



HAL
open science

Feature-based map merging with dynamic consensus on information increments

Rosario Aragues, Carlos Sagüés, Youcef Mezouar

► **To cite this version:**

Rosario Aragues, Carlos Sagüés, Youcef Mezouar. Feature-based map merging with dynamic consensus on information increments. *Autonomous Robots*, 2015, 38 (3), pp.243-259. 10.1007/s10514-014-9406-z . hal-02099924

HAL Id: hal-02099924

<https://hal.science/hal-02099924v1>

Submitted on 15 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Feature-Based Map Merging with Dynamic Consensus on Information Increments

Rosario Aragues · Carlos Sagues · Youcef Mezouar

Received: date / Accepted: date

Abstract We study the problem of feature-based map merging in robot networks. Along its operation, each robot observes the environment and builds and maintains a local map. Simultaneously, each robot communicates and computes the global map of the environment. The communication between the robots is range-limited. Our contributions are the proposal and careful study of the properties of an algorithm that considers separately robot poses and features positions, and that reaches consensus on the latest global map using the map increments between the previous and the current time steps. We give proofs of unbiasedness and consistency of this global map for all the robots, at each iteration. Our algorithm is fully distributed and does not rely on any particular communication topology. Under mild connectivity conditions on the communication graph, our merging algorithm asymptotically converges

to the global map. The proposed approach has been experimentally validated using real RGB-D images.

Keywords Distributed Robot Systems · Networked Robots · Distributed Sensor Fusion · Mapping

1 Introduction

Perception has a great importance in robotics. The availability of a local map allows each robot to make local decisions for, e.g., local navigation or collision avoidance. Teams of cooperative robots often need mechanisms for merging their locally acquired information and building a global representation of the environment. This global map can then be used by the robot team to make global decisions, such as cooperative exploration, or task assignment. As robots operate, they re-observe and improve features estimates, and they introduce new features in their local maps. The latest global map must be properly adapted to reflect this new information. The problem of dynamic map merging consists of correctly building the global map and updating it according to the newly acquired data.

An important challenge in multi-robot systems consists of the proper management of communications. Each robot has a limited communication range, and can only exchange data with its nearby team members, its *neighbors*. Several multi-robot algorithms have been proposed that take these restrictions into account, e.g., [16] for pursuit-evasion, or [19] for distributed transferable belief models. Besides, since the robots move, the set of neighbors change with time. Thus, multi-robot strategies must be specifically designed for coping with switching network topologies. In this paper we investigate the problem of dynamic map merging under limited communication and switching topologies, in which each

This work was supported by grants from the French program investissement d'avenir managed by the National Research Agency (ANR), the European Commission (Auvergne FEDER funds) and the Région Auvergne in the framework of the LabEx IMobS3 (ANR-10-LABX-16-01) and by projects from the Spanish Government DPI2009-08126, and DPI2012-32100.

R. Aragues (corresponding author)
Clermont Université, Institut Pascal, CNRS, UMR 6602, BP 10448, F-63000 Clermont-Ferrand, France
E-mail: raragues@unizar.es

C. Sagues
Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50018 Zaragoza, Spain
E-mail: csagues@unizar.es

Y. Mezouar
Clermont Université, IFMA, Institut Pascal, CNRS, UMR 6602, BP 10448, F-63000 Clermont-Ferrand, France
E-mail: mezouar@univ-bpclermont.fr

robot can only exchange data with its neighbors. Each robot builds a local map of the environment using its own measurements. Robots fuse their local maps and build a global map by applying distributed consensus filters on the information matrices and vectors of the local maps [10, 57]. Robots do not introduce information from the global map into their local maps; thus, local maps between different robots remain independent. After a certain time, the local maps of the robots contain more precise estimates of features, as well as newly discovered features. Robots compute the information increments between their current local maps, and the latest fused ones. Then, they run distributed consensus filters on these information increments to keep the global map up to date.

In our previous approach to the dynamic map merging problem [4], we used consensus algorithms [18, 36] that allowed the latest global map to be weighted with a forgetting factor, as the current global map was computed by the robots. This approach has two limitations: first, robots have to be synchronized, i.e., they must initiate every new map merging phase in a coordinated way; and in second place, the method was designed for graphs which remained fixed during a specific merging phase. Here we propose a method that does not suffer from these limitations. Each robot decides on its own if it propagates to the network its most recent map version, or if it continues merging the previous one. Thus, a robot may wait until there are enough differences relative to its previous local map, or until the latest features detected have been estimated with a certain accuracy. Besides, the proposed scheme benefits from the fact that the space-time fusion algorithm [57] is proved to converge under a wider variety of communication topologies; in particular, if the set of communication graphs that occur infinitely often is jointly connected. It includes, e.g., topologies that totally switch at every step, or sequences of disconnected graphs. A preliminary version of this work appears in [7]. Here we make a deep study of the properties of our maps, and of the memory and communication costs. Our main contributions are: (i) the proposal of an algorithm to reach consensus on the latest global map, using the map increments between the previous and the current time steps; (ii) the analysis of the properties of the robots estimates; in particular, we perform a novel and careful study of the consistency of the global map estimated by each robot at each step; and (iii) the study of the accuracy and consistency of the estimated maps with Monte-Carlo simulations.

The remaining of this paper is organized as follows. Section 2 discusses several state-of-the-art related methods. Section 3 formally describes the dynamic map

merging problem. Section 4 presents our dynamic map merging algorithm and discusses its properties. Section 5 evaluates the performance of the method, and Section 6 states the conclusions.

2 Related Work

Many multi-robot map merging solutions assume centralized schemes, all-to-all communication, or broadcasting methods. Examples include [23] for particle filters, [56] for multi-robot submaps, and [55] for graph maps of laser scans; similar schemes could be applied for many existing submapping methods [44]. The main limitation of these works is that they cannot easily cope with limited communications, switching topologies, or link failures.

Alternatively, distributed estimation methods [1, 12, 20, 28, 39, 41, 42, 53] could be used for computing the global merged map. These approaches consider a linear system without input, which evolves in a way that is known by all the robots. Robots use this knowledge to locally *predict* the new system state. Then, each robot takes observations on its own, which are combined classically in Information Filter form (IF), and they are used to *update* the estimated system state. Measurements can be combined through exact sums of IF data, or through distributed consensus filters. Exact sums of IF data produce estimates equivalent to the centralized system ones, provided that the network is complete [39]. General networks require additional mechanisms to ensure that robots do not sum the same piece of data more than once (problems of cyclic updates / double counting information), such as the channel filter [20, 53] for networks with a tree structure. The covariance intersection method [28] produces consistent but highly conservative estimates in general networks.

More recent distributed estimation approaches [1, 12, 41, 42] use distributed consensus filters to average the IF measurements, which automatically avoids problems of double counting information. However, these methods suffer from the delayed data problem, that takes place when the nodes execute the state prediction without having incorporated all the measurements taken at the current step, giving rise to disagreement in the robot estimates [11]. An interesting solution is given in [42] but its convergence is proved only in the absence of observation and system noises. In the algorithm proposed in [12], authors prove that the nodes' estimates are consistent, although these estimates have disagreement, i.e., that the global maps computed by different robots are not exactly the same. Other algorithms have been proposed that require the previous offline computation of the gains and weights [1]. Despite

the many advances in distributed estimation, these approaches are still limited to linear systems without inputs, and where the evolution of the system is known by all the robots. Their applicability to map merging scenarios is not straightforward, since the system models are in general nonlinear, the evolution of the system is not necessarily known by the robots, and often the robot odometry is introduced in the system as an input. Besides, map merging scenarios require data association methods for establishing correspondences between the data observed by the robots. The previous methods, however, assume that the relationship between the raw data acquired by the different sensors is known by all the sensor network.

Recently, an interesting method which allows for nonlinear systems with inputs, has been proposed in [33]. Here each robot records its own measurements and odometry, as well as the observations and odometry from any other robot it encounters. When a robot is sure it has obtained all the measurements and odometry from all the other robots up to some time instant, it can build an estimate equivalent to the centralized one. The main drawback of this approach is that robots must maintain an unbounded amount of memory, which depends on the time between robot meetings. Moreover, if a single robot fails or leaves the network, the whole system fails. Other interesting approach that allows the robots to measure both the landmarks positions as well as their own odometry, is given by [13]. Each robot has a single representation of the environment that combines its own data and the measurements of its neighbors, being this representation consistent. The main limitation of this work is that the measured information does not go beyond the neighborhood level. Thus, each robot has a better map than as if it was acting on its own. However, it does not have knowledge about the features observed by robots in farther places of the network.

Most of the previous methods have in common that they combine the data acquired by the different robots in the form of raw measurements, and that the local estimate of each robot contains information from the other robots, i.e., local estimates are not independent. Alternatively, information can be processed in the form of local maps, and these local maps can be kept independent by avoiding the introduction of global information into them; this is what we propose here, and it is also the approach followed in [14]. This strategy has the benefit that each robot can produce meaningful representations of the environment, which allows for several high level data association methods [6, 14]. Not introducing global data in the local maps, has the effect of keeping the local maps of different robots independent. Thus, consensus filters can be used without suffer-

ing from the previously mentioned problems of delayed data, and double counting information. An advantage of our approach is the natural robustness that results from its distributed implementation.

The consensus filters literature is greatly wide. A review of the most relevant results can be found in [46] and the references therein. Many recent works consider specific variations of the consensus problem to cope with communication delays [49], or stochastic communication noises [34]. Most of the works in distributed consensus address the static case, i.e., consensus is achieved on a value that depends on the initial conditions of the system. Fewer works [10, 18, 43, 45, 48, 57, 59] consider the dynamic case, where nodes measure a variable along time, and the goal is to track the average of this variable. In map merging scenarios, dynamic consensus strategies are more appealing, since the local maps of the robots will change, and it would be desirable to track the global merged map. Several dynamic consensus methods [18, 43, 45, 48] consider continuous-time systems, and thus they are better suited for systems based on the observation of the states of the neighbors, instead of on communicating the states (in our case, the maps). [59] uses discrete-time communication, but it considers that nodes measure a local continuous physical process. On the other hand, [10, 57] track the average of inputs that change in a discrete-time way, using discrete-time communications. Thus, they are better suited to the problem of map merging, where the local maps are modified at discrete time instances.

In this paper, we discuss distributed sensor fusion methods which are intended for independent observations acquired by several sensors along time. Instead of observations, we use the information increments of the local maps, i.e., the differences between the local maps at steps k and $k + 1$, expressed in Information Filter form, as inputs to the algorithm. As we discuss, the convergence and unbiased mean properties of the original algorithm remain valid regardless of this modification. An important property that any estimation method should have is *consistency* [13, 24–26], i.e., if the estimates at the robots are not overconfident. In this paper, we perform a novel and thorough study of the global map estimated by each robot and each step and prove that they are consistent.

3 Problem Description

We let n be the number of robots. Indices i, j refer to robots, G to the global map, and A to averaged information matrices and vectors. We use $k, k' \in \mathbb{N}$ for time steps. Constants \mathbf{szr} and \mathbf{szf} represent the size of re-

spectively a robot pose and a feature position¹. We let \mathbf{I} be the identity matrix, and $\mathbf{0}$ be a $n \times n$ matrix with all its elements equal to zero (if a subindex $n_1 \times n_2$ appears, this specifies their dimensions). Given a matrix W , $[W]_{ij}$ denotes its (i, j) entry. $W \succeq V$ (\preceq) indicates that matrix $W - V$ is positive- (negative-) semidefinite.

