated orientation (Fig. 5 (c)) is less than 0.43 degrees.

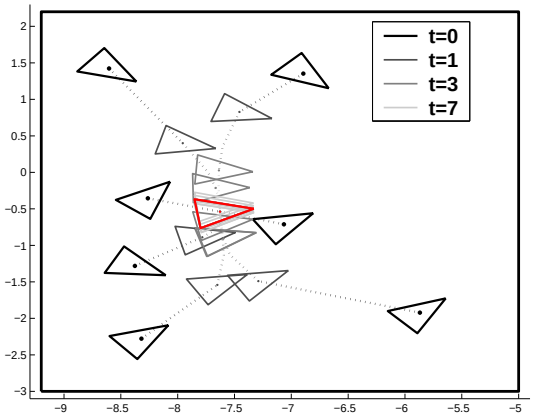
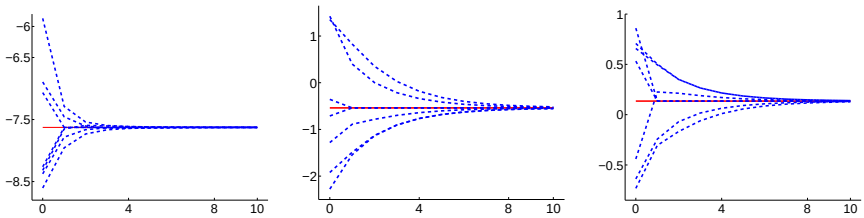


Figure 4: From the initial poses within the black box in Fig. 2, the robots reach a consensus on the global reference frame. Initially ($t = 0$), each robot i estimates the center of mass $(T^i_G(t), R^i_G(t))$ as its own pose (black triangles). During successive iterations, they update $(T^i_G(t), R^i_G(t))$ based on their neighbors estimates and their relative poses $(T^i_j(t), R^i_j(t))$. We display the center of mass estimated by the 7 robots (gray triangles), transformed into a common reference frame, for iterations $t = 1, 3, 7$. Their estimates after 7 iterations are very close to the ground truth center of mass (T^w_G, R^w_G) (red triangle).



(a) $x-$coordinate **(b) $y-$coordinate** **(c) orientation**

Figure 5: Position and orientation of the center of mass estimated by the 7 robots (blue dashed) compared to the true center of mass (red solid), along 10 iterations.

After that, each robot explores the environment and builds its local map expressed into the global reference frame (Fig. 6). Due to the presence of obstacles (gray areas), each robot may have not observed some landmarks. Besides, the precision of the estimated positions (blue crosses and ellipses) of the landmarks depends on the trajectory followed by each robot.

When they finish the exploration, they execute the consistent data association algorithm (Section 4) under the communication graph in Fig. 3 (b). The local data associations $\mathcal{F}(\mathcal{S}_i, \mathcal{S}_j)$ are obtained by applying

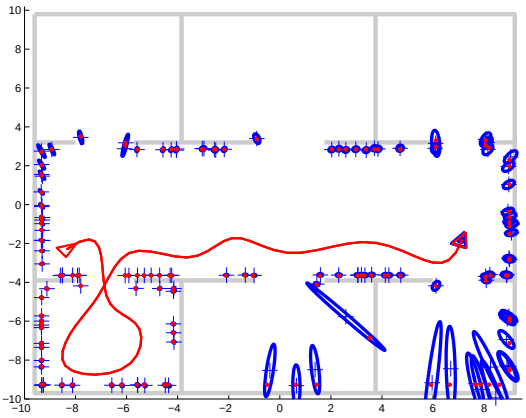


Figure 6: Local map estimated by robot 2. The landmarks close to its trajectory (red line) have been estimated (blue crosses and ellipses) with a high precision. Besides, its estimated positions (blue crosses) are very close to the ground truth locations (red dots). Due to the presence of obstacles (gray areas) some of the landmarks have not been observed, or have been estimated with high uncertainty.

the JCBB method [15] to the local maps of any pair of neighboring robots $(i, j) \in \mathcal{V}$. Since all the trajectories followed by the robots traverse the main corridor (Fig. 2) there is a high overlapping between their local maps (Table 2). Given any 2 local maps with approx. 122 features, there are approximately 89 true matches (ground truth). We can see that, although the local data association method has found a high amount of the ground truth links (good links or true positives), it has also missed a few of them (missing links or false negatives). In addition, some additional links have been detected that link together different features (spurious links or false positives).

The 858 features within all the local maps are observations of 300 different landmarks in the ground truth sense (association sets). From them, 184 were observed by a single robot (ground truth exclusive features), and the remaining where observed by around 6 robots (ground truth size of the remaining association sets). As previously stated, there is a high overlapping between the maps, and each non-exclusive feature has been observed by almost every robot. In the data association graph $\mathcal{G}_{da}$ however, only 296 association sets have been obtained, which means that different features have been mixed up together. There are 184 exclusive features (ground truth exclusive features), although the local data association algorithm has found 187 exclusive features. These additional 3 exclusive features appear due to the presence of the three outliers, the features with high covariance ellipses in Fig. 7. Since their positions have been wrongly estimated, the local data association method has failed to correctly associate them.

The robots execute Algorithm 4.1 on the non-exclusive features to detect and solve any inconsistent associations. From the 109 non exclusive association sets, 102 of them are consistent, and its associated 591 features are classified as consistent (Table. 3). The remaining 7 sets are conflictive, and they have associated 80 conflictive features. After executing the resolution algorithm on the 80 conflictive features, all of them are resolved and the process finishes. The original 7 conflictive sets (association sets resolved) are partitioned into 14 consistent sets (association sets resulting). Due to these additional sets, the number of non-exclusive association sets which initially was 109 (Table 2) is increased into 116 $(109 - 7 + 14)$ after executing the algorithm (Table 4).

The comparison between the final data association graph and the ground truth information can be seen in Table 4. Since the resolution algorithm is based on link deletion, the number of links here is lower than in Table 2. However, the number of association sets is closer to the ground truth results. From the 303 obtained association sets, 3 of them are due to the three outliers in Fig. 7. Thus, there are 300 remaining association sets, which is exactly the same number of association sets in the ground truth data. The same behavior is observed regarding their sizes. This means that the resulting associations are similar to the ground truth ones in spite of the fact that they have less links. From the 26 links erased from $\mathcal{G}_{da}$, 22 were spurious links, and only 4 where good links that now are missing.

Table 2: Local data associations.

Features	**Per local map**	**Total**
Features observed	122	**858**
Data associations	**Per pair of local maps**	**Total**
Links (ground truth)	89	2860
Links	88	2820
Good links	85	2750
Missing links	3	110
Spurious links	2	70
Association sets	**Obtained**	**Ground truth**
Association sets	**296**	**300**
Exclusive features	**187**	**184**
Non-exclusive assoc.	**109**	116
Size of non-exclusive	6.1	5.8

Table 3: Detection and resolution of inconsistent associations.

