2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Dynamic Consensus for Merging Visual Maps under Limited Communications

R. Aragues J. Cortes C. Sagues

Abstract— **In this paper we present an algorithm for merging visual maps in a robot network. Along the operation, each robot observes the environment and builds and maintains its local map. Simultaneously, the robots communicate and build a global map of the environment. The communication between the robots is limited, and, at every time instant, each robot can only exchange data with its neighboring robots. We provide a distributed solution to the problem which does not rely on any particular communication topology and is robust to changes in the topology. Each robot computes and tracks the global map based on local interactions with its neighbors. Our contribution is the extension of distributed sensor fusion ideas to the problem of dynamic map merging. Under mild connectivity conditions on the communication graph, this algorithm asymptotically converges to the global map. The real experiments have been carried out with visual information, which is of special interest in robotics.**

I. INTRODUCTION

Perception tasks have been long studied in the fields of localization, map building and exploration. Many existent solutions for single robot systems 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 [1] assuming that the robots broadcast their controls and their observations. The Constrained Local Submap Filter has been extended to the multi-robot case [2] assuming that each robot builds a local submap and broadcasts it, or transmits it to a central agent. Methods based on graph maps of laser scans [3], [4] make each robot build a new node and broadcast it. The same solution could be applied for many existing submap approaches [5], [6].

However, in multi robot scenarios, distributed approaches are more interesting since they explicitly consider uncomplete communication graphs, switching topologies, link failures, and limited bandwidth. These considerations are becoming popular in cooperative localization [7]. They have motivated an intensive research in distributed implementations of the Kalman Filter. Those methods rely on the information form (IF) of the Kalman filter, which is more suitable for decentralized operation. Measurement updates in IF are additive, and therefore, information coming from different sensors can be fused in any order and any time. While optimal solutions exist for complete communication networks [8], for general communication schemes [9], [10] 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. A solution that reduces this disagreement has been presented [11] 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 general robotic applications. Therefore, it does not solve its associated delayed data problem.

Distributed estimation is problematic due to the delayed data problem associated to the state prediction. However, applications that exclusively use the measurement update can be easily distributed using IF [12]. We propose a map merging solution where the information received from other robots is introduced into the global map estimates, but not into the local maps. Each robot updates its local map with measurements exclusively acquired by itself. Our map merging solution does not produce the same results than a centralized estimator compiling all measurements from all the robots. Since each robot has a lower amount of information available during its operation, the maps may be less precise than the obtained by the centralized system. However, it allows the robots explore on their own and merge their maps in a distributed fashion. In our solution, the local maps of the robots are expressed in IF form, and they are fused in an additive fashion. However, instead of simply sum them [13], we build on ideas from dynamic consensus algorithms [14], [15] to provide a distributed map merging algorithm.

In this paper we do not discuss initial correspondence or map alignment problems. We assume that the maps are aligned during the initial step using a closed form solution [16]. Other approaches that compute relative poses between the robots could be used for that [17]. We do not discuss data association as well. This problem consists of matching equal features in different maps. The widely used joint compatibility branch and bound JCBB [18] or visual appearance methods [19] can be used.

We present a dynamic version of our previous work in distributed map merging [20], where the robots performed the exploration of the environment, and only at the end of it, they ran a static consensus algorithm to merge their maps. Consequently, during the exploration, the robots did not have or incorporate any information from other agents. This paper

This work was supported by projects MEC DPI2006-07928, DPI2009-08126 and IST-1-045062-URUS-STP

R. Aragues and C. Sagues are with DIIS - I3A, University of Zaragoza, María de Luna, 50018 Zaragoza, Spain raragues@unizar.es, csagues@unizar.es

J. Cortes is with the Department of Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Dr, La Jolla, California, 92093-0411, USA cortes@ucsd.edu

instead is a proposal to dynamically merge the information online, i.e., at the same time that they are performing the exploration. This is much more realistic, because as the robots move, they acquire more information and incorporate it into their maps. However, this on-the-fly fusion is more difficult and computationally demanding.

Due to space limitations all proofs of Section III are omitted. They will appear elsewhere.

II. PRELIMINARIES

We consider a team of $n \in \mathbb{N}$ robots with limited communication capabilities. Let $\mathcal{G} = (\mathcal{R}, \mathcal{E})$ be the undirected communication graph. The nodes are the robots, $\mathcal{R} = \{1, \ldots, n\}$. If two robots i, j can exchange information, then there is an edge between them, $(i, j) \in \mathcal{E}$. Let $\mathcal{N}_i$ be the set of neighbors of robot i, $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$. Along this paper, we will use $\mathbf{1} \in \mathbb{R}^n$ for a column vector with all entries equal to 1. We let Π be the matrix $\Pi = I - \frac{\mathbf{1}\mathbf{1}^T}{n}$.

