2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Distributed Robust Data Fusion Based on Dynamic Voting

Eduardo Montijano, Sonia Martínez and Carlos Sagues

Abstract— **Data association mistakes, estimation and measurement errors are some of the factors that can contribute to incorrect observations in robotic sensor networks. In order to act reliably, a robotic network must be able to fuse and correct its perception of the world by discarding any outlier information. This is a difficult task if the network is to be deployed remotely and the robots do not have access to ground-truth sites or manual calibration. In this paper, we present a novel, distributed scheme for robust data fusion in autonomous robotic networks. The proposed method adapts the RANSAC algorithm to exploit measurement redundancy, and enables robots determine an inlier observation with local communications. Different hypotheses are generated and voted for using a dynamic consensus algorithm. As the hypotheses are computed, the robots can change their opinion making the voting process dynamic. Assuming that at least one hypothesis is initialized with only inliers, we show that the method converges to the maximum likelihood of all the inlier observations in a general instance. Several simulations exhibit the good performance of the algorithm, which also gives acceptable results in situations where the conditions to guarantee convergence do not hold.**

Index Terms - Mobile Robotic Systems, Robust Data Fusion, Distributed Algorithms, Dynamic Voting, Sensor Network Calibration.

I. Introduction

The emergence of new sensing, computation and communication technologies has stimulated an intense research activity in multi-robot sensor systems. The integration of multiple robots in complex network and information systems will enable users to close the loop in new applications for remote observation and actuation. On the other hand, the coordination of a team of robots introduces new challenges that designers of these systems must face.

The capacity of single and multi-robot systems to achieve any task depends crucially on their ability to autonomously perceive the world. By exploiting redundancy, multi-robot systems promise to yield increased versatility and robustness to failure. However, without proper mechanisms that ensure robust data fusion while rejecting spurious information, final multi-robot estimates can be highly unreliable. An example of this problem is illustrated in Fig. 1. Here, five robots are looking for an exit door of a room. Due to perception or data association mistakes, two of the robots have measurements of another door and a window. If the information is fused together without additional control mechanisms, the identified exit door location will be biased and unreliable.

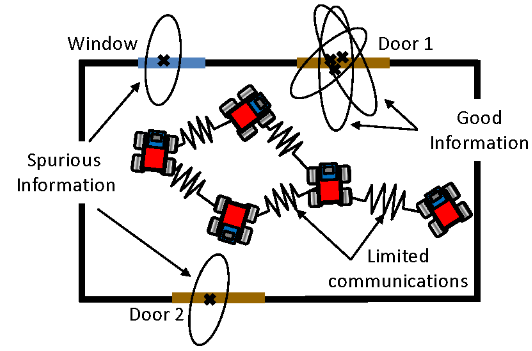


Fig. 1: Five robots are looking for the exit door of a room. Two of the robots have wrong estimates that must be discarded in the fusion algorithm. The proposed algorithm solves this problem in a decentralized way.

This paper proposes a novel fully-distributed algorithm to fuse sensed information in a robust way while discarding outliers. To achieve a robust behavior, our approach combines the RANSAC algorithm principles [5] with the distributed-algorithm tools of [12].

In our approach, the RANSAC hypotheses are defined as the maximum likelihood (ML) of different subsets of observations. The hypotheses are simultaneously generated over the communication network by distributed consensus. As hypotheses are constructed by the robots, they vote for and introduce their own observations into them, also in a decentralized fashion. Robots are allowed to change their opinion, making the voting process dynamic. We identify a set of reasonable conditions under which the method converges to the maximum likelihood of all the inlier observations, assuming that at least one hypothesis is initialized only with inliers. In simulations, the algorithms behave correctly even in cases not satisfying the sufficient conditions for convergence.

Related work: The problem of how to fuse several observations in a centralized manner has received considerable attention in the robotics literature. For example, [6], [7] study the integration of different sensor measurements taken by a single robot. In multi-robot systems, [8] investigates how to perform data fusion of several SLAM maps by a central unit. Compared with these works, our proposed solution is fully distributed, scales well with the size of the network, and is independent from any specific communication topology.