We consider a team of $n \in \mathbb{N}$ robots exploring an unknown environment. There are $m \in \mathbb{N}$ different static features in the environment and we let $\mathbf{x} \in \mathbb{R}^{\mathcal{M}}$ be the vector with their true positions, with $\mathcal{M} = m \text{ szf}$. Up to the time step k , the latest map of each robot i contains estimates $\hat{\mathbf{x}}_i^k \in \mathbb{R}^{\mathcal{M}_i^k}$ of the positions of the $m_i^k \leq m$ features observed by robot i , where $\mathcal{M}_i^k = m_i^k \text{ szf}$, with associated covariance matrix $\Sigma_i^k \in \mathbb{R}^{\mathcal{M}_i^k \times \mathcal{M}_i^k}$. Let $H_i^k \in \{0, 1\}^{\mathcal{M}_i^k \times \mathcal{M}}$ be the observation matrix that relates the elements in \mathbf{x} and $\hat{\mathbf{x}}_i^k$; then, the local map of each robot i contains a partial observation of \mathbf{x} ,

$$\hat{\mathbf{x}}_i^k = H_i^k \mathbf{x} + \mathbf{v}_i^k, \quad E[\mathbf{v}_i^k] = \mathbf{0}, \quad E[\mathbf{v}_i^k (\mathbf{v}_i^k)^T] = \Sigma_i^k, \quad (1)$$

where \mathbf{v}_i^k is a zero mean noise with covariance matrix Σ_i^k . Up to the time step k , the latest map of each robot i contains as well estimates $\hat{\mathbf{r}}_i^k \in \mathbb{R}^{\mathcal{R}_i^k}$ of r_i^k of the poses² of robot i , where $\mathcal{R}_i^k = r_i^k \text{ szr}$, with associated covariance matrix $R_i^k \in \mathbb{R}^{\mathcal{R}_i^k \times \mathcal{R}_i^k}$. Let $\mathbf{r}_i^k \in \mathbb{R}^{\mathcal{R}_i^k}$ be the true values for these r_i^k poses of robot i up to step k , then

$$\begin{aligned} \hat{\mathbf{r}}_i^k &= \mathbf{r}_i^k + \mathbf{w}_i^k, & E[\mathbf{w}_i^k (\mathbf{w}_i^k)^T] &= R_i^k, \\ E[\mathbf{w}_i^k] &= \mathbf{0}, & E[\mathbf{w}_i^k (\mathbf{v}_i^k)^T] &= S_i^k, \end{aligned} \quad (2)$$

where \mathbf{w}_i^k is a zero mean noise with covariance matrix R_i^k , and $S_i^k \in \mathbb{R}^{\mathcal{R}_i^k \times \mathcal{M}_i^k}$ is the cross-covariance between the estimates of the features' positions $\hat{\mathbf{x}}_i^k$ and the robot poses $\hat{\mathbf{r}}_i^k$ in eqs. (1), (2). In this paper, we do not discuss the exploration strategies or the Simultaneous Localization and Map Building (SLAM) algorithms for obtaining the local maps; any method capable of producing stochastic maps as in eqs. (1), (2) can be used. Note that the linear model in eq. (1) refers to the fact that the local maps are an estimate of the features positions; the observation model associated to the sensor used to build the local maps does not need to be linear.

If at step k the information from the n robots was available, e.g., at a central agent, then the global map containing the estimate $\hat{\mathbf{r}}_{G,1}^k, \dots, \hat{\mathbf{r}}_{G,n}^k$ of the set of poses of each robot $\mathbf{r}_1^k, \dots, \mathbf{r}_n^k$ up to step k , as well as the estimate $\hat{\mathbf{x}}_G^k$ of the positions of the static features \mathbf{x} could

¹ e.g., $\text{szr} = 3$ for planar robot poses (position (x, y) and orientation θ); $\text{szf} = 2$ or $\text{szf} = 3$ for respectively 2D or 3D environments.

² e.g., only the last pose ($r_i^k = 1$), the full robot trajectory, or a subset of the trajectory.

be obtained. The local map of each robot i at step k is a partial observation of these elements (eqs. (1), (2)),

$$\begin{bmatrix} \hat{\mathbf{r}}_i^k \\ \hat{\mathbf{x}}_i^k \end{bmatrix} = \begin{bmatrix} L_i^k & \mathbf{0} \\ \mathbf{0} & H_i^k \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^k \\ \vdots \\ \mathbf{r}_n^k \\ \mathbf{x} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{w}}_i^k \\ \hat{\mathbf{v}}_i^k \end{bmatrix},$$

$$\text{where } L_i^k = [\mathbf{0} \dots \mathbf{0}, \mathbf{I}_{\mathcal{R}_i^k}, \mathbf{0} \dots \mathbf{0}]. \quad (3)$$

We assume that the noises are independent for different robots $i \neq j$ and all $k, k' \in \mathbb{N}$, since every robot has constructed the map based on its own observations, i.e., $E[\mathbf{w}_i^k (\mathbf{w}_j^{k'})^T] = \mathbf{0}$, $E[\mathbf{v}_i^k (\mathbf{v}_j^{k'})^T] = \mathbf{0}$, and $E[\mathbf{w}_i^k (\mathbf{v}_j^{k'})^T] = \mathbf{0}$. Note that since the local map of a robot i at step k is an evolution of its map at any previous step $k' < k$, then the noises $\mathbf{w}_i^k, \mathbf{v}_i^k$, and the noises $\mathbf{w}_i^{k'}, \mathbf{v}_i^{k'}$ are not independent.

Let $Y_i^k \in \mathbb{R}^{\mathcal{M}_i^k \times \mathcal{M}_i^k}$, $\mathbf{y}_i^k \in \mathbb{R}^{\mathcal{M}_i^k}$ be the information matrix and vector of the local map at robot i and step k in IF form, for $i \in \{1, \dots, n\}$, where $\mathcal{M}_G^k = \mathcal{R}_1^k + \dots + \mathcal{R}_n^k + \mathcal{M}$,

$$\begin{aligned} Y_i^k &= \begin{bmatrix} L_i^k & \mathbf{0} \\ \mathbf{0} & H_i^k \end{bmatrix}^T \begin{bmatrix} R_i^k & S_i^k \\ (S_i^k)^T & \Sigma_i^k \end{bmatrix}^{-1} \begin{bmatrix} L_i^k & \mathbf{0} \\ \mathbf{0} & H_i^k \end{bmatrix}, \\ \mathbf{y}_i^k &= \begin{bmatrix} L_i^k & \mathbf{0} \\ \mathbf{0} & H_i^k \end{bmatrix}^T \begin{bmatrix} R_i^k & S_i^k \\ (S_i^k)^T & \Sigma_i^k \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{r}}_i^k \\ \hat{\mathbf{x}}_i^k \end{bmatrix}. \end{aligned} \quad (4)$$

The mean vector of the global map containing the estimate $\hat{\mathbf{r}}_{G,1}^k, \dots, \hat{\mathbf{r}}_{G,n}^k$ of the set of poses of each robot $\mathbf{r}_1^k, \dots, \mathbf{r}_n^k$ up to step k , as well as the estimate $\hat{\mathbf{x}}_G^k$ of the positions of the static features \mathbf{x} is given by,

$$((\hat{\mathbf{r}}_{G,1}^k)^T, \dots, (\hat{\mathbf{r}}_{G,n}^k)^T, (\hat{\mathbf{x}}_G^k)^T)^T = \left(\sum_{i=1}^n Y_i^k \right)^{-1} \sum_{i=1}^n \mathbf{y}_i^k, \quad (5)$$

where term $(\sum_{i=1}^n Y_i^k)^{-1}$ is its associated covariance matrix. Merging the maps in IF form is a common practice [50] since the operation is additive, commutative, and associative.

Note that the global map in eq. (5) is different from the one that would be obtained by a centralized multi-robot SLAM, since the local maps in eq. (4) do not include measurements from the other robots. Eq. (5) computes the minimum-variance unbiased estimate of $\mathbf{r}_1^k, \dots, \mathbf{r}_n^k, \mathbf{x}$ given the *local maps* (the maximum-likelihood estimate if the local maps are Gaussian), whereas centralized multi-robot SLAM methods estimate $\mathbf{r}_1^k, \dots, \mathbf{r}_n^k, \mathbf{x}$ given the *measurements and control inputs*. Thus, the accuracy of the global map in eq. (5) depends on the precision of the local maps. The unbiased mean and consistency properties of the global map depend on the local maps having unbiased mean and being consistent.

Since we do not include measurements from the other robots, the local maps of different robots remain independent and can be fused by the addition of the information matrices and vectors as in eq. (5).

Now consider the next time step $k + 1$. Robots have kept on exploring and some of the robot maps have changed. We denote \mathcal{T}_i the time steps at which robot i propagates its latest map to the network, i.e., if robot i decides it wants to initiate the propagation of its latest map, then $k + 1 \in \mathcal{T}_i$; otherwise, $k + 1 \notin \mathcal{T}_i$ and robot i keeps on merging the previous map. We let d_i be the degree of a robot i , containing the total number of times its local map changes (the cardinality of \mathcal{T}_i), and d be the degree of the team,

$$d_i = |\mathcal{T}_i|, \quad d = d_1 + \dots + d_n. \quad (6)$$

In this paper we consider that the number of times robots propagate the changes of their local maps d is finite. These changes give rise to a different global map (eq. (5)) and robots must update their estimates to react to this change.

Problem 1 (Dynamic Map Merging) We consider $n \in \mathbb{N}$ robots exploring and acquiring local maps at some time steps k as in eqs. (1), (2). The communication is range-limited and two robots can exchange data only if they are close enough. We let $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ be the undirected communication graph at step k . The nodes are the robots, $\mathcal{V} = \{1, \dots, n\}$. If robots i, j can communicate then there is an edge between them, $(i, j) \in \mathcal{E}_k$. The set of neighbors \mathcal{N}_i^k of robot i at step k is

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

The goal is the design of distributed algorithms so that each robot $i \in \mathcal{V}$ computes and tracks the global map in eq. (5), and the blocks in the main diagonal of its covariance matrix, based on local interactions with its neighbors \mathcal{N}_i^k . \square

4 Dynamic Map Merging Algorithm

The space-time diffusion methods have been previously used under independent observations of static variables [57]. In our map merging scenario, the map features \mathbf{x} are static but the robot poses \mathbf{r}_i^k vary with time k . Besides, the local map of a robot i at step k is an evolution of its local map at previous steps $k' < k$. Thus, the local maps Y_i^k, \mathbf{y}_i^k (eq. (4)) are not independent and this has to be taken into account, because otherwise the same information would be considered several times. For the previous reasons, we propose to use space-time diffusion ideas using as inputs the *information increments*

associated to the feature estimates instead of the maps Y_i^k, \mathbf{y}_i^k .