Our averaging algorithm is a discrete version of the Proportional Integral (PI) estimator in [14]. Let us consider that each node $i \in \{1, \ldots, n\}$ has an input $u_i \in \mathbb{R}$ and variables $x_i(t) \in \mathbb{R}$, $w_i(t) \in \mathbb{R}$ and it executes the PI algorithm

$$\begin{aligned}
\dot{x}_i(t) &= -\gamma x_i(t) - \sum_{j \in \mathcal{N}_i} a_{ij} \left[x_i(t) - x_j(t)\right] \\
&\quad + \sum_{j \neq i} b_{ji} \left[w_i(t) - w_j(t)\right] + \gamma u_i, \\
\dot{w}_i(t) &= - \sum_{j \in \mathcal{N}_i} b_{ij} \left[x_i(t) - x_j(t)\right], \qquad (1)
\end{aligned}$$

where $\gamma > 0$ is a global estimator parameter, and a_{ij}, b_{ij} are the estimator gains so that, if $j \notin \mathcal{N}_i$, then $a_{ij} = 0$, $b_{ij} = 0$. Let us consider simultaneously all the inputs and variables at the n nodes, $\mathbf{u} \in \mathbb{R}^n = (u_1, \ldots, u_n)^T$, $\mathbf{x} \in \mathbb{R}^n = (x_1, \ldots, x_n)^T$, $\mathbf{w} \in \mathbb{R}^n = (w_1, \ldots, w_n)^T$. Let $\mathbf{e}_{\mathbf{x}}(t) \in \mathbb{R}^n$ be the error vector

$$e_{\mathbf{x}}(t) = \mathbf{x}(t) - \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{u}. \qquad (2)$$

Let $W_P = [a_{ij}]$, $W_I = [b_{ij}]$ be respectively the proportional and the integral weights matrices, and L_P, L_I their associated Laplacians, $L_P = \operatorname{diag}(W_P \mathbf{1}) - W_P$, $L_I = \operatorname{diag}(W_I \mathbf{1}) - W_I$. If each node $i \in \{1, \ldots, n\}$ executes the PI algorithm with γ, L_P and L_I so that

$$\begin{aligned}
&\operatorname{rank}(L_I) = n - 1, && (3a)\\
&\varepsilon \in \mathbb{R} \text{ is such that } \Pi(L_P + L_P^T)\Pi \succeq 2\varepsilon\Pi, && (3b)\\
&\gamma > 0 \text{ is chosen such that } \gamma + \varepsilon > 0, && (3c)
\end{aligned}$$

then, for any input $\mathbf{u}$ and any initial states $\mathbf{x}(0), \mathbf{w}(0)$, the error vector $e_{\mathbf{x}}(t)$ tends to 0 exponentially as $t \to \infty$ [14, Theorem 5].

As weight matrices W_P, W_I, we use the Metropolis weights. Along this paper, we will use $W \in \mathbb{R}^{n \times n} = [w_{ij}]$ for the Metropolis weights matrix [21]

$$\begin{aligned}
w_{ij} &= \begin{cases} \frac{1}{1+\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}} & if\ j \in \mathcal{N}_i, \\ 0 & if\ j \notin \mathcal{N}_i, j \neq i, \end{cases} \\
w_{ii} &= 1 - \sum_{j \in \mathcal{N}_i} w_{ij}, \qquad (4)
\end{aligned}$$

for $i, j \in \mathcal{R}$, $j \neq i$, where $|\mathcal{N}_i|$, $|\mathcal{N}_j|$ are the number of neighbors of nodes i, j. We will use $L_W \in \mathbb{R}^{n \times n}$ for its associated Laplacian, $L_W = \operatorname{diag}(W\mathbf{1}) - W = I - W$. The Metropolis weights are widely used in consensus algorithms. They are convenient for distributed applications, since they can be computed by every robot based on local information. W is symmetric and doubly stochastic $W = W^T$, $W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T W = \mathbf{1}^T$. It has an eigenvalue at 1, and all its other eigenvalues $\lambda(W) \in (-1, 1)$. Its associated Laplacian L_W is symmetric and positive semidefinite [22, Theorem 1.37]. It has an eigenvalue at 0, and all the others $\lambda(L_W) \in (0, 2)$. For any connected communication graph, L_W satisfies (3a). It also satisfies (3b) for $\varepsilon = 0$ taking into account that $\Pi(L_W + L_W^T)\Pi = 2L_W$ since L_W is symmetric, and that L_W is positive semidefinite. Then, condition (3c) reduces to $\gamma > 0$.