Regarding the distributed data fusion literature, recent efforts include [9], which uses Fuzzy logic to mix robot observations, and [11], [12], which take into account uncertainty. The latter paper was employed in [1] to merge several SLAM maps. Although these approaches are distributed, they

This work was supported by the project DPI2009-08126 grant AP2007-03282 of MEC (Spain), NSF-0712746, and NSF-0930919

E. Montijano and C. Sagues are at DIIS - I3A, University of Zaragoza, emonti@unizar.es, csagues@unizar.es

S. Martínez is at the department of Mechanical and Aerospace Engineering, Univ. of California, San Diego, soniamd@ucsd.edu

978-1-61284-380-3/11/$26.00 ©2011 IEEE

are not robust to spurious observations, which is the main topic of this paper. Finally, a distributed robust solution, also based on RANSAC, can be found in our previous work [10]. In contrast with this, the approach presented here considers covariances in the observations of the robots and overcomes several limitations of the previous algorithm. More precisely, in [10] the three steps of RANSAC are performed separately. However, in our new algorithm they are done at once, ensuring convergence to the maximum likelihood of the inlier observations simultaneously to the hypotheses generation and voting processes. This saves in running time and memory as flooding subroutines that were needed before have been now prescinded from; see Section IV for more information.

Finally, a connection can also be made with the recent literature on calibration techniques for sensor networks. In [4], a collaborative sensor-bias calibration scheme for physically distributed static sensors is proposed. The paper [4] purposefully obviates the treatment of outliers, which is the main focus of our work. As such, our algorithm can be seen as a contribution to this body of literature.

Along the paper, we do not discuss how the individual perception data is obtained or the data association problem [2]. We merely focus on the problem of performing robust distributed data fusion, discarding the outlier information.

II. Problem Statement

We consider a network of N robots labeled by $i \in \mathcal{V} = \{1, \ldots, N\}$. At each time $t > 0$, communications among robots are defined according to undirected graphs $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t)\}$, where $\mathcal{E}(t) \subset \mathcal{V} \times \mathcal{V}$ represents the edge set. Thus, robots i and j can communicate at time t if and only if $(i, j) \in \mathcal{E}(t)$. The neighbors of robot $i \in \mathcal{V}$ at time $t > 0$ are those that can directly communicate with it; i.e., $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}(t)\}$.

***Assumption** 2.1 (Conectness):* There exists a positive integer T such that, for any instant of time $t \geq 0$, the graph $\mathcal{G}(\mathcal{V}, \mathcal{E}(t) \cup \mathcal{E}(t+1) \cup \ldots \cup \mathcal{E}(t+T))$ is connected.

Let us consider some object in the environment defined by a set of attributes, $\boldsymbol{\theta} \in \mathbb{R}^d$. These attributes can be, for example, the position of the object in a world coordinate frame, its shape and color, or a set of descriptors that identify it. Each robot has some noisy initial measurement of $\boldsymbol{\theta}$, say $\mathbf{x}_i \in \mathbb{R}^d$, with uncertainty contained in the symmetric, semi-definite positive covariance matrix $\mathbf{\Lambda}_i \in \mathbb{R}^{d \times d}$.

The maximum likelihood (ML) of $\boldsymbol{\theta}$, $\boldsymbol{\theta}_{\text{ML}}$, is estimated using a weighted least-squares approximation from the robot measurements as

$$\boldsymbol{\theta}_{\text{ML}} = (\Sigma_{i=1}^N \mathbf{\Lambda}_i^{-1})^{-1} \Sigma_{i=1}^N \mathbf{\Lambda}_i^{-1} \mathbf{x}_i. \tag{1}$$

A distributed algorithm to compute the ML can be found in [12]. At the beginning, each robot initializes two state variables, $\mathbf{P}_i(0) = \mathbf{\Lambda}_i^{-1}$ and $\mathbf{q}_i(0) = \mathbf{\Lambda}_i^{-1}\mathbf{x}_i$. Then, at each iteration, they update these variables using

$$\begin{aligned} \mathbf{P}_i(t+1) &= a_{ii}(t)\mathbf{P}_i(t) + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)\mathbf{P}_j(t), \\ \mathbf{q}_i(t+1) &= a_{ii}(t)\mathbf{q}_i(t) + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)\mathbf{q}_j(t), \end{aligned} \tag{2}$$