Detection	**Conflictive**	**Consistent**
Association sets	7	102
Features	80	591
Resolution	**Resolved**	**Resulting**
Association sets	7	14
Features	80	80

After associating their features, the 7 robots compute the global map as described in Section 5, under the communication graph in Fig. 3 (b). Here, we just display the global map $\mathbf{x}_G^i(t), \Sigma_G^i(t)$ of robot 2 after 5 iterations. We provide a deeper analysis of the performance of the map fusion algorithm under real data and different communication networks in Section 7.2 below. Since the communication graph has a high connectivity, in a few iterations each robot has received information from any other robot. After 5 iterations, the global map at robot 2 already contains precise estimates of the whole explored environment (Fig. 7).

7.2. Real Experiments

In this section, we illustrate the performance of our algorithm under real data and different communication schemes. We use a data set from [35] with bearing information obtained with vision in an environment of 60×45 m^2 performing 3297 steps. It is an indoor scenario where the robot moves along corridors and rooms. The data set contains real odometry data and images captured at every step (Fig. 8). The images are processed and measurements to natural landmarks are provided. The natural landmarks are vertical lines extracted from the images and processed in the form of bearing-only data. The observations in the dataset are labeled so that we have the ground-truth data association. This dataset is very challenging for a conventional visual map building algorithm due to the limited field of view of the camera (Sony EVI-371DG). Furthermore, the camera is pointing forward in the same direction of robot motion and the robot traverses rooms and corridors with few features in common. Notice that this situation is much more complex than situations where the camera can achieve big parallax, or systems with omnidirectional cameras, where features within 360 degrees around the robot are observed.

We have carried out these experiments with 9 robots. The total area covered by the robots is a square of 30×30 m^2 (Fig. 9). We run a separate SLAM in each robot and obtain 9 maps. We use a bearing-only SLAM algorithm with features parameterized in inverse-depth [36] followed by a transform to Cartesian coordinates (not discussed here) before the merging process. The reader is referred to [37] to find a detailed explanation of the SLAM algorithm we used. We express the local maps in global coordinates according to

Table 4: Results after detecting and solving the inconsistencies.

Features	**Per local map**	**Total**
Features observed	122	858
Data associations	**Per pair of local maps**	**Total**
Links (ground truth)	89	2860
Links	87	2794 **(-26)**
Good links	85	2746 **(-4)**
Missing links	3	114 **(+4)**
Spurious links	2	48 **(-22)**
Association sets	**Obtained**	**Ground truth**
Association sets	303	300
Exclusive features	187	184
Non-exclusive assoc.	116	116
Size of non-exclusive	5.7	5.8

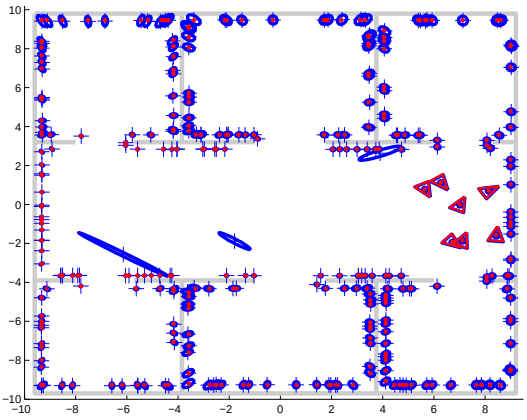


Figure 7: Global map $\mathbf{x}_G^i(t), \Sigma_G^i(t)$ estimated by robot 2 after $t = 5$ iterations. Red dots are the ground truth position of the features while blue crosses and ellipses are their estimated positions. Red triangles are the ground truth poses of the 7 robots after the exploration, and blue triangles are their estimated poses in the global map of robot 2. The three landmarks with high covariance ellipses are outliers that have been wrongly estimated by one of the robots. They have been classified by the algorithm as exclusive, giving rise to the presence of 3 additional exclusive sets in the final association map (303) compared to the ground truth information (300).

the relative robot poses seen in Fig. 9, obtaining the results shown in Fig. 10. Note that this is the result of putting the maps together, without applying a merging method. The team of robots execute the fusion algorithm presented in Section 5 to merge the local maps.

We study the behavior of the map merging method under three different scenarios: a fixed communication graph, a graph with switching topology and a graph with link failures (Fig. 11). We illustrate the performance of our algorithm by comparing the global map estimated by the robots along the iterations with the actual global map. We use two features to do this: F23, which has been observed by several robots, and F368, which belongs to a room visited exclusively by robot 4. In Figs. 12, 13 we show the estimated information matrices $I_G^i(t)$ and vectors $\mathbf{i}_G^i(t)$ (colored lines) during 40 iterations, compared to the global map I_G, $\mathbf{i}_G$ (black line). We display the subcomponent associated to the x-coordinate of features F23 and F368. As can be observed, in all cases the estimates converge to the average value very fast. However, for F368, the consensus is reached faster than for F23. This happens because only robot 4 possess an initial value for F368 (Fig. 13, iteration 0). As the other robots receive information of F368, their estimates are displayed in colors. We can see that, since this initial value is the unique source of information for F368, the other robots

Figure 8: An example of the images used by the 9 robots during the navigation to test the proposed method [35]. Although the data set also provides artificial landmarks (white circles on the floor), we test the algorithm using the lines extracted from natural landmarks (in yellow).

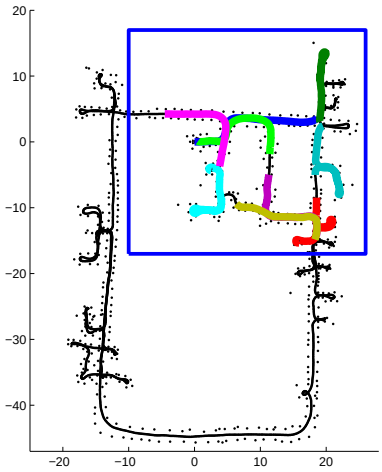


Figure 9: Trajectories followed by the 9 robots. They cover a region of 30×30 m^2 of the whole dataset map. In order to give an idea of the scene structure, we display in black the path in the dataset and a set of artificial landmarks (black dots) placed on both sides of the trajectory, which are not used in the experiment. Here, the rooms can be identified since robots enter and leave them describing short trajectories. The long, straight motions correspond to corridors.

do not disagree with this information and they just incorporate it into their estimates. In a few iterations, all the robots possess an estimate for F368 very close to the average value. However, for F23 there exist many initial values at iteration 0 (Fig. 12) from robots 1,7,8,9. The robots receive different information for F23 from different sources and, therefore, they must reach an agreement. As a result, there is a higher discrepancy in their estimates.