III. AVERAGING ALGORITHM IN DISCRETE-TIME

The averaging algorithm in discrete time, using the Metropolis weights, is

$$\begin{aligned}
\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{w}(t+1) \end{bmatrix} &= A \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{w}(t) \end{bmatrix} + B\mathbf{u}, \\
A = I + h\begin{bmatrix} -\gamma I - L_W & L_W \\ -L_W & 0 \end{bmatrix}, &\quad B = \begin{bmatrix} h\gamma I \\ \mathbf{0} \end{bmatrix}, \qquad (5)
\end{aligned}$$

where $h > 0$ is the step size. Next, we derive the conditions on γ and h to ensure the convergence of (5). Along this section, we will use the notation $\lambda(A)$ for an eigenvalue of a matrix A, and $\mathbf{v}(A)$ for its associated eigenvector. We will use $\lambda_\star$, λ_2 for respectively the maximum and the second-smallest eigenvalues of L_W.

The discrete-time system converges if A has a single eigenvalue at 1, and the other eigenvalues have modulus strictly less than 1.

***Proposition** 3.1:* Let us consider that $\mathcal{G}$ is connected. If the step size h and the parameter γ satisfy

$$\begin{aligned}
h\gamma &< 2, && (6a)\\
\gamma &\geq 2\lambda_\star - \lambda_2, && (6b)\\
h(\gamma + \lambda_\star) &< 2, && (6c)
\end{aligned}$$

then all the eigenvalues of A are real. Besides, one of them is equal to 1, and all the others have modulus strictly less than one.

***Theorem** 3.2:* Let L_W be the Laplacian of the Metropolis weights W of a fixed, connected graph. Let $h > 0$, $\gamma > 0$ be as in Proposition 3.1. Then, for any input $\mathbf{u} \in \mathbb{R}^n$ and any initial states $\mathbf{x}(0) \in \mathbb{R}^n$, $\mathbf{w}(0) \in \mathbb{R}^n$, the states $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{w}(t) \in \mathbb{R}^n$ of the consensus algorithm (5) converge exponentially to

$$\mathbf{x}^* = \lim_{t\to\infty}(\mathbf{x}(t)) = \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{u},$$
$$\mathbf{w}^* = \lim_{t\to\infty}(\mathbf{w}(t)) = \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{w}(0) - \gamma\Pi(L_W + \frac{\mathbf{1}\mathbf{1}^T}{n})^{-1}\Pi\mathbf{u}, \quad (7)$$

as $t \to \infty$. Moreover, the error vector $e_{\mathbf{xw}}(t) = [\mathbf{x}(t)^T, \mathbf{w}(t)^T]^T - [(\mathbf{x}^*)^T, (\mathbf{w}^*)^T]^T$ after t iterations satisfies

$$||e_{\mathbf{xw}}(t)||_2 \leq \sqrt{2n}\ \beta\ |\lambda_{\max}(A)|^t\ ||e_{\mathbf{xw}}(0)||_2, \quad (8)$$

where

$$\beta = \frac{8\gamma}{\sqrt{(\gamma+\lambda_\star)^2 - (2\lambda_\star)^2}}, \quad (9)$$

$e_{\mathbf{xw}}(0)$ is the initial error, and $|\lambda_{\max}(A)|$ is the second eigenvalue of A with maximum absolute value.

In particular, the selection of $\gamma = 4$, and $h = 0.33$ satisfies Proposition 3.1 for any connected communication graph, since the eigenvalues of L_W satisfy $0 < \lambda_2 \leq \lambda_\star < 2$.

IV. DYNAMIC MAP MERGING

In this section, we explain how the dynamic consensus algorithm presented in the previous section is used for merging visual stochastic maps. We consider $n \in \mathbb{N}$ robots that explore an environment. There are $m \in \mathbb{N}$ features whose true positions $\theta \in \mathbb{R}^{2m}$ are unknown. Every feature has associated a unique identifier. The identifiers of all the features are stored in the identification vector $I_\theta \in \mathbb{N}^m$. Up to step $k \in \mathbb{N}$, each robot $i \in \{1, \ldots, n\}$, has observed m_i^k of them, with $m_i^k \leq m$, whose identities are $I_{\theta_i}^k \in \mathbb{N}^{m_i^k}$. Robot i estimates the feature positions and its own pose (position and orientation) based on its own observations. It builds a local map with mean $\hat{\mathbf{x}}_i^k \in \mathbb{R}^{2m_i^k}$ and covariance $\Sigma_i^k \in \mathbb{R}^{2m_i^k \times 2m_i^k}$. The n local maps are expressed in a common reference frame. Each robot i has communication capabilities to exchange information with its neighbors $\mathcal{N}_i$. The goal is that the robots compute and track the global merged map.

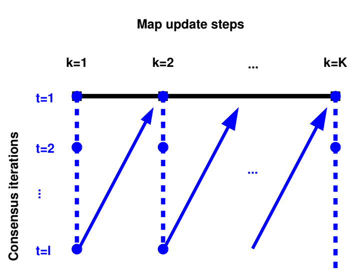


Fig. 1. **Dynamic map merging**. Robots construct a local map using the observations taken along K map update steps. Between any two map update steps, they execute l consensus iterations to compute the global map. Then, they use these global map estimates as an initial solution for the consensus iterations in the next step.