We first pay attention to eq. (5). Using classical matrix block-wise inversion rules [22, Chap.0.7], the global estimates $\hat{\mathbf{x}}_G^k$ of the positions of the static features \mathbf{x} in eq. (5), and its associated block $\Sigma_G^k \doteq E[\hat{\mathbf{x}}_G^k (\hat{\mathbf{x}}_G^k)^T]$ within the covariance matrix $(\sum_{i=1}^n Y_i^k)^{-1}$ are given by

$$\hat{\mathbf{x}}_G^k = (I_G^k)^{-1} \mathbf{i}_G^k, \quad \Sigma_G^k = (I_G^k)^{-1}, \quad (7)$$

where $I_G^k \in \mathbb{R}^{\mathcal{M} \times \mathcal{M}}$, $\mathbf{i}_G^k \in \mathbb{R}^{\mathcal{M}}$ are the information matrix and vector of the estimates of the features' positions \mathbf{x} in the global map at step k in IF form,

$$I_G^k = \sum_{i=1}^n I_i^k, \quad \mathbf{i}_G^k = \sum_{i=1}^n \mathbf{i}_i^k, \quad (8)$$

and $I_i^k \in \mathbb{R}^{\mathcal{M} \times \mathcal{M}}$ and $\mathbf{i}_i^k \in \mathbb{R}^{\mathcal{M}}$ are the information matrix and vector of the local estimates $\hat{\mathbf{x}}_i^k$ of the features' positions \mathbf{x} in the local map (eq. (1)) at robot $i \in \{1, \dots, n\}$ and step k in IF form,

$$I_i^k = (H_i^k)^T (\Sigma_i^k)^{-1} H_i^k, \quad \mathbf{i}_i^k = (H_i^k)^T (\Sigma_i^k)^{-1} \hat{\mathbf{x}}_i^k. \quad (9)$$

The global estimates $\hat{\mathbf{r}}_{G,i}^k$ of the set of poses \mathbf{r}_i^k of each robot i up to step k in eq. (5), and its associated block $R_{G,ii}^k \doteq E[\hat{\mathbf{r}}_{G,i}^k (\hat{\mathbf{r}}_{G,i}^k)^T]$ within the covariance matrix $(\sum_{i=1}^n Y_i^k)^{-1}$, can be obtained from $\hat{\mathbf{x}}_G^k$, Σ_G^k (eq. (7)) and from the local maps $\hat{\mathbf{r}}_i^k, \hat{\mathbf{x}}_i^k, R_i^k, S_i^k, \Sigma_i^k, H_i^k$ (eqs. (1), (2)) as follows:

$$\begin{aligned} \hat{\mathbf{r}}_{G,i}^k &= \hat{\mathbf{r}}_i^k + S_i^k (\Sigma_i^k)^{-1} (H_i^k \hat{\mathbf{x}}_G^k - \hat{\mathbf{x}}_i^k), \\ R_{G,ii}^k &= R_i^k - S_i^k (\Sigma_i^k)^{-1} (S_i^k)^T \\ &\quad + S_i^k (\Sigma_i^k)^{-1} H_i^k \Sigma_G^k (H_i^k)^T (\Sigma_i^k)^{-1} (S_i^k)^T. \end{aligned} \quad (10)$$

Here we are interested (Problem 1) in computing the blocks in the main diagonal of the covariance matrix $(\sum_{i=1}^n Y_i^k)^{-1}$. The expressions for the off-diagonal terms can be found e.g., in our recent work [5].

Thus, the original problem can be decomposed into two parts: the estimation of the features' positions (eqs. (7)–(9)), which requires the robots to reach consensus on the information matrices and vectors of the features' positions (eq. (8)); and the estimation of the robot poses (eq. (10)), that only requires information local to each robot i , and the features estimates $\hat{\mathbf{x}}_G^k, \Sigma_G^k$. We propose an algorithm that consists of keeping up-to-date estimates of the features' positions $\hat{\mathbf{x}}_G^k, \Sigma_G^k$, using dynamic average consensus on the information increments of the local information matrices associated to features' positions I_i^k, \mathbf{i}_i^k in eq. (9). At each step k , each robot i uses its most recent estimates of $\hat{\mathbf{x}}_G^k$ and Σ_G^k to obtain the estimates of its robot poses $\hat{\mathbf{r}}_{G,i}^k, R_{G,ii}^k$ (eq. (10)) and propagates this vector $\hat{\mathbf{r}}_{G,i}^k$ and the main diagonal elements of matrix $R_{G,ii}^k$ to the remaining robots in the

network. In the remaining of this section, we deeply discuss the part concerning the consensus on the information increments. We revise properties of convergence and unbiased mean. We carry out a careful study to show that the estimates are consistent.

For each robot $i \in \{1, \dots, n\}$ we define the following *increment* information matrix $\Delta_i^k \in \mathbb{R}^{\mathcal{M} \times \mathcal{M}}$ and vector $\delta_i^k \in \mathbb{R}^{\mathcal{M}}$,

$$\begin{aligned} \Delta_i^k &= I_i^k - I_i^{k-1}, & \delta_i^k &= \mathbf{i}_i^k - \mathbf{i}_i^{k-1}, \text{ for } k \geq 1, \\ \Delta_i^k &= I_i^k, & \delta_i^k &= \mathbf{i}_i^k, \text{ for } k = 0. \end{aligned} \quad (11)$$

Note that for all the robots such that $k \notin \mathcal{T}_i$, the increment information matrices and vectors will be zero. The associated features' position estimates within the global map at step k in eq. (8) can be expressed in terms of the previous global estimate at step $k-1$ and the map increments at k as follows:

$$I_G^k = I_G^{k-1} + \sum_{i=1}^n \Delta_i^k, \quad \mathbf{i}_G^k = \mathbf{i}_G^{k-1} + \sum_{i=1}^n \delta_i^k. \quad (12)$$

Equivalently, the estimates of the features' positions in the local map of each robot i at step k , and in the global map at step k can be expressed in terms of the map increments at all the previous steps $k' = 0, \dots, k$,

$$\begin{aligned} I_i^k &= \sum_{k'=0}^k \Delta_i^{k'}, & \mathbf{i}_i^k &= \sum_{k'=0}^k \delta_i^{k'}, \\ I_G^k &= \sum_{i=1}^n \sum_{k'=0}^k \Delta_i^{k'}, & \mathbf{i}_G^k &= \sum_{i=1}^n \sum_{k'=0}^k \delta_i^{k'}. \end{aligned} \quad (13)$$

Each robot i maintains an estimate of the averaged information matrix $\hat{I}_i^A(k) \in \mathbb{R}^{\mathcal{M} \times \mathcal{M}}$ and vector $\hat{\mathbf{i}}_i^A(k) \in \mathbb{R}^{\mathcal{M}}$, and of its degree $d_i(k)$ containing the number of times it has updated its local map; recall that each robot i propagates the changes in its local map at specific and locally decided time steps $k \in \mathcal{T}_i$. Robot $i \in \{1, \dots, n\}$ initializes its variables with

$$d_i(-1) = 0, \quad \hat{I}_i^A(0) = \mathbf{0}, \quad \hat{\mathbf{i}}_i^A(0) = \mathbf{0}, \quad (14)$$

and updates them at all $k \geq 0$ with the following algorithm.

Algorithm 1 (Dynamic map merging - robot i , iteration k)

(*Measurement update:*)

If $k \in \mathcal{T}_i$, $d_i(k) = d_i(k-1) + 1$,

$$\begin{aligned} \hat{I}_i^A(k_+) &= (1 - 1/d_i(k))\hat{I}_i^A(k) + \Delta_i^k/d_i(k), \\ \hat{\mathbf{i}}_i^A(k_+) &= (1 - 1/d_i(k))\hat{\mathbf{i}}_i^A(k) + \delta_i^k/d_i(k); \end{aligned} \quad (15)$$

otherwise, $d_i(k) = d_i(k-1)$,

$$\hat{I}_i^A(k_+) = \hat{I}_i^A(k), \quad \hat{\mathbf{i}}_i^A(k_+) = \hat{\mathbf{i}}_i^A(k). \quad (16)$$

(*Spatial update:*) If $d_i(k) > 0$,

$$\begin{aligned} \hat{I}_i^A(k+1) &= \sum_{j \in \mathcal{N}_i^k \cup \{i\}} \mathcal{W}_{ij}(k) \hat{I}_j^A(k_+), \\ \hat{\mathbf{i}}_i^A(k+1) &= \sum_{j \in \mathcal{N}_i^k \cup \{i\}} \mathcal{W}_{ij}(k) \hat{\mathbf{i}}_j^A(k_+), \end{aligned} \quad (17)$$

where the space-time weight matrix $\mathcal{W}(k) \in \mathbb{R}^{n \times n}$ is

$$\begin{aligned} \mathcal{W}_{ii}(k) &= 1 - \sum_{j \in \mathcal{N}_i^k} \mathcal{W}_{ij}(k), \text{ and for } j \neq i \\ \mathcal{W}_{ij}(k) &= d_j(k) / \max\{d_i^{st}(k), d_j^{st}(k)\}, \text{ if } (i, j) \in \mathcal{E}_k, \\ \mathcal{W}_{ij}(k) &= 0 \text{ otherwise,} \end{aligned} \quad (18)$$

and where $d_i^{st}(k)$ is the space-time degree of each robot i at step k , containing the number of map changes propagated by both robot i and its neighbors \mathcal{N}_i^k up to step k , $d_i^{st}(k) = \sum_{j \in \mathcal{N}_i^k \cup \{i\}} d_j(k)$. \square

Robots decide on their own when they want to execute a new measurement update step. If up to step k a robot i never tried to merge its map with the other robots, then $d_i(k) = 0$, and thus it does not execute the spatial update. We consider these robots i as disconnected from the others ($(i, j) \notin \mathcal{E}_k$ for all $j \neq i$), even if they are in communication range.

The superscript A in the variables of the previous algorithm refers to the fact that the matrices and vectors estimated by the robots track the *average* of the information increments, instead of its sum. In several places in this paper, we will refer instead to the *global* estimates of a robot i , which are obtained from the averaged variables $\hat{I}_i^A(k), \hat{\mathbf{i}}_i^A(k)$ as follows,

$$\begin{aligned} \hat{I}_i^G(k) &= d(k) \hat{I}_i^A(k), & \hat{\mathbf{x}}_i^G(k) &= \left(\hat{I}_i^A(k)\right)^{-1} \hat{\mathbf{i}}_i^A(k), \\ \hat{\mathbf{i}}_i^G(k) &= d(k) \hat{\mathbf{i}}_i^A(k), & \hat{\Sigma}_i^G(k) &= \left(\hat{I}_i^A(k)\right)^{-1} / d(k), \end{aligned} \quad (19)$$

where $d(k) = d_1(k) + \dots + d_n(k)$ is the degree of the robot team containing the number of times robots propagated changes in their local maps, up to step k . Note that in order to obtain the global map estimate in Information Filter form $\hat{I}_i^G(k), \hat{\mathbf{i}}_i^G(k)$, or to compute the covariance matrix $\hat{\Sigma}_i^G(k)$, robots need to estimate in parallel the total amount of measurement update steps $d(k)$. This can be done, for instance, using similar techniques as for estimating the number of nodes [32, 54]. Later in this section (Theorem 1) we provide an expression $(\hat{I}_i^A(k+1))^{-1}/d_i(k)$ for the covariance that ensures it remains consistent. When robots compute the unbiased mean $\hat{\mathbf{x}}_i^G(k)$ (eq. (19)) and consistent covariance $(\hat{I}_i^A(k+1))^{-1}/d_i(k)$ (eq. (26)) form, they do not need to know $d(k)$, but only $d_i(k)$ which is local to each robot.