We analyze the evolution of the mean $\mathbf{x}_G^i(t)$ and covariance $\Sigma_G^i(t)$ (45) for F23 and F368 estimated by each robot i (Figs. 14, 15). We compare them with the mean $\mathbf{x}_G$ and covariance Σ_G of the global map, and with the numerical covariance $Q_G^i(t)$ at each iteration. It can be seen that $\mathbf{x}_G^i(t)$ converges to $\mathbf{x}_G$, and $\Sigma_G^i(t)$ converges to $n\Sigma_G$. Besides, it can also be seen that the numerical covariance $Q_G^i(t)$ remains bounded by $\Sigma_G^i(t)$ for all the iterations (Figs. 14, 15). When we analyzed the estimates in the information form, we observed that only robot 4 was providing information for F368. However, when we study the evolution of the estimates in the mean and covariance form (Fig. 14, 15), we can see that robots which did not observe F368 are providing estimates which disagree with those from robot 4. This is due to the effect of the correlations between F368 and the other features. Therefore, modifications in the estimates of features correlated with

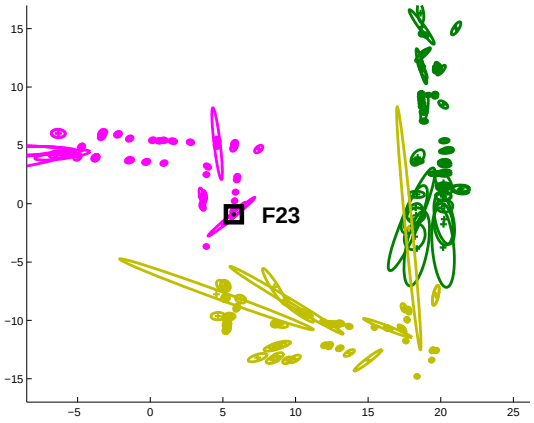
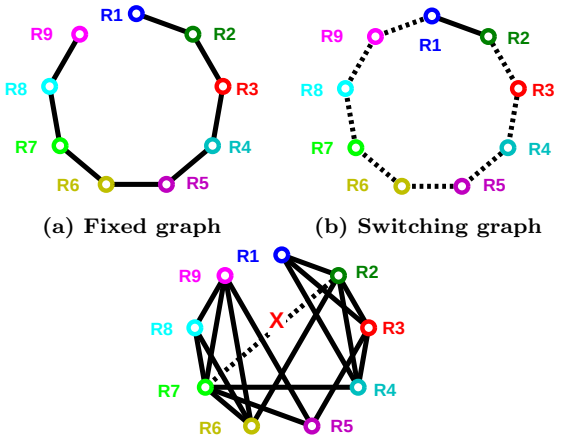


Figure 10: Local maps obtained by robots 2 (green), 6 (yellow), and 9 (pink) after following their trajectories in Fig. 9. The feature F23 within the black box will be used for testing purposes within this section.



(a) Fixed graph **(b) Switching graph**

(c) Link failure graph

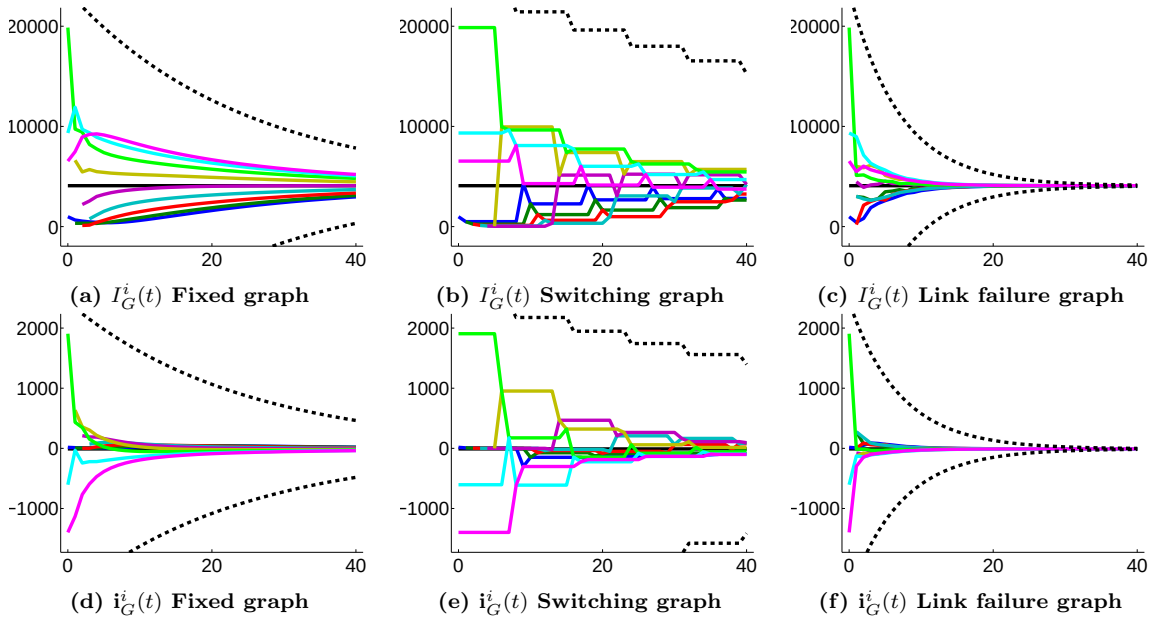
Figure 11: Communication graphs. (a) The topology is a string, with robots 1 and 9 in the extremes. Each robot has two neighbors, except the extreme robots, that have a single neighbor. This graph remains fixed for all the iterations of the algorithm. (b) For each iteration t, there exists a single edge linking robots $((t-1) \mod 9)+1$ and $(t \mod 9)+1$. (c) A connected communication graph where at each iteration one of its link fails.

F368 produce modifications in the estimate of F368.

We analyze the effects of the communication topology on the performance of the algorithm. In the fixed and the switching communication graphs (Fig. 12, first and second column), the convergence is slower than for the link failure graph (Fig. 12, third column). In this fixed graph (Fig. 11 (a)) the topology is a string, with robots 1 and 9 in the extremes. This is a specially bad configuration since the time needed to propagate information from the extreme robots to the whole network is maximal. The per step convergence factor $\gamma = |\lambda_2(W)|$ (Section 6) depends on the Metropolis weight matrix, which is

$$W = \frac{1}{3}\begin{bmatrix} 2 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \mathbf{0} & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \mathbf{0} & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}. \tag{49}$$