In our dynamic map merging approach (Fig.1), the robots locally build their maps. At some instants, the map update steps $k = 1, \ldots, K$, each robot i transforms its local map into the inputs

$$\mathbf{u}_{P_i}^k = (H_i^k)^T(\Sigma_i^k)^{-1}H_i^k, \quad \mathbf{u}_{q_i}^k = (H_i^k)^T(\Sigma_i^k)^{-1}\hat{\mathbf{x}}_i^k. \quad (10)$$

Those inputs are the local maps in IF form. The observation matrix $H_i^k \in \mathbb{R}^{2m_i \times 2m}$ is a permutation of an identity matrix with additional zero-columns. It relates the features $I_{\theta_i}^k$ locally observed by robot i and the features observed by all the robots I_θ. It also relates the n last local robot poses with its global representation. Each robot i initializes its consensus states with its last estates at $k-1$,

$$\mathbf{x}_{P_i}^k(0) = \mathbf{x}_{P_i}^{k-1}(l), \qquad \mathbf{x}_{q_i}^k(0) = \mathbf{x}_{q_i}^{k-1}(l),$$
$$\mathbf{w}_{P_i}^k(0) = \mathbf{w}_{P_i}^{k-1}(l), \qquad \mathbf{w}_{q_i}^k(0) = \mathbf{w}_{q_i}^{k-1}(l). \quad (11)$$

For $k = 1$, it initializes its consensus estates to zero,

$$\mathbf{x}_{P_i}^1(0) = \mathbf{w}_{P_i}^1(0) = \mathbf{0},\ \mathbf{x}_{q_i}^1(0) = \mathbf{w}_{q_i}^1(0) = \mathbf{0}. \quad (12)$$

since it has no previous information. Then, each robot i runs l iterations of the consensus algorithm (5) to compute the global map up to k. Notice that this algorithm was presented in the previous section for scalar variables $u_i, x_i(t), w_i(t) \in \mathbb{R}$. Its extension to the multi-dimensional variables $\mathbf{u}_{P_i}^k, \mathbf{x}_{P_i}^k(t), \mathbf{w}_{P_i}^k(t) \in \mathbb{R}^{m\times m}$, and $\mathbf{u}_{q_i}^k, \mathbf{x}_{q_i}^k(t), \mathbf{w}_{q_i}^k(t) \in \mathbb{R}^m$ simply consists of executing multiple copies of (5) on each of the entries. While the robots execute (5) to compute the global map up to k, simultaneously they locally take measurements and update their local maps. This new information introduced into their local maps will not be propagated until the next step $k+1$. They continue this process until the last map update step K. At this last step, they run the remaining $L - lK$ consensus iterations, where L is the total number of iterations.

Here we consider a fixed number of consensus iterations l per step. We do not specify the number of SLAM iterations between map update steps k and $k+1$. Using this strategy, if a map update step starts, and a robot is not ready for transmitting its updated local map, it can act as if it was disconnected from the communication network. For simplicity, along this section we present the algorithm equations assuming that the nodes know the observation matrices H_i^k before starting the consensus iterations. However, in the implemented algorithm, the robots incrementally discover I_θ in the information received from their neighbors. This procedure relies on the expansion and arrangement of the consensus states $\mathbf{x}_{P_i}^k$, $\mathbf{w}_{P_i}^k$, $\mathbf{x}_{q_i}^k$, $\mathbf{w}_{q_i}^k$. The version discovering I_θ provides the same results than the algorithm based on known observation matrices H_i^k, see [20] for a detailed discussion in the static case.

Later we will compare the results of our algorithm with the global merged map that would be computed by a centralized system up to step k. Let $\hat{\theta}_c^k$ be its mean and Σ_c^k its covariance matrix. Let $\mathbf{u}_{P_i}^k$, $\mathbf{u}_{q_i}^k$ be the local maps in IF form at step k, for $i \in \{1, \ldots, n\}$, defined in (10). If all these maps were available to a centralized system, the maximum likelihood global merged map would be [21]

$$\hat{\theta}_c^k = \left[\sum_{j=1}^n \mathbf{u}_{P_j}^k\right]^{-1}\left[\sum_{j=1}^n \mathbf{u}_{q_j}^k\right],\ \Sigma_c^k = \left[\sum_{j=1}^n \mathbf{u}_{P_j}^k\right]^{-1}. \quad (13)$$