4.1 Initial correspondence and data association

The expressions in eqs. (4), (7)-(10) implicitly assume that the local maps are expressed in a common reference frame. This issue is related to initial correspondence or map alignment problems. The robots usually start their operation at unknown poses and, before merging their maps, they must agree on a common reference frame. This common frame needs to be computed at least once, and usually only requires the robots to know the relative pose of its nearby teammates, see e.g., [21, 51, 58] where different methods for computing robot-to-robot transformations are presented. There exist several distributed algorithms that combine these measurements to produce the common frame, e.g., [2, 17, 30, 31, 38] and references therein.

Equivalently, for simplicity, we have presented the formulation in Sections 3 and 4 including the structures of the information matrices and vectors $\hat{\mathbf{i}}_i^A(k)$, $\hat{I}_i^A(k)$, as if robots knew the total amount of features m and the relationship between their local features and the global ones, encoded in the observation matrices H_i^k in eq. (9). The problem of establishing a relationship between the elements observed by the different robots is known as data association, and it has been investigated in the context of distributed map merging [3, 6, 14, 27, 37] and multi-robot target tracking [29, 47, 52]. First, local matches are established between the variables of neighboring sensors; after that, exclusive variables are identified without requiring any extra efforts: they are variables that have not been associated to any other one. In practice, our robots execute the distributed data association method in [3, 6] for feature-based maps. Robots discover the features observed by the others in the messages exchanged at each iteration, and introduce new columns and rows in $\hat{\mathbf{i}}_i^A(k)$, $\hat{I}_i^A(k)$ accordingly. As a result, the information matrices and vectors do not contain non-informative zero rows and columns. Information matrices $\hat{I}_i^A(k)$ (eq. (19)) are invertible at each iteration of the algorithm and thus the global map can always be estimated. Note also that the total number of features m is used only as a tool for presenting the formulation, but it does not need to be known by the robots or even to be fixed. Instead, the variables managed by the robots $\hat{\mathbf{i}}_i^A(k)$, $\hat{I}_i^A(k)$ have a structure that is adapted according to the features observed by the robot team.

4.2 Properties of the Dynamic Map Merging Algorithm

We first discuss properties of convergence and unbiased mean to check that our method performs correctly.

Lemma 1 (Convergence) *Assume all the robots $i \in \mathcal{V}$ execute the dynamic map merging algorithm (Algorithm 1), and assume that the set of communication graphs that occur infinitely often is jointly connected. Let $k_\star \geq \max\{k \in \mathcal{T}_i\}$ for all $i \in \mathcal{V}$ be a step when all map updates have been propagated by the robots, and $\hat{\mathbf{x}}_G^{k_\star}$, $\Sigma_G^{k_\star}$ be the centralized global estimate of the features' positions at this step, ($I_G^{k_\star}$, $\mathbf{i}_G^{k_\star}$ in IF form), given by eqs. (5), (7)-(9). Then, the estimated information matrix $\hat{I}_i^G(k)$, information vector $\hat{\mathbf{i}}_i^G(k)$, mean $\hat{\mathbf{x}}_i^G(k)$, and covariance $\hat{\Sigma}_i^G(k)$ as in eq. (19), at each robot $i \in \mathcal{V}$ asymptotically converge to this global estimate,*

$$\begin{aligned} \lim_{k \rightarrow \infty} \hat{I}_i^G(k) &= I_G^{k_\star}, & \lim_{k \rightarrow \infty} \hat{\mathbf{x}}_i^G(k) &= \hat{\mathbf{x}}_G^{k_\star}, \\ \lim_{k \rightarrow \infty} \hat{\mathbf{i}}_i^G(k) &= \mathbf{i}_G^{k_\star}, & \lim_{t \rightarrow \infty} \hat{\Sigma}_i^G(k) &= \Sigma_G^{k_\star}. \end{aligned} \quad (20)$$

Proof As it is stated by [57, Th. 2], if the set of communication graphs \mathcal{G}_k that occur infinitely often is jointly connected, then

$$\lim_{k \rightarrow \infty} \hat{I}_i^A(k) = \sum_{i=1}^n \sum_{k' \in \mathcal{T}_i} \frac{\Delta_i^{k'}}{d}, \quad \lim_{k \rightarrow \infty} \hat{\mathbf{i}}_i^A(k) = \sum_{i=1}^n \sum_{k' \in \mathcal{T}_i} \frac{\delta_i^{k'}}{d},$$

which equals $I_G^k/d(k)$, $\mathbf{i}_G^k/d(k)$ in eq. (8) when all measurements have been taken, i.e., when $k_\star \geq \max\{k \in \mathcal{T}_i\}$ for all $i \in \mathcal{V}$. \square

Now, we present a more compact expression for eqs. (15)-(17) in Algorithm 1 which will simplify the analysis of the remaining properties:

$$\begin{aligned} d_i(k) \hat{I}_i^A(k+1) &= \sum_{j=1}^n \mathcal{W}(k)_{ji} d_j(k-1) \hat{I}_j^A(k) \\ &+ \sum_{j=1}^n \mathcal{W}(k)_{ji} (d_j(k) - d_j(k-1)) \Delta_j^k, \end{aligned} \quad (21)$$

Moreover, since $\hat{I}_i^A(0) = \mathbf{0}$, then

$$d_i(k) \hat{I}_i^A(k+1) = \sum_{k'=0}^k \sum_{j=1}^n [\Phi(k, k')]_{ij} \Delta_j^{k'}, \quad (22)$$

where matrix $\Phi(k, k')$, with $k' \leq k$, is

$$\Phi(k, k') = \mathcal{W}(k)^T \dots \mathcal{W}(k')^T (D(k') - D(k' - 1)), \quad (23)$$

and $D(k) \in \mathbb{R}^{n \times n}$ is a diagonal matrix; each entry of its main diagonal $D_{ii}(k)$ equals the degree $d_i(k)$ of robot i at step k . The equivalent expressions for $\hat{\mathbf{i}}_i^A(k+1)$ are got by replacing $\hat{I}_j^A(k)$, Δ_j^k with $\hat{\mathbf{i}}_j^A(k)$, δ_j^k .

Lemma 2 (Unbiased mean) *The estimates of the features' positions in the global map mean $\hat{\mathbf{x}}_i^G(k)$, for each robot $i \in \mathcal{V}$, after k iterations of Algorithm 1, such that $d_i(k-1) > 0$, are unbiased estimates of the true feature positions \mathbf{x} ,*

$$\mathbb{E} [\hat{\mathbf{x}}_i^G(k)] = \mathbb{E} \left[\left(\hat{I}_i^A(k) \right)^{-1} \hat{\mathbf{i}}_i^A(k) \right] = \mathbf{x}. \quad (24)$$

Proof It can be done in a similar fashion as in [57] by noting that the local features' positions estimates $\hat{\mathbf{x}}_j^k$ at each robot j (eq. (1)) are an observation of the true \mathbf{x} ,

$$\hat{\mathbf{x}}_j^k = H_i^k \mathbf{x}_G^k + \mathbf{v}_j^k, \quad \text{with} \quad \mathbb{E} [\mathbf{v}_j^k] = 0,$$

and the increment information vector $\delta_j^k = \mathbf{i}_j^k - \mathbf{i}_j^{k-1}$ is

$$\delta_j^k = (H_j^k)^T (\Sigma_j^k)^{-1} \mathbf{v}_j^k - (H_j^{k-1})^T (\Sigma_j^{k-1})^{-1} \mathbf{v}_j^{k-1} + \Delta_j^k \mathbf{x},$$

which combined with eq. (22) gives

$$\hat{\mathbf{i}}_i^A(k) = \hat{I}_i^A(k) \mathbf{x} + \frac{1}{d_i(k-1)} \sum_{k'=1}^{k-1} \sum_{j=1}^n [\Phi(k-1, k')]_{ij} \left((H_j^{k'})^T (\Sigma_j^{k'})^{-1} \mathbf{v}_j^{k'} - (H_j^{k'-1})^T (\Sigma_j^{k'-1})^{-1} \mathbf{v}_j^{k'-1} \right),$$

and thus $\mathbb{E}[(\hat{I}_i^A(k))^{-1} \hat{\mathbf{i}}_i^A(k)] = \mathbf{x}$ since the noises $\mathbf{v}_j^{k'}$ have zero mean for all k and all $j \in \mathcal{V}$. \square

Next we present our main result, regarding the consistency of the maps estimated by the robots, at each iteration. This property is of high interest in map merging scenarios. This means that at each step k , robots have indeed a map that they can use. As a result, the robots do not need to wait for any specific number of iterations of the map merging algorithm. Instead, they can make decisions on their temporal global map estimates whenever they need. Our result relies on condition $I_j^{k+1} \succeq I_j^k$, which means that the local estimates of the features' positions at successive steps have more information, or equivalently, that they become more precise. Note that this is the behavior expected in classical SLAM approaches [15] as more observations are taken, and in our experiments it has been always observed. There is an additional condition, $d_i(k) > 0$; recall that $d_i(k) = 0$ means that robot i has not initiated the map merging process yet. Since robot i has not computed any covariance yet, it does not make sense to question whether its covariance is consistent or not.

Theorem 1 (Consistent covariance) *Assume that the local map at each robot j satisfies, for successive steps $k, k+1$,*

$$I_j^{k+1} \succeq I_j^k, \quad (25)$$

Then, the covariance $(\hat{I}_i^A(k+1))^{-1}/d_i(k)$ estimated by each robot i for which $d_i(k) > 0$, at each iteration k , is consistent with respect to the centralized covariance matrix Σ_G^k ,

$$(\hat{I}_i^A(k+1))^{-1}/d_i(k) \succeq \Sigma_G^k. \quad (26)$$

Proof Along this proof we use the following change of variables; we let $\hat{J}_i^A(k)$ be

$$\hat{J}_i^A(k) = d_i(k-1) \hat{I}_i^A(k), \quad (27)$$

and note that if $d_i(k-1) = 0$, then $\hat{J}_i^A(k) = \mathbf{0}$. From eq. (21), this variable evolves according to

$$\begin{aligned} \hat{J}_i^A(k+1) &= \sum_{j=1}^n \mathcal{W}_{ji}(k) \hat{J}_j^A(k) \\ &+ \sum_{j=1}^n \mathcal{W}_{ji}(k) (d_j(k) - d_j(k-1)) (I_j^k - I_j^{k-1}), \end{aligned} \quad (28)$$

where $d_j(k) - d_j(k-1) = 1$ if robot j introduced a new map increment during the last step k , and zero otherwise, and $\mathcal{W}(k)_{ji}$, is given by eq. (18). Note that the entries of matrix $\mathcal{W}(k)$ are numbers between 0 and 1, and recall that $\mathcal{W}_{ji}(k) = 0$ if $d_i(k) = 0$ or $d_j(k) = 0$.