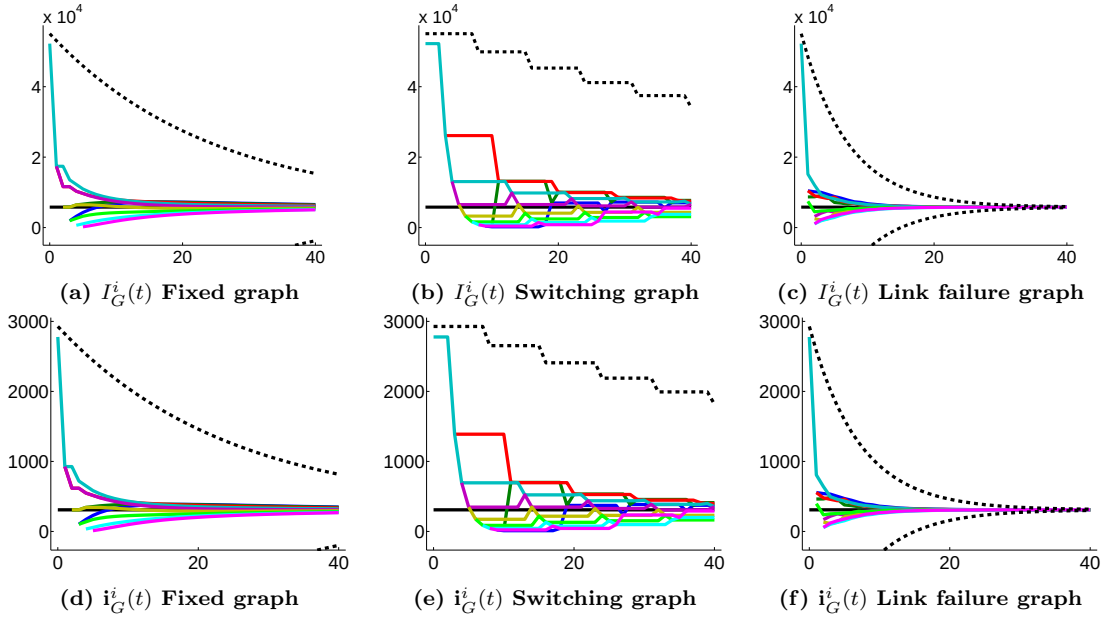
We obtain a value for $\gamma = 0.96$ close to 1. This produces a slow convergence. The convergence bounds are



(a) $I^i_G(t)$ **Fixed graph** **(b)** $I^i_G(t)$ **Switching graph** **(c)** $I^i_G(t)$ **Link failure graph**

(d) $\mathbf{i}^i_G(t)$ **Fixed graph** **(e)** $\mathbf{i}^i_G(t)$ **Switching graph** **(f)** $\mathbf{i}^i_G(t)$ **Link failure graph**

Figure 12: Estimated position ($x-$coordinate) of F23 at each robot along 40 iterations. We display its associated components within the information matrices $I^i_G(t)$ (first row) and vectors $\mathbf{i}^i_G(t)$ (second row). We analyze the results for the fixed (first column), the switching (second column) and the link failure (third column) communication graphs.

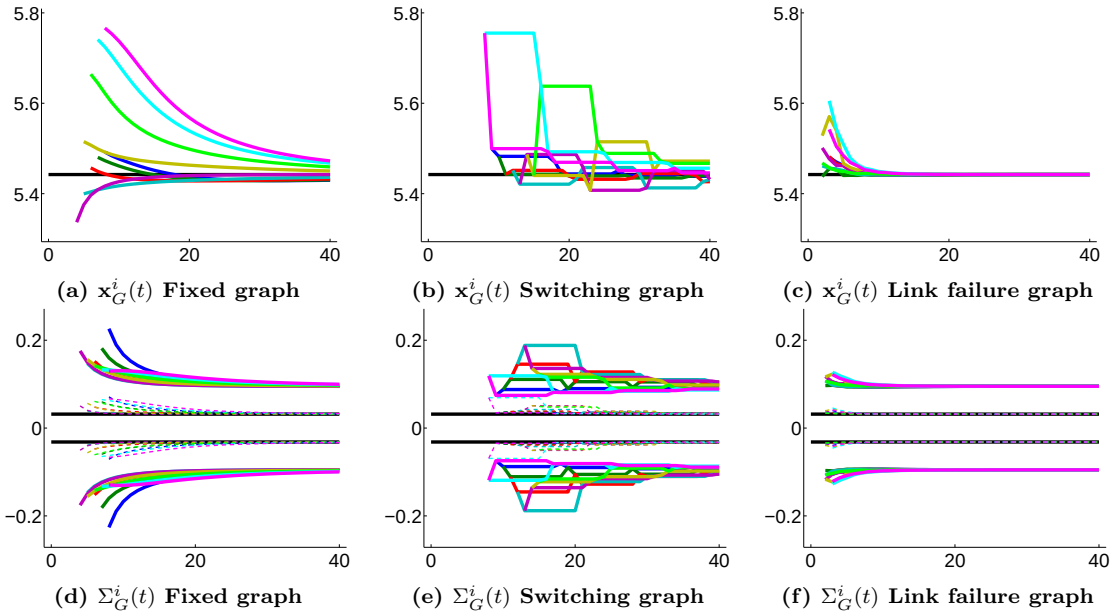


(a) $I^i_G(t)$ **Fixed graph** **(b)** $I^i_G(t)$ **Switching graph** **(c)** $I^i_G(t)$ **Link failure graph**

(d) $\mathbf{i}^i_G(t)$ **Fixed graph** **(e)** $\mathbf{i}^i_G(t)$ **Switching graph** **(f)** $\mathbf{i}^i_G(t)$ **Link failure graph**

Figure 13: Feature F368, information matrices $I^i_G(t)$ and vectors $\mathbf{i}^i_G(t)$. We display the entry within the information matrix $I^i_G(t)$ associated to the $x-$coordinate of F368, along 40 iterations. For the three network topologies, the estimates at each robot (color solid lines) remain within the theoretical bounds (black dashed lines) while they asymptotically approach the global map I_G (black solid line).

displayed in black dashed lines (Figs. 12, 13, first column). In the switching graph case (Fig. 11 (b)), at every time instant, only one communication link exists in the graph and this sequence takes place in a circular fashion. This is a very extreme communication scheme where, although the conditions for convergence are



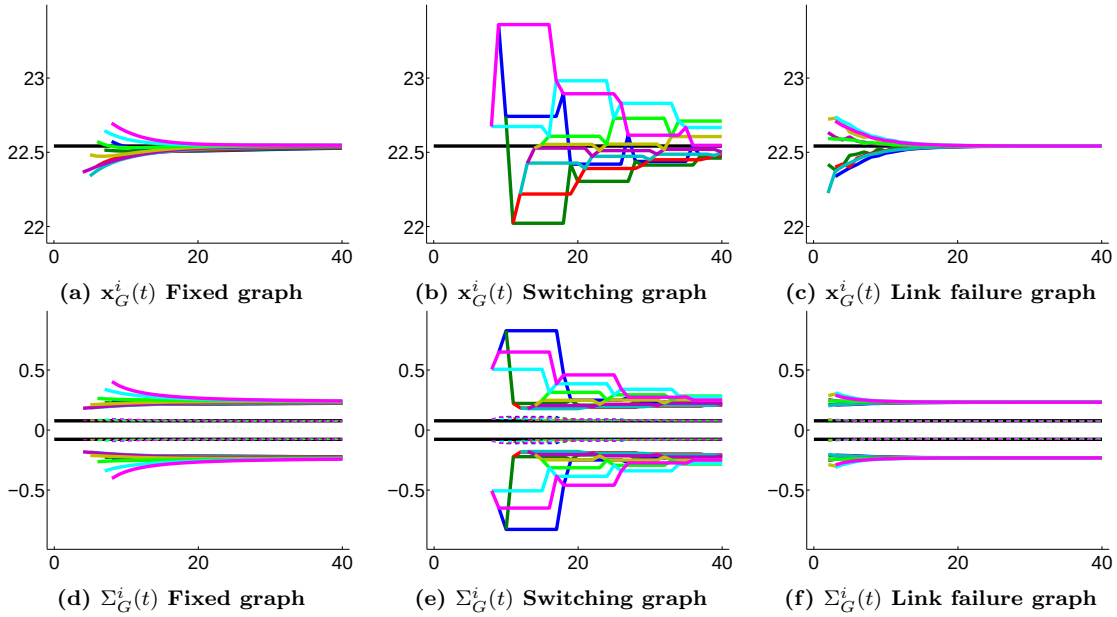
(a) $\mathbf{x}_G^i(t)$ **Fixed graph** **(b)** $\mathbf{x}_G^i(t)$ **Switching graph** **(c)** $\mathbf{x}_G^i(t)$ **Link failure graph**