A. Convergence with a finite number K of map update steps

Suppose that the robots perform a finite number K of map update steps. In this last step K, each robot i initializes its consensus filter using its previous estimates

$$\mathbf{x}_{P_i}^K(0) = \mathbf{x}_{P_i}^{K-1}(l), \qquad \mathbf{x}_{q_i}^K(0) = \mathbf{x}_{q_i}^{K-1}(l),$$
$$\mathbf{w}_{P_i}^K(0) = \mathbf{w}_{P_i}^{K-1}(l), \qquad \mathbf{w}_{q_i}^K(0) = \mathbf{w}_{q_i}^{K-1}(l). \tag{14}$$

and runs the consensus algorithm (5). Then, as we saw in the previous sections, the states at each robot i asymptotically converge to the average of the inputs at step K

$$\lim_{t\to\infty} \mathbf{x}_{P_i}^K(t) = \frac{1}{n}\sum_{j=1}^{n}(H_j^K)^T(\Sigma_j^K)^{-1}H_j^K,$$
$$\lim_{t\to\infty} \mathbf{x}_{q_i}^K(t) = \frac{1}{n}\sum_{j=1}^{n}(H_j^K)^T(\Sigma_j^K)^{-1}\hat{\mathbf{x}}_j^K. \tag{15}$$

For each robot $i \in \{1,\ldots,n\}$, $k = 1,2,\ldots$, $t = 0,1,\ldots$, we define $\hat{\theta}_i^k(t) = [\mathbf{x}_{P_i}^k(t)]^{-1}\mathbf{x}_{q_i}^k(t)$, and $\Sigma_{\hat{\theta}_i}^k(t) = \frac{1}{n}[\mathbf{x}_{P_i}^k(t)]^{-1}$. For the last step K, all the $\hat{\theta}_i^K(t)$ and $\Sigma_{\hat{\theta}_i}^K(t)$ asymptotically converge to

$$\lim_{t\to\infty}\hat{\theta}_i^K(t) = \hat{\theta}_c^K, \; \lim_{t\to\infty}\Sigma_{\hat{\theta}_i}^K(t) = \Sigma_c^K. \tag{16}$$

The mean and covariances are the same that would be obtained by a centralized IF for merging the local maps up to time K (13).

The convergence of the consensus iterations for step K is guaranteed for any initial state $\mathbf{x}_{P_i}^K(0), \mathbf{x}_{q_i}^K(0), \mathbf{w}_{P_i}^K(0), \mathbf{w}_{q_i}^K(0)$. In particular, it is guarantied for the initialization (14). Initial states (14) which are closer to the merged map than a no informative initialization (12) will produce a faster convergence.

B. Convergence with infinite map update steps

Now, let us suppose that the robots perform infinite map update steps, executing l consensus iterations between any two steps k, $k+1$. We use a simplified notation $\hat{\mathbf{x}}(\tau)$, $\Sigma(\tau)$ for the local map and covariance of a robot i, up to the SLAM step τ. These steps do not necessary coincide with the map update steps we used along this document. The map contains the last robot pose $\hat{\mathbf{x}}_v(\tau)$ and the M feature position $\hat{\mathbf{p}}_1(\tau),\ldots,\hat{\mathbf{p}}_M(\tau)^T$ estimates,

$$\hat{\mathbf{x}}(\tau) = \left[\hat{\mathbf{x}}_v(\tau)^T, \hat{\mathbf{p}}_1(\tau)^T, \ldots, \hat{\mathbf{p}}_M(\tau)^T\right]^T,$$
$$\Sigma(\tau) = \begin{bmatrix} \Sigma_{vv}(\tau) & \Sigma_{vm}(\tau) \\ \Sigma_{vm}^T(\tau) & \Sigma_{mm}(\tau) \end{bmatrix}, \tag{17}$$

where $\Sigma_{vv}(\tau)$ is the covariance matrix associated with the robot pose, $\Sigma_{mm}(\tau)$ is associated with the features, and $\Sigma_{vm}(\tau)$ is the cross covariance of the robot pose and features.

As more observations are made, the feature estimates of the local map converge [23]. Its covariance Σ_{mm} reaches a lower bound. However, the problematic part is the robot pose. If it continues moving around the environment, its pose estimate will vary. The covariances Σ_{vv}, Σ_{vm} will vary due to the noise associated to the robot motion. Then, every new step k, the robots will execute a new consensus. They will be tracking the robot poses.

C. Properties of the temporal global map estimates

An interesting property of this map merging algorithm is that the temporal global maps $\hat{\theta}_i^k(t)$ estimated at each robot i, are unbiased estimates of the true feature positions θ,

$$\mathrm{E}\left[\hat{\theta}_i^k(t)\right] = \mathrm{E}\left[(\mathbf{x}_{P_i}^k(t))^{-1}\mathbf{x}_{q_i}^k(t)\right] = \theta. \tag{18}$$