where $\mathbf{A}(t) = [a_{ij}(t)]$ are the adjacency matrices associated with $\mathcal{G}(t)$, which must be doubly stochastic and non degenerate $\forall t$ [3]. In [12] it is shown that under these assumptions

$$\begin{aligned} &\lim_{t\to\infty} \mathbf{P}_i(t) = \frac{1}{N}\sum_{j\in\mathcal{V}} \mathbf{P}_j(0), \quad \lim_{t\to\infty} \mathbf{q}_i(t) = \frac{1}{N}\sum_{j\in\mathcal{V}} \mathbf{q}_j(0), \\ &\lim_{t\to\infty} \mathbf{P}_i^{-1}(t)\mathbf{q}_i(t) = \boldsymbol{\theta}_{\text{ML}}, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{3}$$

The speed of convergence of the algorithm will depend on the evolution of the topology. The covariance associated to the ML estimate is $\mathbf{\Lambda}_{\text{ML}} = \frac{1}{N}\lim_{t\to\infty} \mathbf{P}_i^{-1}(t)$, which will always be smaller than any of the initial covariances of the robots. Note that the inverse of $\mathbf{P}_i^{-1}(t)$ is always well defined because of the properties of $\mathbf{\Lambda}_i$ and eq. (2). For additional details about the algorithm, we refer the reader to [12].

The algorithm relies only on local communications and supports changes in the network topology. Moreover, the intermediate estimates, $\boldsymbol{\theta}_i(t) = \mathbf{P}_i^{-1}(t)\mathbf{q}_i(t)$, are unbiased; that is, $\text{E}[\boldsymbol{\theta}_i(t)] = \boldsymbol{\theta}_{\text{ML}}, \forall t \geq 0$. However, if some of the initial measurements contain extreme noise or spurious information on the attributes of the object being measured, the final estimation will be erroneous and unreliable.

To filter the outliers from the algorithm, we propose an algorithm that follows the RANSAC approach [5]. The algorithm generates a set of hypotheses computed from random subsets of robots and voted by all of them. The hypothesis with the larger number of votes, will be considered the right one and the robots who voted for it will constitute the set of inliers. In the following sections we will explain how all the process is done in a decentralized way using distributed consensus techniques.

III. Distributed Generation of Random Hypotheses

Following the RANSAC principles, we assume that each observation has equal probability of being a good observation p_{in} independent from the probability of the rest of observations. We will denote by $\mathcal{V}_{\text{in}} \subset \mathcal{V}$ the subset of robots with inlier information. The goal is to estimate the ML of the observations of the robots in $\mathcal{V}_{\text{in}}$ in a distributed way.

We define one hypothesis as the ML of the observations of a subset of the robots. For one hypothesis, h, let $\varnothing \neq \mathcal{V}_h \subseteq \mathcal{V}$ be the subset of robots whose observations generate the hypothesis and

$$\boldsymbol{\theta}_{\text{ML}}^h = (\sum_{i\in\mathcal{V}_h} \mathbf{\Lambda}_i^{-1})^{-1} \sum_{i\in\mathcal{V}_h} \mathbf{\Lambda}_i^{-1}\mathbf{x}_i. \tag{4}$$

***Proposition** 3.1:* The variable $\boldsymbol{\theta}_i(t) = \mathbf{P}_i^{-1}(t)\mathbf{q}_i(t)$, updated using (2) with initial conditions

$$[\mathbf{P}_i(0), \mathbf{q}_i(0)] = \begin{cases} [\mathbf{\Lambda}_i^{-1}, \mathbf{\Lambda}_i^{-1}\mathbf{x}_i] \text{ if } i \in \mathcal{V}_h \\ [\mathbf{0}, \mathbf{0}] \text{ otherwise} \end{cases}, \tag{5}$$

asymptotically converges to (4) for all $i \in \mathcal{V}$. ■

This means that the network is able to compute partial maximum likelihoods of different subsets of the robots in a decentralized way. The covariance associated to the partial ML will be $\mathbf{\Lambda}_{\text{ML}}^h = \frac{1}{N}\lim_{t\to\infty} \mathbf{P}_i^{-1}(t)$. We show now how the robots configure $\mathcal{V}_h$ for the different hypotheses in a distributed way.