(d) $\Sigma_G^i(t)$ **Fixed graph** **(e)** $\Sigma_G^i(t)$ **Switching graph** **(f)** $\Sigma_G^i(t)$ **Link failure graph**

Figure 14: Estimated position ($x-$coordinate) of F23 at each robot along 40 iterations. We display its associated components within the mean $\mathbf{x}_G^i(t)$ (first row) and covariance $\Sigma_G^i(t)$ (second row) estimated by each robot i. We analyze the results for the fixed (first column), the switching (second column) and the link failure (third column) communication graphs. All the mean estimates $\mathbf{x}_G^i(t)$ (color solid lines) approach the average value (black solid line). The numerical covariance matrix $Q_G^i(t)$ (color dashed lines) asymptotically approaches the covariance matrix Σ_G (black solid line) of the global map. The numerical covariance $Q_G^i(t)$, which cannot be computed by the robots using local information, is bounded by the locally computed covariance matrix $\Sigma_G^i(t)$ (color solid lines). This matrix converges to $n\Sigma_G$.

satisfied, the converge speed is expected to be slow. We can see that (Figs. 12, 13, second column) for each robot, estimates remain unchanged during long periods of time, then they experiment two consecutive changes, and then they remain unchanged again. Each robot remains isolated during 7 iterations, maintaining its estimates unchanged. Then, it exchanges information with its previous neighbor and, in the next iteration, with its next neighbor. In our case, at each iteration t, there exists a single edge linking robots $((t-1) \mod 9)+1$ and $(t \mod 9)+1$. The index of joint connectivity is $\tau = 8$ since every 8 iterations the joint graph is connected. There are only 9 different Metropolis weight matrices $W(t)$, depending on the linked robots at time t, that are repeated successively. We obtained a value for $\delta = 0.89$ using (47). We draw the bounds using black dashed lines (Figs. 12, 13, second column). In the link-failure graph (Fig. 11 (c)), at each iteration one of the links in the graph fails although the graph remains connected. Thus, we obtain an index of joint connectivity of $\tau = 1$. Evaluating all the possible Metropolis weight matrices in this graph, we obtain $\delta = 0.80$. We show the convergence speed bounds (Figs. 12, 13, third column) using black dashed lines. This communication scheme exhibits the fastest convergence speed, since $0.80^t \leq 0.96^t \leq 0.89^{\lfloor t/8 \rfloor}$ for all $t \geq 0$. This faster convergence can also be observed in the estimated mean and covariance (Figs. 14, 15), where the estimates approach the global map faster for the link failure graph (third column). It is noted that regardless of the presence of link failures or changes in the communication topology, the numerical covariance remains bounded by the locally computed covariance matrix (Figs. 14, 15).

In addition, we display (Fig. 16) the global map estimated by robot 1 after 5, and 20 iterations (colored lines) of the merging algorithm, and under the fixed communication graph (Fig. 11 (a)). The maps estimated by the 9 robots are similar. We compare the estimates at robot 1 to the global map in (4) (black lines). Due to the network configuration, after 5 iterations robot 1 has received information from the initial local maps of robots 1 to 6. However, it still knows nothing of the local maps of robots 7 to 9 (Fig. 16 (a)). As previously stated, this fixed communication graph has a slow convergence speed. However, after 20 iterations the map estimated by robot 1 is very close to the global map (Fig. 16 (b)). In addition, it is observed that

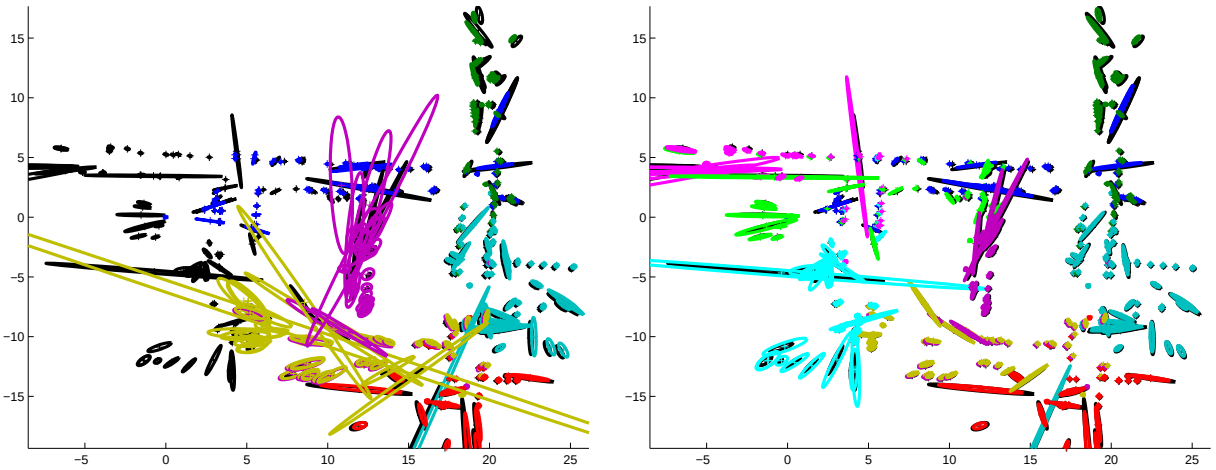


(a) $\mathbf{x}_G^i(t)$ **Fixed graph** **(b)** $\mathbf{x}_G^i(t)$ **Switching graph** **(c)** $\mathbf{x}_G^i(t)$ **Link failure graph**

(d) $\Sigma_G^i(t)$ **Fixed graph** **(e)** $\Sigma_G^i(t)$ **Switching graph** **(f)** $\Sigma_G^i(t)$ **Link failure graph**

Figure 15: Estimated position ($x-$coordinate) of F368 at each robot along 40 iterations. We display its associated components within the mean $\mathbf{x}_G^i(t)$ (first row) and covariance $\Sigma_G^i(t)$ (second row) estimated by each robot i. We analyze the results for the fixed (first column), the switching (second column) and the link failure (third column) communication graphs. All the mean estimates $\mathbf{x}_G^i(t)$ (color solid lines) approach the average value (black solid line). The numerical covariance matrix $Q_G^i(t)$ (color dashed lines) asymptotically approaches the covariance matrix Σ_G (black solid line) of the global map. The numerical covariance $Q_G^i(t)$, which cannot be computed by the robots using local information, is bounded by the locally computed covariance matrix $\Sigma_G^i(t)$ (color solid lines). This matrix converges to $n\Sigma_G$.

the information fusion leads to a great improvement in the map quality, where not only the uncertainty is greatly decreased, but also the local maps are corrected.