The temporal values of $\mathbf{x}_{P_i}^k(t)$, $\mathbf{x}_{q_i}^k(t)$, that evolve according to (5), can be alternatively expressed as a function of the inputs $\mathbf{u}_{P_j}^1,\ldots,\mathbf{u}_{P_j}^k$, $\mathbf{u}_{q_j}^1,\ldots,\mathbf{u}_{q_j}^k$, (10), and the initial states $\mathbf{x}_{P_j}^1(0)$, $\mathbf{w}_{P_j}^1(0)$, $\mathbf{x}_{q_j}^1(0)$, $\mathbf{w}_{q_j}^1(0)$, for $j \in \{1,\ldots,n\}$. When the initial states are zero (12), $\mathbf{x}_{P_i}^k(t)$, $\mathbf{x}_{q_i}^k(t)$ are

$$\mathbf{x}_{P_i}^k(t) = \sum_{s=1}^{k-1}\sum_{j=1}^{n}\Omega_{ij}^{k-(s+1),t,l}\mathbf{u}_{P_j}^s + \sum_{j=1}^{n}\Phi_{ij}^{t-1}\mathbf{u}_{P_j}^k,$$
$$\mathbf{x}_{q_i}^k(t) = \sum_{s=1}^{k-1}\sum_{j=1}^{n}\Omega_{ij}^{k-(s+1),t,l}\mathbf{u}_{q_j}^s + \sum_{j=1}^{n}\Phi_{ij}^{t-1}\mathbf{u}_{q_j}^k, \tag{19}$$

where $\Phi_{ij}^t, \Omega_{ij}^{k,t,l}$ are the elements at the i row and j column of the matrices $\Phi^t, \Omega^{k,t,l} \in \mathbb{R}^{2n\times n}$,

$$\Phi^t = \left[A^t + A^{t-1} + \cdots + A^1 + I\right]B,$$
$$\Omega^{k,t,l} = A^{t+kl}\Phi^{l-1}, \tag{20}$$

and the matrices A, B are (5). The local map $\hat{\mathbf{x}}_j^k$ at each robot j are an estimate of the positions of the features θ,

$$\hat{\mathbf{x}}_j^k = H_j^k\theta + \mathbf{v}_j^k, \tag{21}$$

where the noises $\mathbf{v}_j^k$ have zero mean and covariance Σ_j^k. Therefore, the inputs $\mathbf{u}_{q_j}^k$ are

$$\begin{aligned}\mathbf{u}_{q_j}^k &= (H_j^k)^T(\Sigma_j^k)^{-1}\hat{\mathbf{x}}_j^k \\ &= (H_j^k)^T(\Sigma_j^k)^{-1}H_j^k\theta + (H_j^k)^T(\Sigma_j^k)^{-1}\mathbf{v}_j^k.\end{aligned} \tag{22}$$

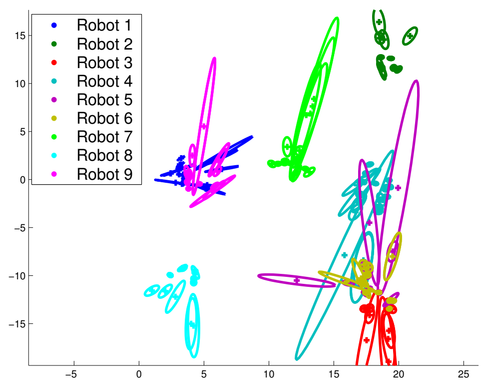
Then, $\hat{\theta}_i^k(t) = (\mathbf{x}_{P_i}^k(t))^{-1}\mathbf{x}_{q_i}^k(t)$ is

$$\hat{\theta}_i^k(t) = \theta + (\mathbf{x}_{P_i}^k(t))^{-1}\{\sum_{s=1}^{k-1}\sum_{j=1}^{n}\Omega_{ij}^{k-(s+1),t,l}(H_j^s)^T(\Sigma_j^s)^{-1}\mathbf{v}_j^s$$
$$+\sum_{j=1}^{n}\Phi_{ij}^{t-1}(H_j^k)^T(\Sigma_j^k)^{-1}\mathbf{v}_j^k\}. \tag{23}$$

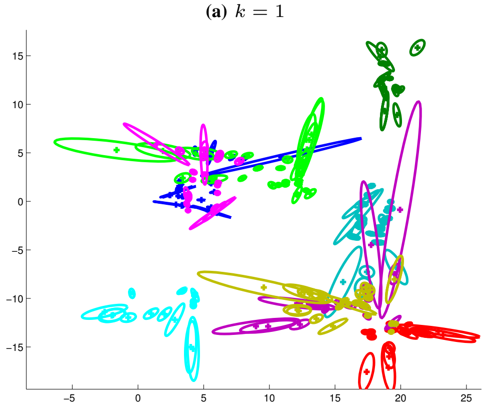
Since the noises $\mathbf{v}_j^k$ have zero mean for all $k = 1,2,\ldots$, $j \in \{1,\ldots,n\}$, the expected value of $\hat{\theta}_i^k(t)$ is θ. As a result, the robots do not need to wait for any specific number of iterations of the map merging algorithm. Instead, they can make decisions on their temporal global map estimates whenever they need it.