We want to prove that, for all i and k ,

$$\hat{J}_i^A(k+1) \preceq \sum_{j=1}^n I_j^k = I_G^k; \quad (29)$$

this is done by induction. We consider first that case $k=0$, where the robots states $\hat{J}_j^A(k)$ are initialized with zeros, where $d_j(-1) = 0$, and where the map increments are exactly the maps at $k=0$, since $I_j^{k-1} = \mathbf{0}$ for $k=0$; we have that for all i ,

$$\hat{J}_i^A(1) = \sum_{j=1}^n \mathcal{W}_{ji}(0) d_j(0) I_j^0. \quad (30)$$

Since the weights $\mathcal{W}_{ji}(k)$ are numbers between 0 and 1, the degrees $d_j(0)$ are equal to 0 or to 1, and the local information matrices I_j^k are positive semidefinite, $I_j^k \succeq 0$, then we have $\mathcal{W}_{ji}(k) d_j(k) I_j^k \preceq I_j^k$, and thus

$$\hat{J}_i^A(1) \preceq \sum_{j=1}^n I_j^0 = I_G^0. \quad (31)$$

Now that we have proved that it is true for $k=0$, we assume it is true for k , i.e., $\hat{J}_j^A(k) \preceq I_G^{k-1} = \sum_{j'=1}^n I_{j'}^{k-1}$ for all j , and we try to prove that then it holds for $k+1$

as well. Considering eq. (28), and taking into account that the weights satisfy $\sum_{j=1}^n \mathcal{W}_{ji}(k) = 1$, we have

$$\begin{aligned} \hat{J}_i^A(k+1) &= \sum_{j=1}^n \mathcal{W}_{ji}(k) \hat{J}_j^A(k) \\ &+ \sum_{j=1}^n \mathcal{W}_{ji}(k) (d_j(k) - d_j(k-1)) (I_j^k - I_j^{k-1}) \\ &\preceq \sum_{j=1}^n \mathcal{W}_{ji}(k) \left(\sum_{j'=1}^n I_{j'}^{k-1} \right) + \sum_{j=1}^n \mathcal{W}_{ji}(k) (I_j^k - I_j^{k-1}) \\ &= \sum_{j'=1}^n I_{j'}^{k-1} + \sum_{j=1}^n \mathcal{W}_{ji}(k) (I_j^k - I_j^{k-1}). \end{aligned} \quad (32)$$

From condition (25), $I_i^k - I_i^{k-1} \succeq \mathbf{0}$, and thus, using again the fact that $\mathcal{W}_{ji}(k)$ are positive numbers between 0 and 1, we have

$$\hat{J}_i^A(k+1) \preceq \sum_{j=1}^n I_j^{k-1} + \sum_{j=1}^n (I_j^k - I_j^{k-1}) = \sum_{j=1}^n I_j^k = I_G^k, \quad (33)$$

concluding that $\hat{J}_i^A(k+1) \preceq I_G^k$. Thus, when $d_i(k) > 0$,

$$\frac{(\hat{I}_i^A(k+1))^{-1}}{d_i(k)} = (\hat{J}_i^A(k+1))^{-1} \succeq (I_G^k)^{-1} = \Sigma_G^k, \quad (34)$$

which concludes the proof. \square

Note that the results about the estimated merged maps being unbiased and consistent (Lemma 2 and Theorem 1) rely on the local maps being consistent as in eqs. (1) and (2). Depending on the sensing model, e.g., if robots can only obtain partial observations of the features positions, and depending on the local mapping method used, the local maps may not be consistent. Even in this case, the global maps estimated by our algorithm are more conservative than the centralized map.

We finally note that the estimates of the robot poses $\hat{\mathbf{r}}_{G,i}^k$, $R_{G,ii}^k$ are obtained by each robot i by replacing $\hat{\mathbf{x}}_G^k$ and Σ_G^k in eq. (10) with its most recent estimates of the features' positions. It can be easily checked that by using $\hat{\mathbf{x}}_i^G(k)$, $\hat{\Sigma}_i^G(k)$ (eq. (19)), the estimates of $\hat{\mathbf{r}}_{G,i}^k$, $R_{G,ii}^k$ are convergent as in Lemma 1; and by using $\hat{\mathbf{x}}_i^G(k)$ and the expression for the consistent covariance $(\hat{I}_i^A(k+1))^{-1}/d_i(k)$ the estimates of $\hat{\mathbf{r}}_{G,i}^k$, $R_{G,ii}^k$ are unbiased and consistent, as in Lemma 2 and Theorem 1.

4.3 Communication and Memory Costs

Now we discuss what are the benefits of using consensus-based approaches instead of classical propagation methods, in terms of communication and memory costs.

Several distributed map merging methods rely on propagating local data whenever this data changes, e.g., raw data, or local map representations. The ones based on raw (not processed) data, have several inconveniences, and they usually present large memory and communication costs. The ones that propagate local maps seem appealing from the communication point of view, since each piece of data traverses the network only once, whereas consensus-based methods transmit information at each iteration. However, methods based on propagating local maps have the inconvenience that, in addition to the global map, each robot must store the local map of every other robot in the network. Note that we are considering scenarios where the communication network can get disconnected at any moment, and individual or small groups of robots can leave the remaining team for long periods of time. In order to properly re-synchronize with them in posterior meetings, and correctly replace the old information in the global map, robots must keep track of all the information (local maps) available. Thus, the memory cost is $\sum_{i=1}^n ((\mathcal{R}_i^k + \mathcal{M})^2 + (\mathcal{R}_i^k + \mathcal{M}))$ for storing either the n information matrices and vectors, or the n mean vectors and covariance matrices³, plus $(\mathcal{M}_G^k)^2 + \mathcal{M}_G^k$ for the global map. The memory cost does not scale well with the size of the network, i.e., if the number of robots is increased without changing the scene size, the memory cost increases as well. Consensus-based approaches do not suffer from this problem, since each robot keeps a single representation of the scene, and thus the memory cost does not depend on the number of nodes.

Similarly to the memory cost discussion, in consensus-based approaches, robots send their single representation of the scene at each iteration, so that the communication cost per iteration exclusively depends on the size of the scene, and it is almost equal for all the robots. However, propagation methods do not have any control about the amount of new information that arrives to a particular robot; thus, they are prone to generate high communications peaks and bottlenecks in some areas of the network. The communication load is not properly balanced, so some particular robots may be sending large amounts of data. Due to the iterative nature of consensus methods, the total final communication cost may be larger than for other approaches depending on the number of iterations executed by the robots. This convergence speed depends on the network topology and it is related to the algebraic connectivity of

³ $(\mathcal{R}_i^k + \mathcal{M})^2$ is a worst case cost for the information matrices; in practical applications, a better performance can be achieved by taking advantage of their sparse structure. E.g., for full robot trajectories approaches, it can be order $(\mathcal{M} + (l+1)\mathcal{R}_i^k)$, where l is the average number of features observed from each robot pose.

the communication graph, as discussed in [3, 4]. There exist several methods for estimating this algebraic connectivity, e.g., [8]. The number of iterations can also be easily optimized in a local way, by executing a new consensus iteration only if the neighborhood has changed, or if there have been great modifications in the state of some of the robots in the neighborhood.

Thus, using consensus strategies is a more efficient choice whenever there is common information that was observed by several robots, whereas propagation methods make sense when there is no overlapping in the features observed by the robot team. Our method combines the benefits of both approaches: consensus is executed to estimate the feature positions $\hat{\mathbf{x}}_G^k$, Σ_G^k , with memory cost $(\mathcal{M})^2 + \mathcal{M}$ and communication cost per iteration $(\mathcal{M})^2 + \mathcal{M}$; and each robot i locally estimates its poses $\hat{\mathbf{r}}_{G,i}^k$, $R_{G,ii}^k$ (eq. (10)) and propagates vector $\hat{\mathbf{r}}_{G,i}^k$ and the main diagonal of $R_{G,ii}^k$. Thus, the memory cost per robot for storing the global map is $(\sum_{i=1}^n \mathcal{R}_i^k) + (\mathcal{M})^2 + \mathcal{M}_G^k$, and there is no need to keep any additional information from the other robots. The communication cost associated to the propagation (vector $\hat{\mathbf{r}}_{G,i}^k$ and the main diagonal of $R_{G,ii}^k$) is light, since these elements are vectors. Moreover, in practice, our robots execute the algorithm described in this paper for estimating the global mean $\hat{\mathbf{x}}_G^k$ and covariance Σ_G^k of the *common* features, i.e., using the information increments of the features that appear in several local maps. In addition, a robot i may have been the only one that has observed some exclusive features. These exclusive features are managed in the same fashion as for the estimated robot poses, i.e., they are re-estimated and its mean and the main diagonal entries of their covariance matrix are propagated. As a result, all the robots have the information of the exclusive features of the other robots. Thus, the size \mathcal{M} used in this paragraph refers to the number of common features, and the sizes \mathcal{R}_i^k to the number of poses and exclusive features at robot i . Equivalently, the computational cost of our method, which is cubic on the size \mathcal{M} (eq. (19)), refers to the number of common features as well.

5 Experiments

5.1 Experiments with RGB-D data

We have performed experiments using RGB-D sensors (Figure 1 (a)), which provide both regular RGB (Figure 1 (b)) and depth image information (Figure 1 (c)). Thus, it is possible to compute the cloud of points in 3D from a single image (Figure 1 (d)). We consider a robot team composed by 9 robots that acquire infor-

mation with RGB-D sensors, and that extract SIFT features [35] from the images. The robots take 473 images in total, and from each image around 1333 SIFT points are extracted. Each robot uses a standard EKF SLAM method for computing the local maps, taking as features SIFT points. It processes the measurements (image coordinates and depth data) and builds its local map, composed of the estimates of the 3D position of the SIFT features, as well as the estimate of its own last 3D position and orientation. The robot motions are estimated by comparing the RGB-D + SIFT point clouds obtained in the previous and current steps. Candidate relative translations and rotations are obtained from three matched points, and are voted with a RANSAC method. We solve both initialization and data association in a centralized fashion, using the same method (obtaining the most voted relative translations and rotations between RGB-D + SIFT point clouds). Note that the observation model consisting of the image coordinates and the depth of the SIFT features is not linear. However, the local maps build by the robots contain an estimate of the Cartesian coordinates of the features as in eq. (1).

We illustrate the behavior of our algorithm in the following scenario: four of the robots ($R3$, $R5$, $R7$, $R9$) have already finished their exploration when the merging process begins; they provide their local maps at the step $k = 0$ and remain static during the execution of the algorithm. Robots $R2$, $R6$, $R8$ on the other hand, keep on moving and updating their local maps, simultaneously to the merging process. Finally, robots $R1$ and $R4$ explore and update their maps as well, but they form a different exploration cluster and remain disconnected from the team for several steps. A summary of the time steps when robots propagated their local maps in our experiment can be seen in Table 1. The local maps of the robots contain around 962 features per map at the last step; the smallest and largest local maps belong to robots $R2$ and $R9$ and have respectively 163 and 2858 features.

Table 1 Steps \mathcal{T}_i at which robot i propagates its local map.