(a) 5 iterations **(b) 20 iterations**

Figure 16: Global map estimated by robot 1 after 5 **(a)** and 20 **(b)** iterations of the merging algorithm, and under the fixed communication graph (Fig. 11 (a)). Different colors identify the source local map. Although the global map contains a single estimate per feature, the features observed by more than one robot are displayed by multiple colored ellipses. The global map $\mathbf{x}_G$, Σ_G is displayed in black. Although there is a slow convergence speed associated to the fixed communication graph used here, the estimated global map after 20 iterations is very close to the global map.

Finally, we have studied the performance of the map merging algorithm in terms of its execution times (Fig. 17). During the first iterations, the peaks on the execution times are due to the expansion and arrangement of the information matrices and vectors $I^i_G(t)$ and $\mathbf{i}^i_G(t)$ that are performed by the robots whenever they discover new features in its neighbors' information. These memory allocation operations, which are computationally expensive, give rise to this behavior. In the fixed graph case (in blue solid) this situation continues until iteration 5, when robot 5, the robot in the central position within the string graph, has received information from all the robots. Its information matrix $I^i_G(5)$ reaches its maximum size and, from here to the end of the experiment, its global map estimate changes but its size remains unchanged. The execution times reach a peak at iteration 5 and from here on, it decreases. From iterations 5 to 9 other robots achieve the maximal size of their matrices $I^i_G(t)$, and finally, from iteration 9 to the end of the experiment, the global map size remains unchanged for all the robots. For this reason, we can see that the execution times are drastically reduced from iteration 9 to the end of the experiment. For the switching graph (in black dashed), the first robots that receive information from all the others are 8 and 9, at iteration 8, and from iterations 9 to 15 robots 1 to 7 successively expand their maps to the maximum size. Finally, from iteration 16 to the end, all the robot's global map estimates present the maximal size and only its contents change. For this reason, the execution times decrease. Finally, for the link failure graph (in dotted red), due to its higher connectivity, all the expansion operations are carried out during the first 4 iterations, giving rise to the larger peak observed in the plot. After these expansion operations, the execution times are similar for the three communication graphs.

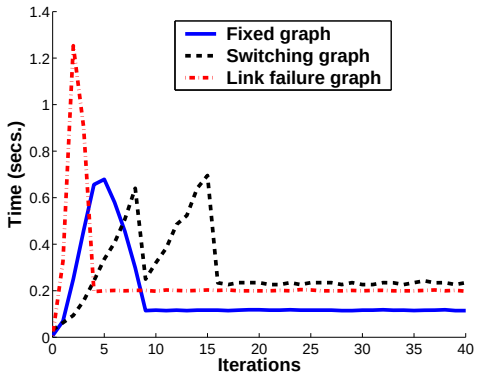


Figure 17: Execution times per iteration and robot exhibited by the merging algorithm under the three tested communication graphs (times obtained with a Q9550 processor and 3.24 GB RAM).

8. CONCLUSIONS

We have presented a new method for merging stochastic feature-based maps acquired by a team of robots with heterogeneous visual sensors for scenarios with limited communication. Each robot explores an unknown environment and builds a local stochastic map of the explored region. The robots are placed at unknown poses and before fusing their maps, they must reach a consensus on a global reference frame. This can be executed either before or after the exploration. Once they have transformed their local maps into the global frame, the robots solve a consistent data association that produces a conflict-free global association graph. Additionally, each robot obtains a global labeling for its features. Then, they fuse their local maps and build a global map according to this conflict-free data association. The whole method is fully decentralized, relying exclusively on local interactions between neighboring robots. Under the mild condition that the communication graphs must be connected, the estimates at each robot asymptotically converge to the global map. Moreover, the intermediate estimates at each robot present interesting properties that allow their use at any time: (i) the mean of the global map estimated by each robot is unbiased at each iteration; (ii) the numerical covariance of the global map estimated by each robot, which cannot be locally

computed, is bounded by the locally computed covariance. Experimental results show the performance of the method for robots equipped with omnidirectional and conventional cameras. The robustness of the map fusion algorithm under link failures and changes in the communication topology has been demonstrated both theoretically and experimentally. To the best of our knowledge this is the first method that solves the map merging problem in a fully decentralized way.

Acknowledgments

This work was partially supported by project MICINN DPI2009-08126, grant MEC BES-2007-14772 and NSF award CCF-0917166.

The data set used in the experiments was provided by U. Frese and J. Kurlbaum. The authors gratefully acknowledge them for providing data and support.

Appendix A. Averaging algorithms and Metropolis weights

Throughout this paper, we frequently refer to averaging algorithms. They have become very popular in sensor networks due to their capability to reach agreement in a distributed way. Let us assume that each robot $i \in \mathcal{V}$ has initially a scalar value $z_i(0) \in \mathbb{R}$. Let $W \in \mathbb{R}^{n\times n}_{\geq 0}$ be a doubly stochastic matrix such that $W_{i,j} > 0$ if $(i,j) \in \mathcal{E}$ and $W_{i,j} = 0$ when $j \notin \mathcal{N}_i$. This matrix is such that $W_{i,i} \in [\alpha, 1]$, $W_{i,j} \in \{0\} \cup [\alpha, 1]$ for all $i, j \in \mathcal{V}$, for some $\alpha \in (0, 1]$. Assume the communication graph $\mathcal{G}$ is connected. If each robot $i \in \mathcal{V}$ updates $z_i(t)$ at each time step $t \geq 0$ with the following averaging algorithm,

$$z_i(t+1) = \sum_{j=1}^{n} W_{i,j}\ z_j(t), \tag{A.1}$$

then, as $t \to \infty$, the variables $z_i(t)$ reach the same value for all $i \in \mathcal{V}$, i.e., they reach a consensus. Moreover, the consensus value is the average of the initial values,

$$\lim_{t\to\infty} z_i(t) = z_\star = \frac{1}{n}\sum_{j=1}^{n} z_j(0), \tag{A.2}$$

for all $i \in \mathcal{V}$ [32, 38]. Observe that each robot i updates its variables $z_i(t)$ using local information since the weight matrix has zero entries for non-neighboring nodes, $W_{i,j} = 0$ when $j \notin \mathcal{N}_i$. Let $\mathbf{e}(t) = (z_1(t), \ldots, z_n(t))^T - (z_\star, \ldots, z_\star)^T$ be the error vector at iteration t. The number of iterations t necessary for reaching $||\mathbf{e}(t)||_2/||\mathbf{e}(0)||_2 < \epsilon$ ranges between a single iteration for complete graphs, and order $n^2 \log(\epsilon^{-1})$ iterations for networks with lower connectivity like string and circular graphs [32, Theorems 1.79 and 1.80].