Depending on the model to fit the observations, RANSAC requires c samples to generate one hypothesis [5]. Taking into account our definition of one hypothesis, c can be chosen arbitrarily because for any $c \geq 1$ the ML of the samples can be computed. Given a fixed c, to build one hypothesis h we require c robots to belong to $\mathcal{V}_h$, $|\mathcal{V}_h| = c$. We use a max consensus algorithm with random initial conditions to decide which c robots form the subset. Initially, each robot generates a random number $h_i > 0$ and the hypothesis set $\mathbf{h}_i(0) = \{h_i\}$, which is updated using

$$\mathbf{h}_i(t+1) = \max{}^c(\mathbf{h}_i(t) \textstyle\bigcup_{j \in \mathcal{N}_i} \mathbf{h}_j(t)), \tag{6}$$

where $\max^c$ selects the c maximum elements of the set. The max consensus algorithm is proved to converge in at most Diam($\mathcal{G}$) iterations [3]. In this case $\mathbf{h}_i(\text{Diam}(\mathcal{G}))$ will converge to the c maximum values of the network. Assuming that each robot generates a different random number the subset $\mathcal{V}_h$ is established as

$$\mathcal{V}_h = \{i \in \mathcal{V} \mid h_i \in \mathbf{h}_i(\text{Diam}(\mathcal{G}))\}. \tag{7}$$

Let us note that each robot knows if it belongs to $\mathcal{V}_h$ in a local way. The process can also be executed for all the hypotheses in parallel.

The number of hypotheses, K, is computed using the RANSAC formulas [5].

IV. Distributed Voting of the Hypotheses

A. Distributed Static Voting

For the sake of completeness, and also for comparison purposes with our current approach, we present a variation of the distributed voting method introduced in [10] in terms of stochastic variables. Assume that the hypotheses have already been computed. In order to vote for the set of hypotheses, each robot initializes a voting vector, $\mathbf{v}_i \in \mathbb{R}^K, i = 1, \ldots, N$, with as many elements as hypotheses to be voted for.

For every hypothesis h, the h^{th} component of the voting vector of robot i, $\mathbf{v}_i^h$, is initialized as

$$\mathbf{v}_i^h(0) = \begin{cases} 1, & \text{if } d(\mathbf{x}_i, \boldsymbol{\theta}_{\text{ML}}^h, \boldsymbol{\Lambda}_i^h) \leq \chi^2_{d,p}, \\ 0, & \text{if } d(\mathbf{x}_i, \boldsymbol{\theta}_{\text{ML}}^h, \boldsymbol{\Lambda}_i^h) > \chi^2_{d,p}, \end{cases} \tag{8}$$

where $d(\mathbf{x}_i, \boldsymbol{\theta}_{\text{ML}}^h, \boldsymbol{\Lambda}_i^h)$ is the Mahalanobis distance from the observation $\mathbf{x}_i$ to the partial ML estimate, $\boldsymbol{\theta}_{\text{ML}}^h$, and $\chi^2_{d,p}$ is the value of the Chi square distribution for d degrees of freedom and confidence probability p. The covariance, $\boldsymbol{\Lambda}_i^h$, is computed depending on whether robot i belonged to $\mathcal{V}_h$ or not,

$$\boldsymbol{\Lambda}_i^h = \begin{cases} \boldsymbol{\Lambda}_{\text{ML}}^h + \boldsymbol{\Lambda}_i & \text{if } i \notin \mathcal{V}_h, \\ \boldsymbol{\Lambda}_{\text{ML}}^h + \boldsymbol{\Lambda}_i - 2\boldsymbol{\Lambda}_{\text{ML}}^h \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\Lambda}_{\text{ML}}^h & \text{if } i \in \mathcal{V}_h. \end{cases} \tag{9}$$

In the first case, $\mathbf{x}_i$ and $\boldsymbol{\theta}_{\text{ML}}^h$ are independent and both covariances are considered. In the second case, there exists a correlation between $\mathbf{x}_i$ and $\boldsymbol{\theta}_{\text{ML}}^h$, which is taken care of by the third member of the sum.

After this, the robots exchange messages averaging $\mathbf{v}_i$ [3]. The vectors of all the robots will converge to the average of the initial conditions, which in this case is, for each hypothesis, the number of votes divided by the total number of robots. At this point, all the robots are able to select the best hypothesis, $h^* = \arg\max_h \mathbf{v}_i^h$ and by eq. (8) they also know if their observation was an inlier or an outlier. The final ML can then be obtained by a similar algorithm, where only the inlier robots contribute to the final result.