Fixed agents	Exploring agents	Other cluster
$\mathcal{T}_3 = \{0\}$	$\mathcal{T}_2 = \{0, 4, 8\}$	$\mathcal{T}_1 = \{5, 15, 25\}$
$\mathcal{T}_5 = \{0\}$	$\mathcal{T}_6 = \{0, 5, 10, 20\}$	$\mathcal{T}_4 = \{5, 15, 25\}$
$\mathcal{T}_7 = \{0\}$	$\mathcal{T}_8 = \{0, 5, 10, 20\}$	
$\mathcal{T}_9 = \{0\}$		

As robots move, the communication graph \mathcal{G}_k changes and new links appear and disappear (Figure 2); for instance, $R2$ gets isolated for some steps ($k = 6$); $R1$ and $R4$ remain isolated from the others ($k = 0$, $k = 6$) until step $k = 33$; and the neighbors of all the robots

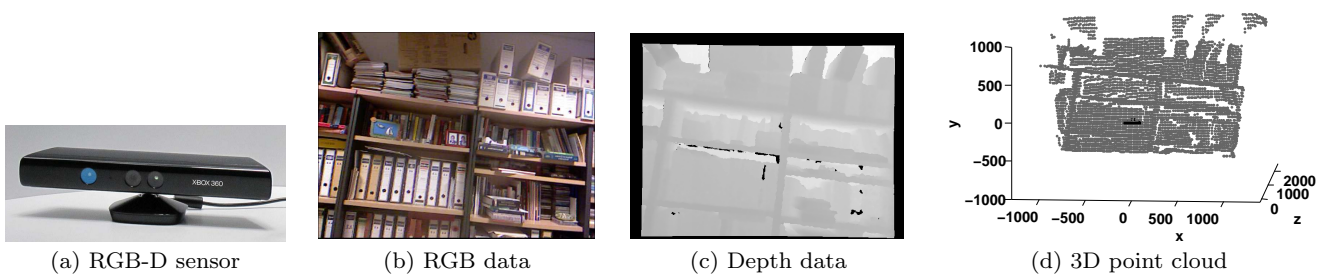


Fig. 1 Data given by a RGB-D sensor. x -, y - and z - axes in (d) are in millimeters.

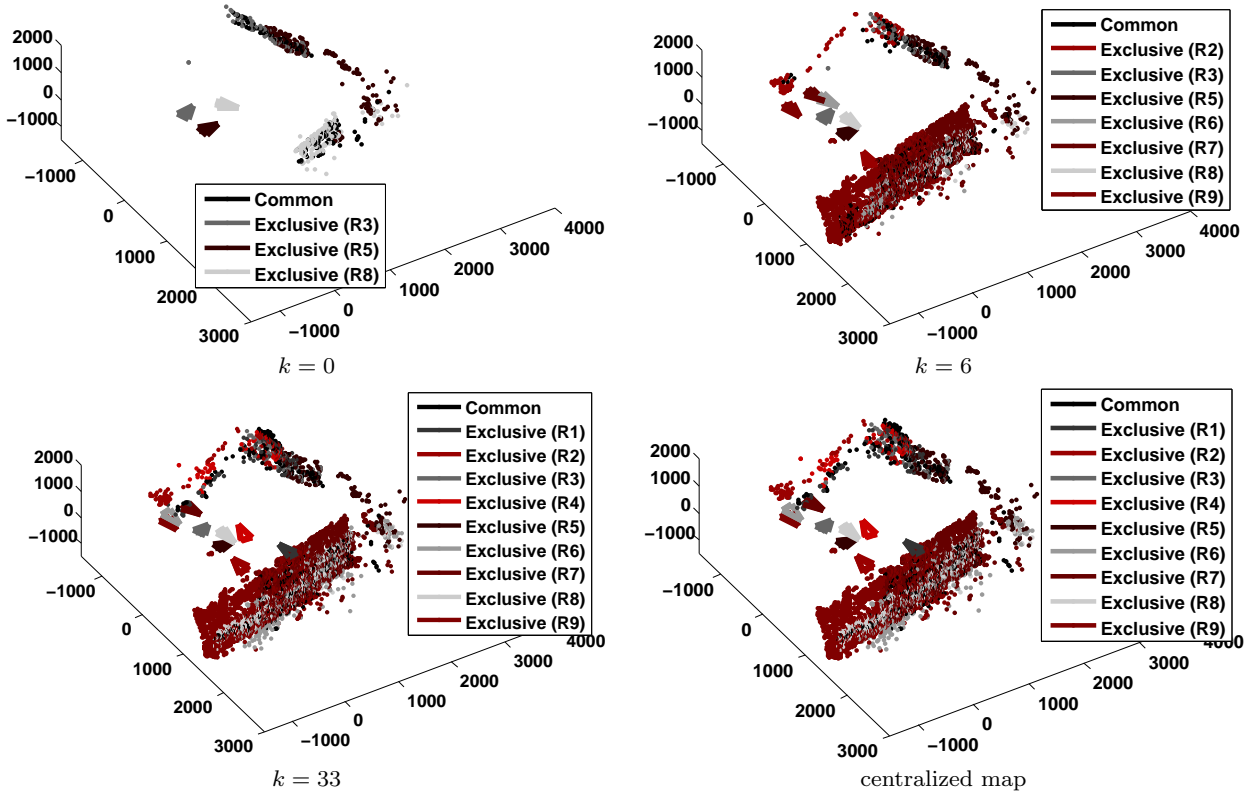


Fig. 3 Global map estimated by robot $R8$ at different steps. We display in different colors the features that originally belonged to the local maps of different robots. The common features belong to several different local maps. Feature covariances have been omitted for clarity. The global map estimated by $R8$ at step $k = 33$ is already very similar to the one that would be obtained by a centralized system. x -, y - and z - axes in are in millimeters.

change several times ($k = 0$ to $k = 40$). Note that in none step it is a complete (all-to-all) graph. We show the global map estimated by robot $R8$ with the proposed map merging algorithm (Figure 3). At $k = 0$ it only contains information from its immediate neighbors; at successive steps, this global map contains data from more distant robots ($k = 6$); at step $k = 33$, robot $R1$ establishes communication with the robot team for the first time and it sends them the global map associated to the cluster of $R1$ and $R4$. Thus, $R8$ finally has information from all the robot teams, and obtains a global map estimate that contains 7670 features, and

that is very similar to the one that would be obtained by merging the maps in a centralized fashion.

We show the evolution of the covariances and mean vectors, and information matrices and vectors of the global map estimated by the robots (Figure 4). We illustrate it using the x -coordinate of a feature $F_{2,31}$ which was observed by robots in the cluster ($R1, R4$), and in the remaining team ($R2, R3, R5, R6, R7, R8, R9$). At each step, we display (blue solid) the estimate that would be obtained by a centralized system (eq.(5)) considering all the robot local maps. Note that the centralized estimates change whenever a robot propagates changes of its local map (Table 1). The mean $\hat{\mathbf{x}}_i^G(k)$,

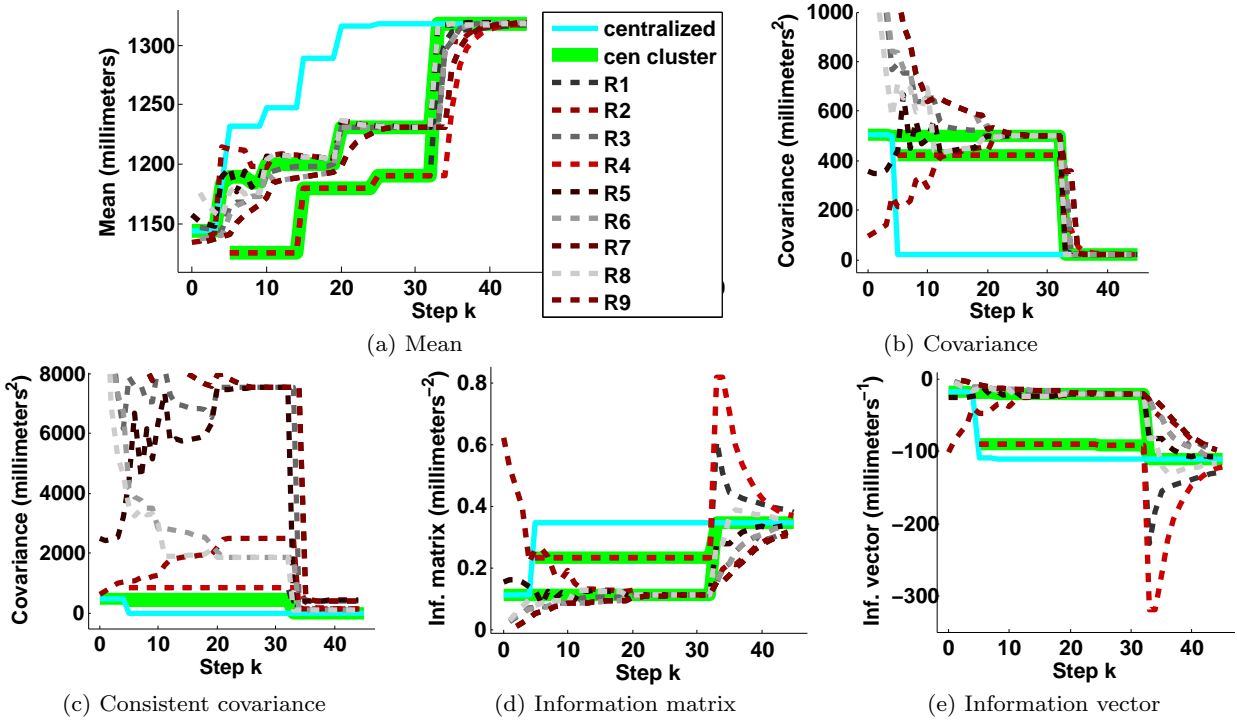


Fig. 4 Robots execute the algorithm for fusing their maps for 45 iterations k (x -axis). We show the evolution of the estimates at each robot (different colors, dashed) of: (a) the mean vector $\hat{\mathbf{x}}_i^G(k)$; (b) the covariance matrix $\hat{\Sigma}_i^G(k)$; (c) the consistent expression for the covariance matrix, $(\hat{I}_i^A(k+1))^{-1}/d_i(k)$ (Theorem 1); (d) the information matrix $\hat{I}_i^G(k)$; and (e) the information vector $\hat{\mathbf{i}}_i^G(k)$. We focus on the evolution of the entry associated to the x -coordinate of feature $F_{2,31}$. We display in blue solid the value of this feature coordinate in the global map (eq. (5)). Until step $k = 33$, robots remain in two separated clusters, one of them composed by $R1, R4$, and the other by the remaining robots. We display as well (green solid) the centralized map that would be obtained by considering all the available local maps within each cluster. After step $k = 33$, both cluster global maps (green solid) become the equal to the global map (eq. (5)) that considers the local maps of all the robots (blue solid).