A common choice for the matrix $W \in \mathbb{R}^{n\times n}$ is given by the Metropolis weights given by [28],

$$W_{i,j} = \begin{cases} \frac{1}{1+\max\{|\mathcal{N}_i|,|\mathcal{N}_j|\}} & if\ j \in \mathcal{N}_i, j \neq i, \\ 0 & if\ j \notin \mathcal{N}_i, j \neq i, \\ 1 - \sum_{j\in\mathcal{N}_i} W_{i,j}, & if\ j = i, \end{cases} \tag{A.3}$$

for $i, j \in \mathcal{V}$, $j \neq i$, where $|\mathcal{N}_i|$, $|\mathcal{N}_j|$ are the number of neighbors of nodes i, j. Note that each agent can compute the weights that affect its evolution using only local information. The algorithm (A.1) using the Metropolis weights W converges to the average of the inputs.

Appendix B. Sensor fusion based on averaging

In the sensor fusion algorithm proposed in [28], each robot i maintains the variables $\hat{I}^i_G(t) \in \mathbb{R}^{\mathcal{M}_G\times\mathcal{M}_G}$, $\hat{\mathbf{i}}^i_G(t) \in \mathbb{R}^{\mathcal{M}_G}$, initialized as

$$\hat{I}^i_G(0) = H_i^T \Sigma_i^{-1} H_i, \qquad\qquad \hat{\mathbf{i}}^i_G(0) = H_i^T \Sigma_i^{-1} \hat{\mathbf{x}}_i, \tag{B.1}$$

and updated at each time step $t \geq 0$ by

$$\hat{I}_G^i(t+1) = \sum_{j=1}^{n} W_{i,j} \hat{I}_G^j(t), \qquad \hat{\mathbf{i}}_G^i(t+1) = \sum_{j=1}^{n} W_{i,j} \hat{\mathbf{i}}_G^j(t), \tag{B.2}$$

where $\hat{\mathbf{x}}_i$, Σ_i is the local map of robot i and $W_{i,j}$ are the Metropolis weights (A.3) of $\mathcal{G}$. This is a distributed averaging algorithm that, for a connected communication graph $\mathcal{G}$, guarantees that the estimates asymptotically converge to the average of the initial states,

$$\lim_{t\to\infty} \hat{I}_G^i(t) = \frac{1}{n} I_G, \qquad \lim_{t\to\infty} \hat{\mathbf{i}}_G^i(t) = \frac{1}{n} \mathbf{i}_G, \tag{B.3}$$

being I_G, $\mathbf{i}_G$ the information matrix and vector of the global map (3). Then, the variables $\hat{\mathbf{x}}_G^i(t) \in \mathbb{R}^{\mathcal{M}_G}$ and $\hat{\Sigma}_G^i(t) \in \mathbb{R}^{\mathcal{M}_G \times \mathcal{M}_G}$ at each robot $i \in \mathcal{V}$ and each $t \geq 0$, defined as

$$\hat{\mathbf{x}}_G^i(t) = \left(\hat{I}_G^i(t)\right)^{-1} \hat{\mathbf{i}}_G^i(t), \qquad \hat{\Sigma}_G^i(t) = \left(\hat{I}_G^i(t)\right)^{-1}, \tag{B.4}$$

asymptotically converge to

$$\lim_{t\to\infty} \hat{\mathbf{x}}_G^i(t) = \hat{\mathbf{x}}_G, \qquad \lim_{t\to\infty} \hat{\Sigma}_G^i(t) = n\Sigma_G, \tag{B.5}$$

where $\hat{\mathbf{x}}_G$, Σ_G are the mean and covariance of the global map in (4).

This algorithm is fully distributed because the nodes only use information about its direct neighbors in the communication graph ($W_{i,j} = 0$ if $j \notin \mathcal{N}_i$). Besides, under mild connectivity conditions, it converges to the average even if the communication graph is time-varying, $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. The convergence is asymptotic, and hence the global map is obtained by each robot i in the limit as $t \to \infty$. However, for complete communication graphs, the strategy converges in a single iteration. For general networks, the intermediate estimates (B.4) have interesting properties that allow their use at any time t:

(i) the intermediate estimates $\hat{\mathbf{x}}_G^i(t)$ are unbiased,

$$\mathbf{E}\left[\hat{\mathbf{x}}_G^i(t)\right] = \mathbf{x}, \tag{B.6}$$

for all $t \geq 0$ and all $i \in \mathcal{V}$ such that the information matrix $\hat{I}_G^i(t)$ can be inverted [28, Theorem 3];

(ii) the numerical covariance $Q_G^i(t) \in \mathbb{R}^{\mathcal{M}_G \times \mathcal{M}_G}$ of $\hat{\mathbf{x}}_G^i(t)$ is bounded by the locally estimated global map covariance $\hat{\Sigma}_G^i(t)$,

$$Q_G^i(t) = \mathbf{E}\left[\left(\hat{\mathbf{x}}_G^i(t) - \mathbf{x}\right)\left(\hat{\mathbf{x}}_G^i(t) - \mathbf{x}\right)^T\right] = \left(\hat{I}_G^i(t)\right)^{-1} \left(\sum_{j=1}^{n} ([W^t]_{i,j})^2 H_j^T \Sigma_j^{-1} H_j\right) \left(\hat{I}_G^i(t)\right)^{-1}, \tag{B.7}$$

for all $t \geq 0$ and all $i \in \mathcal{V}$ such that $\hat{I}_G^i(t)$ can be inverted. Since $0 \leq [W^t]_{i,j} \leq 1$ for all $i, j \in \mathcal{V}$ and all $t \geq 0$, then $([W^t]_{i,j})^2 \leq [W^t]_{i,j}$ and

$$Q_G^i(t) \preceq \left(\hat{I}_G^i(t)\right)^{-1} = \hat{\Sigma}_G^i(t), \tag{B.8}$$

where $\preceq$ means that $\hat{\Sigma}_G^i(t) - Q_G^i(t)$ is a positive semidefinite matrix [27, Lemma 2.1]. Therefore, the numerical covariance $Q_G^i(t)$ is bounded by the locally estimated global map covariance $\hat{\Sigma}_G^i(t)$. In particular, since $\hat{\Sigma}_G^i(t) - Q_G^i(t)$ is a positive semidefinite matrix, the elements in the main diagonal satisfy $[\hat{\Sigma}_G^i(t) - Q_G^i(t)]_{r,r} \geq 0$, and thus $[\hat{\Sigma}_G^i(t)]_{r,r} \geq [Q_G^i(t)]_{r,r}$. Therefore, any decision taken by the robots based on the entries in the main diagonal of the covariance matrix, can also be taken based on $\hat{\Sigma}_G^i(t)$.