V. EXPERIMENTS

In order to analyze the behavior of the map merging method in real situations, we have carried out experiments using real data. We use a data set [24] with bearing information obtained with vision (Sony EVI-371DG). We select 9 subsections of the whole path for the operation of 9 different robots. The robots run a total of $K = 5$ map update steps. Between consecutive map update steps k, $k+1$, each robot performs 20 steps of a bearing-only SLAM algorithm [25]. We display in different colors the 9 local maps for the map update steps $k = 1$, and $k = 2$. (Fig. 2). The robots execute



(a) $k = 1$

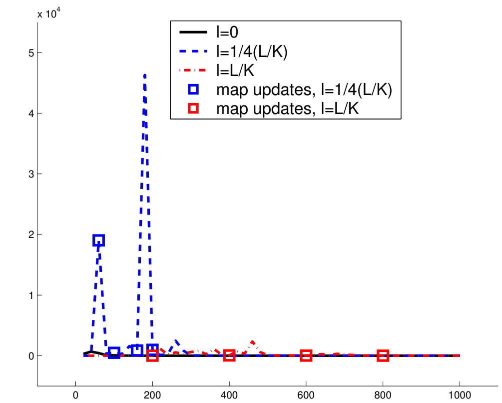


(b) $k = 2$

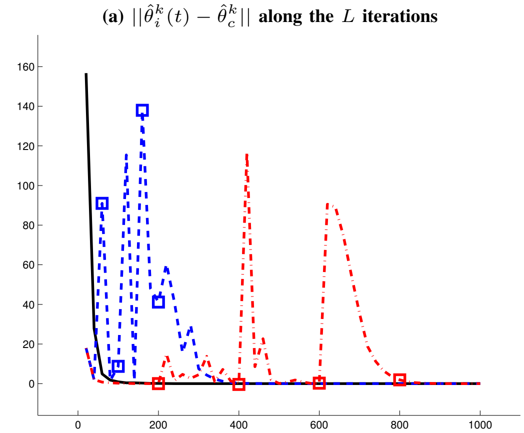
Fig. 2. Local maps of the 9 robots used for the update steps $k = 1$ (a) and $k = 2$ (b). They have been obtained after, respectively, 20 and 40 SLAM steps. They are expressed in a common reference frame. As it can be seen, in $k = 2$ the feature estimates have been updated. In addition, the robots have introduced new features into their local maps.

a total of $L = 1000$ consensus iterations. We experiment with 3 different configurations. In the first one, the robots execute all the consensus iterations after the last map update step, $l = 0$. This is equivalent to a static map merging. In the second case, we use $l = \frac{1}{4}(L/K)$, and in the last one, we use an equal number of iterations per step $l = (L/K)$. The best results are obtained with the first configuration $l = 0$, since the robots only need to agree on the last map $k = K$. In the second case, $l = \frac{1}{4}(L/K)$, the robots employ their first 50 iterations on reaching consensus on the first map $k = 1$. Then, every 50 iterations, the input maps change again. They start to reach consensus on the map $k = K$ after the iteration 200. And after that, they converge very fast to the global map.

In the last configuration, $l = (L/K)$, the robots use more consensus iterations than in $l = \frac{1}{4}(L/K)$ for the maps at $k = 1, 2, \ldots, 4$. Their estimates of these temporal maps are better. However, the robots start to estimate the last map $k = K$ after iteration 800. After the L iterations, the global maps



(a) $\|\hat{\theta}_i^k(t) - \hat{\theta}_c^k\|$ **along the** L **iterations**



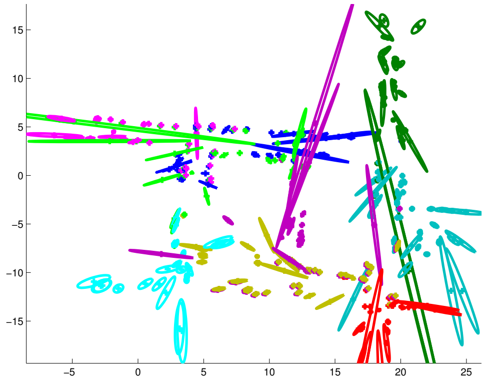
(b) $\|\mathrm{Tr}(\Sigma^k_{\hat{\theta}_i}(t)) - \mathrm{Tr}\left(\Sigma^k_c(t)\right)\|$ **along the** L **iterations**

Fig. 3. Estimation errors at robot 1 along the L consensus iterations. (a) $\|\hat{\theta}_i^k(t) - \hat{\theta}_c^k\|$, (b) $\|\mathrm{Tr}(\Sigma^k_{\hat{\theta}_i}(t)) - \mathrm{Tr}\left(\Sigma^k_c(t)\right)\|$. The configuration $l = 0$ (black solid line) employs all the iterations in reaching consensus on the last map $k = 5$. In the configuration $l = \frac{1}{4}(L/K)$ (blue dashed line), every 50 iterations the local maps change (blue squares). The robots start the consensus on the last map after the iteration 200. In the last configuration $l = (L/K)$ (red dash-dotted line), the map update steps start every 200 iterations (red squares). The consensus on the last map begins at iteration 800.