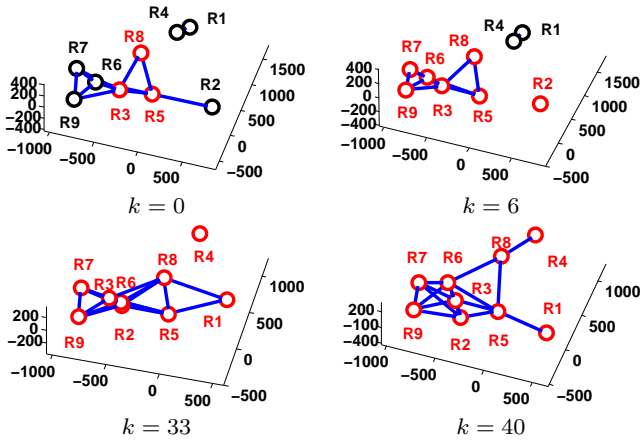


Fig. 2 Communication graphs \mathcal{G}_k at different steps k . Robot $R8$ has received information of the local maps of the robots displayed in red. x -, y - and z - axes in are in millimeters.

covariance $\hat{\Sigma}_i^G(k)$, and information matrix $\hat{I}_i^G(k)$ and vector $\hat{\mathbf{i}}_i^G(k)$ estimated by all the robots (different colors, dashed) correctly converge to the centralized value (blue solid). Note that the covariance estimates (Fig. 4 (b),

different colors, dashed) can become smaller than the global one (blue solid) for some robots and iterations, whereas the consistent expression of the covariance matrix $(\hat{I}_i^A(k+1))^{-1}/d_i(k)$ in Theorem 1 (Fig. 4 (c), different colors, dashed) remains larger than the centralized covariance (blue solid) for all robots and all steps. Since up to step $k = 33$ robots remain divided into two different clusters, we show as well (green solid) the estimate that would be obtained by a centralized system (eq.(5)), but considering *only* the robot local maps in each cluster. During the time both clusters are separated, the estimates of different robots (different colors, dashed), correctly track this cluster centralized value (green solid) that contains all the information that could be available in the best case to the robots. The robot estimates react to changes in the local maps in an appropriate way. In particular, up to iteration $k = 33$, since the cluster composed by $R1, R4$ has a complete (all-to-all) topology, their estimates are exactly equal to the cluster centralized ones (green solid).

We make an analysis of the communication and memory costs of our algorithm (Fig. 5, left column

(a), (c), (e)). These cost include both the consensus on the common features, as well as the propagation of the mean and the elements in the main diagonal of the covariance matrix for the exclusive features and robot poses (Section 4.3). These exclusive features and robot poses are re-estimated at each step based on the most recent estimates of the common features. We consider numbers encoded with single precision (4 bytes). A benefit of using a consensus-based algorithm is its low memory cost (Fig. 5 (e)) of around 45 MBytes per robot, which does not depend on the number of robots but only on the scene size. In addition, the communication cost per iteration (Figs. 5 (a), (c)) is almost the same for all the robots; observe that there are almost no differences between the average (gray solid) and maximum costs (black dashed). We have compared our performance against a method based on propagation (Fig. 5, right column (b), (d), (f)). The memory usage (Fig. 5 (f)) of the propagation method is much higher than for our method (Fig. 5 (e)). If we sum up the average communication costs per robot (Figs. 5 (a)-(d), gray solid line) for the 45 iterations (sum of the along the x-axis), we obtain a total of 234 MBytes versus the 61 MBytes used by the propagation method. This means that, due to the iterative nature of our algorithm, we obtain a total communication costs larger than for the propagation method. However, paying attention to the communication costs per robot (black dashed), in our method all the robots exchange similar amounts of data (Figs. 5 (a), (c)), whereas the propagation solution exhibits large communication cost peaks (Figs. 5 (b), (d)). If robots propagated their observations, i.e., the 3D SIFT point clouds extracted from their images, then the costs per robot up to step $k = 33$ would be 322 MBytes storage and 286 MBytes communication. Obviously, propagating the raw RGB + depth images instead is an even worse option; the costs per robot up to step $k = 33$ would be 693 MBytes storage and 616 MBytes communication. After propagating the observations, one of the robots would compute and propagate the global map, with an associated extra cost. Thus, as it can be seen, propagating measurements is not efficient, and propagating local maps is memory demanding and is prone to large peaks in the communication costs.

Note that the communication costs in Fig. 5 do not include the data association. This cost is highly dependent on the method used to match the features. A deep discussion of the performance of different matching strategies can be found in [37].

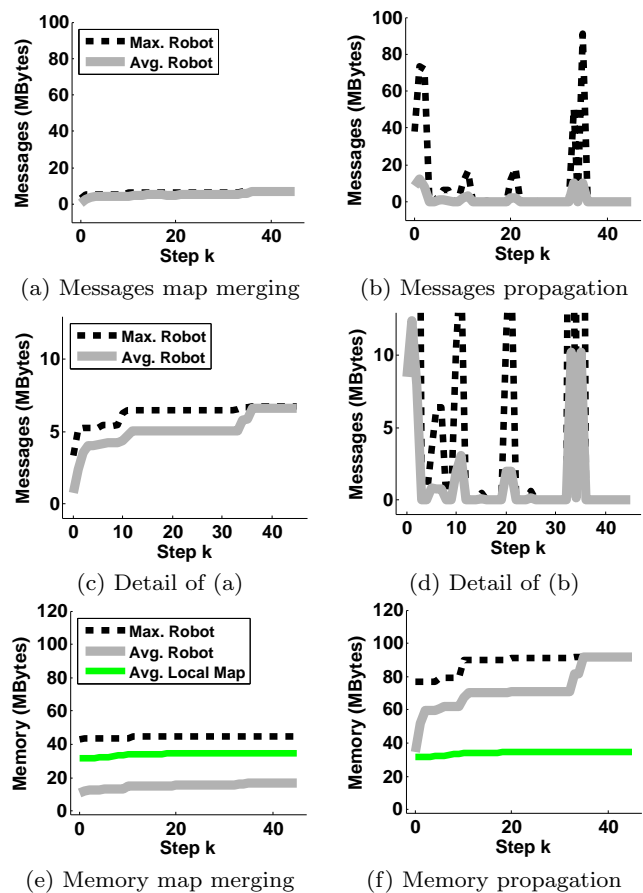


Fig. 5 Messages exchanged (a)-(d) and memory usage (e)-(f) per robot along 45 iterations of our algorithm (left column) against a simple propagation method (right column). Numbers are encoded with single precision (4 bytes). We show the average amount of information (gray solid) as well the largest amount of information per robot (black dashed). Figures (c)-(d) show a detail of (a)-(b). In figures exhibiting memory costs (e)-(f), we show the average memory used by the local maps (green solid).

5.2 Monte-Carlo simulations

We have performed Monte-Carlo simulations with 5 robots following the trajectories in Fig. 6(a). They start in the right part of the scenario and finish in the left part. We consider a 10x10x10 meters scenario with features spread over two walls and the floor. Three of the robots observe the walls and two of them the floor at different heights. Red crosses represent the ground-truth position of the features, and red triangles the ground-truth robot trajectories. Robots measure features that have a depth between 0.4 and 5 meters, and which are placed in front of them and within the image limits. These observations are corrupted with noises with standard deviation $0.0012 + 0.0019(\text{depth} - 0.4)^2$ for the depth [40], and with standard deviation 1 pixel for the image coordinates. The algorithm used for build-

ing and merging the maps is very similar to the one in the experiment with real RGB-D data, with the exception that we use the ground-truth data association for the observed features, and the ground truth initial correspondence for the robots. Robots run the method discussed in Section 4 for 50 steps. During the first 30 steps, they move and build their local maps, and simultaneously, they run the map merging method. During the last 20 steps there are no changes in the local maps, and thus, they run the map merging algorithm to agree on the latest local maps. The robots propagate the changes in their local maps after each 3 steps. Our agents exchange data if they are closer than 3.5 meters. Figure 6(a) shows the 3D features position estimated by robot 1 at the last map merging step $k = 50$, for the 100 Monte-Carlo simulations. Since the observation noise is small, the points are very accurate and similar to each other. Visually, they are almost indistinguishable.

We have studied the performance of the method using the following metrics [9]: the average root mean square error (RMS); and the average normalized state estimation error squared (NEES). For each robot i , step k , and Monte-Carlo simulation l , we let $\tilde{\mathbf{x}}_i^{G,l}(k)$ be the difference between the estimates of the common features' positions in the global map mean $\hat{\mathbf{x}}_i^G(k)$ in eq. (19) and their ground-truth position \mathbf{x} . Equivalently, we let $(\hat{I}_i^{A,l}(k+1))^{-1}/d_i(k)$ be the consistent expression of the covariance matrix of the common features, as in Theorem 1, for robot i , step k , and Monte-Carlo simulation l . Figures 6(b) and (c) show the RMS and NEES per step k computed as follows:

$$RMS = \frac{\sqrt{\sum_{l=1}^{100} \sum_{i=1}^n \frac{(\tilde{\mathbf{x}}_i^{G,l}(k))^T (\tilde{\mathbf{x}}_i^{G,l}(k))}{100n}}}{\mathcal{M}}, \quad (35)$$

$$NEES = \sum_{l=1}^{100} \sum_{i=1}^n \frac{(\tilde{\mathbf{x}}_i^{G,l}(k))^T (d_i(k) \hat{I}_i^{A,l}(k+1)) (\tilde{\mathbf{x}}_i^{G,l}(k))}{100n},$$

where \mathcal{M} is the size of common features.

Figure 6(b) shows the RMS per step (blue solid). Due to the information share, the estimated features positions become more accurate as the iterations go by, reaching estimation errors per coordinate smaller than 1 millimeter. Figure 6(c) displays the NEES value obtained (red solid), which should follow a χ^2 distribution with \mathcal{M} degrees of freedom. Thus, if the estimated merged maps are consistent, the expected value for the NEES is \mathcal{M} (black solid, *dof*), and it should not overpass the value $\chi_{0.99,dof}^2$ (black dashed). During all the steps, the estimated features' positions are consistent. This is the expected behavior for systems where robots observe the full 3D position of the features. Recall that we ensure consistency (Theorem 1) as long as the centralized map is consistent, and this depends on the local

maps being consistent. Thus, for scenarios where robots only get partial measurements of the features positions, and depending on the particular local mapping method, the local maps may not be consistent. Even in this case, our algorithm will produce estimates more conservative than the centralized map.

6 Conclusions

In this paper we have presented a method for merging feature-based maps in a dynamic way, i.e., robots compute the global map at the same time as they improve their local maps. Consensus is reached on the latest global map, using the map increments between the previous and the current time steps. Robots decide on their own when they want to propagate their local map modifications to the global map. Our method explicitly takes into account the limited communication between the robots, and it is robust to modifications in the communication topology. Robots compute the same global map that would be obtained if all the local maps were available to a centralized fusion unit. In addition, the global map estimated by each robot at each iteration is unbiased and consistent. We have demonstrated the performance of the map merging algorithm under switching topologies. Our consensus-based method outperforms map merging techniques based on propagation, since it keeps the memory cost at each robot dependent on the scene size, and not in the number of robots. However, the communication cost of our algorithm could be greatly improved by, e.g., executing a new consensus iteration in a neighborhood only if the set of neighbors change, or if their states experienced important modifications since the previous step. Future extensions of this work are in this line.

References

1. P. Alriksson and A. Rantzer. Distributed Kalman filtering using weighted averaging. In *Int. Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, July 2006.
2. R. Aragues, L. Carlone, G. Calafiore, and C. Sagues. Multi-agent localization from noisy relative pose measurements. In *IEEE Int. Conf. on Robotics and Automation*, pages 364–369, Shanghai, China, May 2011.
3. R. Aragues, J. Cortes, and C. Sagues. Distributed consensus algorithms for merging feature-based maps with limited communication. *Robotics and Autonomous Systems*, 59(3-4):163–180, 2011.
4. R. Aragues, J. Cortes, and C. Sagues. Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 28(4):840–854, 2012.