Those properties of the intermediate estimates allow the robots to make decisions based on their global map estimates at every time instant.

REFERENCES

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

[2] S. B. Williams, H. Durrant-Whyte, Towards multi-vehicle simultaneous localisation and mapping, in: IEEE Int. Conf. on Robotics and Automation, Washington, DC, USA, 2002, pp. 2743–2748.

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

[4] M. Pfingsthorn, B. Slamet, A. Visser, A scalable hybrid multi-robot SLAM method for highly detailed maps, in: U. Visser, F. Ribeiro, T. Ohashi, F. Dellaert (Eds.), Lecture Notes in Artificial Intelligence, Vol. 5001, 2008, pp. 457–464.

[5] K. Konolige, J. Gutmann, B. Limketkai, Distributed map-making, in: Workshop on Reasoning with Uncertainty in Robotics, Int. Joint Conf. on Artificial Intelligence, Acapulco, Mexico, 2003.

[6] L. M. Paz, J. D. Tardós, J. Neira, Divide and conquer: EKF SLAM in $o(n)$, IEEE Transactions on Robotics 24 (5) (2008) 1107–1120.

[7] S. Thrun, Y. Liu, Multi-robot SLAM with sparse extended information filters, in: Int. Symposium of Robotics Research, Sienna, Italy, 2003, pp. 254–266.

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

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

[10] X. S. Zhou, S. I. Roumeliotis, Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, China, 2006, pp. 1785–1792.

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

[12] C. Sagues, A. C. Murillo, J. J. Guerrero, T. Goedemé, T. Tuytelaars, L. V. Gool, Localization with omnidirectional images using the 1d radial trifocal tensor, in: IEEE Int. Conf. on Robotics and Automation, 2006, pp. 551–556.

[13] M. Kaess, F. Dellaert, Covariance recovery from a square root information matrix for data association, Robotics and Autonomous Systems 57 (12) (2009) 1198 – 1210.

[14] A. Gil, O. Reinoso, M. Ballesta, M. Julia, Multi-robot visual SLAM using a rao-blackwellized particle filter, Robotics and Autonomous Systems 58 (1) (2009) 68–80.

[15] J. Neira, J. D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Transactions on Robotics and Automation 17 (6) (2001) 890–897.

[16] T. Bailey, E. M. Nebot, J. K. Rosenblatt, H. F. Durrant-Whyte, Data association for mobile robot navigation: a graph theoretic approach, in: IEEE Int. Conf. on Robotics and Automation, San Francisco, USA, 2000, pp. 2512–2517.

[17] C. Cadena, F. Ramos, J. Neira, Efficient large scale SLAM including data association using the Combined Filter, in: European Conference on Mobile Robotics, ECMR, Mlini/Dubrovnik, Croatia, 2009, pp. 217–222.

[18] D. L. Rizzini, S. Caselli, Metric-topological maps from laser scans adjusted with incremental tree network optimizer, Robotics and Autonomous Systems 57 (10) (2009) 1036–1041.

[19] H. S. Lee, K. M. Lee, Multi-robot SLAM using ceiling vision, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, St. Louis, USA, 2009, pp. 912–917.

[20] V. Ferrari, T. Tuytelaars, L. V. Gool, Wide-baseline multiple-view correspondences, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, Madison, USA, 2003, pp. 718–725.

[21] E. M. Nebot, M. Bozorg, H. F. Durrant-Whyte, Decentralized architecture for asynchronous sensors, Autonomous Robots 6 (2) (1999) 147–164.

[22] R. Olfati-Saber, Distributed Kalman filter with embedded consensus filters, in: IEEE Conf. on Decision and Control, Seville, Spain, 2005, pp. 8179–8184.

[23] D. P. Spanos, R. Olfati-Saber, R. M. Murray, Distributed sensor fusion using dynamic consensus, in: IFAC World Congress, Prague, CZ, 2005.

[24] R. Carli, A. Chiuso, L. Schenato, S. Zampieri, Distributed kalman filtering based on consensus strategies, IEEE Journal on Selected Areas in Communications 26 (2008) 622–633.

[25] R. Olfati-Saber, Distributed Kalman filtering for sensor networks, in: IEEE Conf. on Decision and Control, New Orleans, LA, 2007, pp. 5492–5498.

[26] P. Alriksson, A. Rantzer, Distributed Kalman filtering using weighted averaging, in: Int. Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006.

[27] G. Calafiore, F. Abrate, Distributed linear estimation over sensor networks, International Journal of Control 82 (5) (2009) 868–882.

[28] L. Xiao, S. Boyd, S. Lall, A scheme for robust distributed sensor fusion based on average consensus, in: Symposium on Information Processing of Sensor Networks (IPSN), Los Angeles, CA, 2005, pp. 63–70.

[29] R. Aragues, E. Montijano, C. Sagues, Consistent data association in multi-robot systems with limited communications, in: Robotics: Science and Systems, Zaragoza, Spain, 2010.

[30] M. Moakher, Means and averaging in the group of rotations, SIAM Journal on Matrix Analysis and Applications 24 (1) (2002) 1–16.

[31] R. Tron, R. Vidal, A. Terzis, Distributed pose averaging in camera networks via consensus on SE(3), in: ACM/IEEE International Conference on Distributed Smart Cameras, Stanford University, USA, 2008.

[32] F. Bullo, J. Cortés, S. Martínez, Distributed Control of Robotic Networks, Applied Mathematics Series, Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[33] L. Xiao, S. Boyd, Fast linear iterations for distributed averaging, Systems & Control Letters 53 (2004) 65–78.

[34] G. Calafiore, Distributed randomized algorithms for probabilistic performance analysis, Systems & Control Letters 58 (3) (2009) 202–212.
[35] U. Frese, J. Kurlbaum, A data set for data association, electronically available at http://www.sfbtr8.spatial-cognition.de/insidedataassociation/ (Jun. 2008).
[36] J. M. M. Montiel, J. Civera, J. Davison, Unified inverse depth parametrization for monocular SLAM, in: Robotics: Science and Systems, Philadelphia, USA, 2006.
[37] R. Aragues, C. Sagues, Parameterization and initialization of bearing-only information: a discussion, in: Int. Conf. on Informatics in Control, Automation and Robotics, Vol. RA-1, Funchal, Portugal, 2008, pp. 252–261.
[38] W. Ren, R. W. Beard, Distributed Consensus in Multi-vehicle Cooperative Control, Communications and Control Engineering, Springer Verlag, London, 2008.