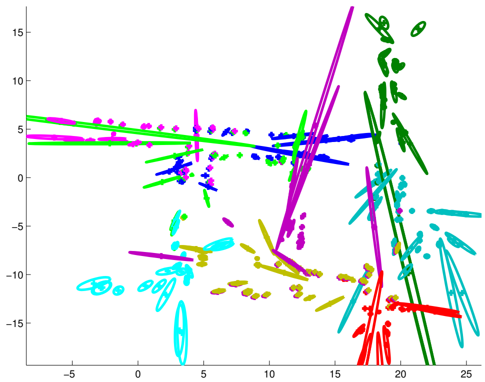
$\hat{\theta}_i^K(L)$, $\Sigma^K_{\theta_i}(L)$, computed by the dynamic map merging algorithm, are very close to the global map $\hat{\theta}_c^K$, Σ_c^K (13) that would be obtained by a centralized system (Fig.4 b). We show the global map at robot 1, for the $l = \frac{1}{4}(L/K)$ configuration (Fig.4 a), which is very similar to the maps computed by the other robots. Similar results have been obtained using the other configurations.

VI. CONCLUSIONS

In this paper, we have presented an algorithm for dynamically merging visual maps in a robot network with limited communications. It correctly propagates the new information added by the robots to their local maps. We have shown that, with the proposed strategy, the robots correctly track the global map. They finally obtain the last global map, which



(a) Global map estimated by robot 1



(b) Global map computed by a centralized system

Fig. 4. Global map estimated by robot $i = 1$ at the last consensus iteration, $\hat{\theta}_i^K(L)$, $\Sigma_{\hat{\theta}_i}^K(L)$, for the configuration $l = \frac{1}{4}(L/K)$ (a), and global map that would be obtained by a centralized system, $\hat{\theta}_c^K$,Σ_c^K (b). We display in different colors the sections that correspond to different initial local maps.

contains the last updated information at all the robots. This algorithm is extremely interesting for the robots, since they have an estimation of the global map during their operation. However, the performance of the algorithm is very sensitive to the parameter l, the number of consensus iterations per map update step. Although it is asymptotically correct for any value of l, the error reduction speed is quite different for the tested configurations. Thus, in the cases where the robots can obtain the global map after the exploration, a static map merging strategy is preferred. As future work, we will study the convergence speed of the averaging algorithm. We will use this speed for analyzing the precision of the global map estimates for the intermediate map update steps.

VII. ACKNOWLEDGMENTS

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

REFERENCES

[1] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.

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

[3] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, Jul. 2006.

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

[5] P. Pinies and J. D. Tardos, "Large-scale SLAM building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1094–1106, 2008.

[6] L. M. Paz, J. D. Tardos, and J. Neira, "Divide and conquer: EKF SLAM in $o(n)$," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1107–1120, 2008.

[7] N. Trawny, S. I. Roumeliotis, and G. B. Giannakis, "Cooperative multi-robot localization under communication constraints," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 4394–4400.

[8] E. M. Nebot, M. Bozorg, and H. F. Durrant-Whyte, "Decentralized architecture for asynchronous sensors," *Autonomous Robots*, vol. 6, no. 2, pp. 147–164, 1999.

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

[10] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed sensor fusion using dynamic consensus," in *IFAC World Congress*, Prague, CZ, Jul. 2005.

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

[12] E. W. Nettleton, H. F. Durrant-Whyte, P. W. Gibbens, and A. H. Goektogan, "Multiple-platform localization and map building," in *Sensor Fusion and Decentralized Control in Robotic Systems III. Proceedings of SPIE*. SPIE, 2000, vol. 4196, pp. 337–347.

[13] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," in *Int. Symposium of Robotics Research*, Sienna, Italy, Oct. 2003, pp. 254–266.

[14] 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*, San Diego, CA, Dec. 2006, pp. 398–403.

[15] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

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

[17] X. S. Zhou and 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, Oct. 2006, pp. 1785–1792.

[18] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

[19] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 1180–1187.

[20] R. Aragues, J. Cortes, and C. Sagues, "Distributed map merging in a robotic network," in *Workshop on Network Robot Systems, IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, Nice, France, Sep. 2008.

[21] L. Xiao, S. Boyd, and 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, Apr. 2005, pp. 63–70.

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

[23] M. W. M. 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*, vol. 17, no. 3, pp. 229–241, 2001.

[24] U. Frese and J. Kurlbaum, "A data set for data association," Jun. 2008. [Online]. Available: http://www.sfbtr8.spatial-cognition.de/insidedataassociation/

[25] R. Aragues and 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, May 2008, pp. 252–261.