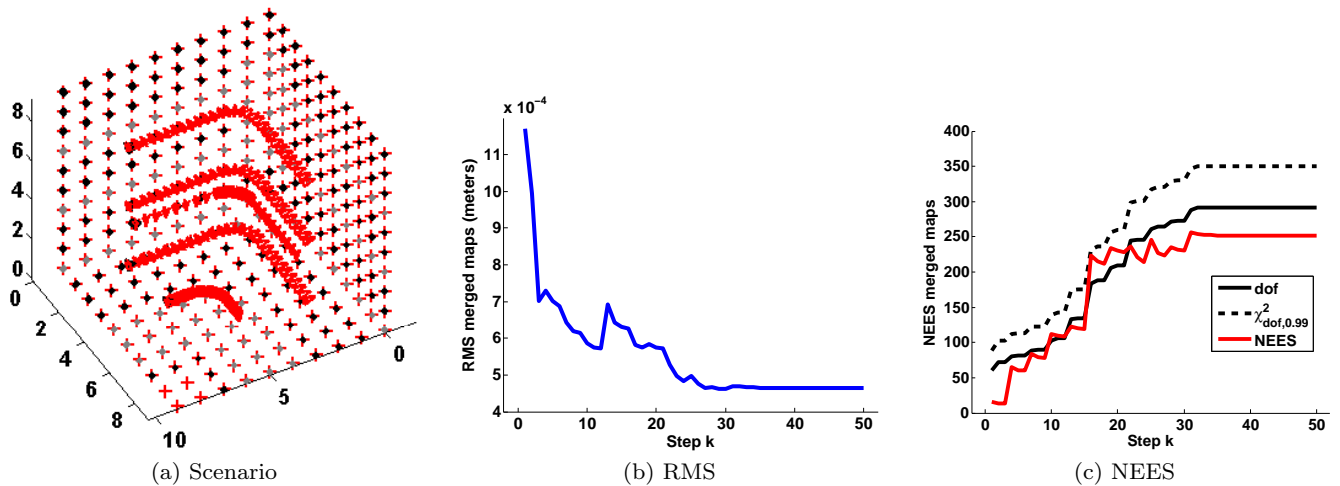


Fig. 6 (a) Scenario. 5 robots observing features; 3 robots point towards the walls, and 2 observe the floor. Red crosses are the ground-truth position of the features. Red triangles represent the ground-truth robot trajectories. Gray and black dots are the estimated position of the features observed respectively by several robots and by only one robot. For each feature, we display the 100 points obtained during the 100 Monte-Carlo simulations. (b) Average root mean square error (RMS), eq. (35). (c) Average normalized state estimation error squared (NEES), eq. (35).

5. R. Aragues, J. Cortes, and C. Sagues. Distributed map merging with consensus on the common part. In *European Control Conference*, pages 736–741, Zurich, Switzerland, July 2013.
6. R. Aragues, E. Montijano, and C. Sagues. Consistent data association in multi-robot systems with limited communications. In *Robotics: Science and Systems*, pages 97–104, Zaragoza, Spain, June 2010.
7. R. Aragues, C. Sagues, and Y. Mezouar. Feature-based map merging with dynamic consensus on information increments. In *IEEE Int. Conf. on Robotics and Automation*, pages 2710–2715, Karlsruhe, Germany, May 2013.
8. R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson. Distributed algebraic connectivity estimation for adaptive event-triggered consensus. In *American Control Conference*, pages 32–37, Montreal, Canada, June 2012.
9. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
10. G. Calafiore and F. Abrate. Distributed linear estimation over sensor networks. *International Journal of Control*, 82(5):868–882, 2009.
11. R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633, 2008.
12. D. W. Casbeer and R. Beard. Distributed information filtering using consensus filters. In *American Control Conference*, pages 1882–1887, St. Louis, USA, June 2009.
13. A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE Int. Conf. on Robotics and Automation*, pages 5220–5227, Karlsruhe, Germany, May 2013.
14. A. Cunningham, K. M. Wurm, W. Burgard, and F. Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *IEEE Int. Conf. on Robotics and Automation*, pages 1093–1100, St. Paul, USA, May 2012.
15. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
16. J. W. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion without mapping or global localization via local frontier. *Autonomous Robots*, 32(1):81–95, 2012.
17. M. Franceschelli and A. Gasparri. On agreement problems with gossip algorithms in absence of common reference frames. In *IEEE Int. Conf. on Robotics and Automation*, pages 4481–4486, Anchorage, USA, May 2010.
18. R. A. Freeman, P. Yang, and K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *IEEE Conf. on Decision and Control*, pages 398–403, San Diego, CA, USA, December 2006.
19. A. Gasparri, F. Fiorini, M. Di Rocco, and S. Panzieri. A networked transferable belief model approach for distributed data aggregation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):391–405, 2012.
20. S. Grime and H.F. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849–863, 1994.
21. J. J. Guerrero, A. C. Murillo, and C. Sagues. Localization and matching using the planar trifocal tensor with bearing-only data. *IEEE Transactions on Robotics*, 24(2):494–501, 2008.
22. R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.
23. A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, 25(12):1243–1256, 2006.
24. G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. On the consistency of multi-robot cooperative localization. In *Robotics: Science and Systems*, pages 65–72, Seattle, WA, USA, June 2009.
25. G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. Observability-based consistent ekf estimators for multi-robot cooperative localization. *Autonomous Robots*, 30(1):99–122, 2011.
26. S. Huang, Z. Wang, G. Dissanayake, and U. Frese. Iterated d-slam map joining: Evaluating its performance

- in terms of consistency, accuracy and efficiency. *Autonomous Robots*, 27(4):409–429, 2009.
27. V. Indelman, E. Nelson, N. Michael, and F. Dellaert. Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. In *IEEE Int. Conf. on Robotics and Automation*, pages 593–600, Hong Kong, China, May 2014.
 28. S. Julier and J. K. Uhlmann. General decentralised data fusion with covariance intersection (CI). In D. L. Hall and J. Llinas, editors, *Handbook of Multisensor Data Fusion*. CRC Press, 2001.
 29. A. T. Kamal, C. Ding, B. Song, J. A. Farrell, and A. K. Roy-Chowdhury. A generalized kalman consensus filter for wide-area video networks. In *IEEE Conf. on Decision and Control*, pages 7863–7869, Orlando, FL, USA, December 2011.
 30. J. Knuth and P. Barooah. Collaborative 3D localization of robots from relative pose measurements using gradient descent on manifolds. In *IEEE Int. Conf. on Robotics and Automation*, pages 1101–1106, St. Paul, MN, USA, May 2012.
 31. J. Knuth and P. Barooah. Collaborative localization with heterogeneous inter-robot measurements by riemannian optimization. In *IEEE Int. Conf. on Robotics and Automation*, pages 1526–1531, Karlsruhe, Germany, May 2013.
 32. A. Leshem and L. Tong. Estimating sensor population via probabilistic sequential polling. *IEEE Signal Processing Letters*, 12(5):395–398, 2005.
 33. K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu. Decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3554–3561, Taipei, Taiwan, October 2010.
 34. T. Li and J. F. Zhang. Consensus conditions on multi-agent systems with time-varying topologies and stochastic communication noises. *IEEE Transactions on Automatic Control*, 55(9):2043–2057, 2010.
 35. D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE Int. Conf. on Computer Vision*, pages 1150–1157, 1999.
 36. K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, 2008.
 37. E. Montijano, R. Aragues, and C. Sagues. Distributed data association in robotic networks with cameras and limited communications. *IEEE Transactions on Robotics*, 29(6):1408–1423, 2013.
 38. I. Navarro and F. Matía. Distributed orientation agreement in a group of robots. *Autonomous Robots*, 33(4):445–465, 2012.
 39. E. M. Nebot, M. Bozorg, and H. F. Durrant-Whyte. Decentralized architecture for asynchronous sensors. *Autonomous Robots*, 6(2):147–164, 1999.
 40. C.V. Nguyen, S. Izadi, and D. Lovell. Modeling Kinect sensor noise for improved 3d reconstruction and tracking. In *Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 524–530, Zurich, Switzerland, October 2012.
 41. R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *IEEE Conf. on Decision and Control*, pages 8179–8184, Sevilla, Spain, 2005.
 42. R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *IEEE Conf. on Decision and Control*, pages 5492–5498, New Orleans, LA, USA, December 2007.
 43. R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *IEEE Conf. on Decision and Control*, pages 6698–6703, Sevilla, Spain, 2005.
 44. L. M. Paz, J. D. Tardos, and J. Neira. Divide and conquer: EKF SLAM in $o(n)$. *IEEE Transactions on Robotics*, 24(5):1107–1120, 2008.
 45. W. Ren. Consensus seeking in multi-vehicle systems with a time-varying reference state. In *American Control Conference*, pages 717–722, New York, USA, 2007.
 46. W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, 2007.
 47. N. F. Sandell and R. Olfati-Saber. Distributed data association for multi-target tracking in sensor networks. In *IEEE Conf. on Decision and Control*, pages 1085–1090, Cancun, Mexico, December 2008.
 48. D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Dynamic consensus on mobile networks. In *IFAC World Congress*, Prague, Czech Republic, 2005.
 49. Y. G. Sun, L. Wang, and G. Xie. Average consensus in networks of dynamic agents with switching topologies and multiple time-varying delays. *Systems & Control Letters*, 57(2):175–183, 2008.
 50. S. Thrun, Y. Liu, D. Koller, A. Ng, and H. Durrant-Whyte. Simultaneous localisation and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7–8):693–716, 2004.
 51. N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis. Inter-robot transformations in 3-d. *IEEE Transactions on Robotics*, 26(2):226–243, 2010.
 52. N. A. Tsokas and K. J. Kyriakopoulos. Multi-robot multiple hypothesis tracking for pedestrian tracking. *Autonomous Robots*, 32(1):63–79, 2012.
 53. S. Utete and H. F. Durrant-Whyte. Routing for reliability in decentralised sensing networks. In *American Control Conference*, volume 2, pages 2268–2272, June 1994.
 54. D. Varagnolo, G. Pillonetto, and L. Schenato. Distributed statistical estimation of the number of nodes in sensor networks. In *IEEE Conf. on Decision and Control*, pages 1498–1503, Atlanta, USA, 2010.
 55. R. Vincent, D. Fox, J. Ko, K. Konolige, B. Limketkai, B. Morisset, C. Ortiz, D. Schulz, and B. Stewart. Distributed multirobot exploration, mapping, and task allocation. *Annals of Mathematics and Artificial Intelligence*, 52(1):229–255, 2008.
 56. S. B. Williams and H. Durrant-Whyte. Towards multi-vehicle simultaneous localisation and mapping. In *IEEE Int. Conf. on Robotics and Automation*, pages 2743–2748, Washington, DC, USA, May 2002.
 57. L. Xiao, S. Boyd, and S. Lall. A space-time diffusion scheme for peer-to-peer least-square estimation. In *Symposium on Information Processing of Sensor Networks (IPSN)*, pages 168–176, Nashville, TN, USA, April 2006.
 58. X.S. Zhou and S.I. Roumeliotis. Robot-to-robot relative pose estimation from range measurements. *IEEE Transactions on Robotics*, 24(6):1379–1393, 2008.
 59. M. Zhu and S. Martínez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.