However, due to the asymptotic convergence of (2), the exact ML of the hypotheses will not be available to the robots, and this will require a stopping criterion to initialize the voting. Moreover, the previous approach requires three different consensus steps. We propose next a dynamic voting approach where the robots vote for (or not) a hypothesis when their observations pass the Chi-square test. If one hypothesis is supported by all the robots in $\mathcal{V}_{\text{in}}$, then the desired ML will be obtained by the algorithm in only one consensus step.

B. Distributed Dynamic Voting

In order to make the presentation clearer, we describe the algorithm just for one hypothesis, omitting the superscript h. As explained in the previous section, the robots initially decide the subset whose observations configure the hypothesis, eq. (6), and all of them initialize their states according to (5). In contrast with a static voting approach, now the voting process starts at the same time as the hypotheses generation. The initial votes are

$$\mathbf{v}_i(0) = \begin{cases} 1, & \text{if } i \in \mathcal{V}_h, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The new local updates for each robot are

$$\begin{aligned} \mathbf{P}_i(t+1) &= a_{ii}(t)\mathbf{P}_i(t) + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)\mathbf{P}_j(t) + \mathbf{u}_i^{\mathbf{P}}(t), \\ \mathbf{q}_i(t+1) &= a_{ii}(t)\mathbf{q}_i(t) + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)\mathbf{q}_j(t) + \mathbf{u}_i^{\mathbf{q}}(t), \\ \mathbf{v}_i(t+1) &= a_{ii}(t)\mathbf{v}_i(t) + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)\mathbf{v}_j(t) + \mathbf{u}_i^{\mathbf{v}}(t), \end{aligned} \tag{11}$$

where $\mathbf{u}_i^{\mathbf{P}}(t), \mathbf{u}_i^{\mathbf{q}}(t)$ and $\mathbf{u}_i^{\mathbf{v}}(t)$ are the dynamic inputs in the consensus rule and $\mathbf{A}(t) = [a_{ij}(t)]$ is defined as in (2).

In order to decide the inputs, each robot executes the Chi-square test with the current value of $\boldsymbol{\theta}_i(t) = \mathbf{P}_i^{-1}(t)\mathbf{q}_i(t)$. For abbreviation, we will denote

$$d_i(t) = d(\mathbf{x}_i, \boldsymbol{\theta}_i(t), \boldsymbol{\Lambda}_i). \tag{12}$$

Note that the inverse of $\mathbf{P}_i(t)$ is not always well defined. For the time instants t for which $\mathbf{P}_i^{-1}(t)$ does not exist, we cannot compute the Mahalanobis distance. However, at these instants we assign the distance $d_i(t)$ a value larger than $\chi^2_{d,p}$.

Denote the set of time instants in which robot i changes its opinion as follows:

$$\begin{aligned} \mathcal{T}_i^+ &= \{t \in \mathbb{N} \mid d_i(t) \leq \chi^2_{d,p} \ \wedge \ d_i(t-1) > \chi^2_{d,p}\}, \\ \mathcal{T}_i^- &= \{t \in \mathbb{N} \mid d_i(t) > \chi^2_{d,p} \ \wedge \ d_i(t-1) \leq \chi^2_{d,p}\}. \end{aligned} \tag{13}$$

The control inputs of robot i are given by:

$$[\mathbf{u}_i^{\mathbf{P}}(t), \mathbf{u}_i^{\mathbf{q}}(t), \mathbf{u}_i^{\mathbf{v}}(t)] = \begin{cases} [\mathbf{\Lambda}_i^{-1}, \mathbf{\Lambda}_i^{-1}\mathbf{x}_i, 1] & \text{if } t \in \mathcal{T}_i^{+}, \\ -[\mathbf{\Lambda}_i^{-1}, \mathbf{\Lambda}_i^{-1}\mathbf{x}_i, 1] & \text{if } i \in \mathcal{T}_i^{-}, \\ [\mathbf{0}, \mathbf{0}, 0] & \text{otherwise.} \end{cases} \tag{14}$$

Let us note that we have not considered the current estimation of the covariance matrix of the hypothesis in the Mahalanobis distance in eq. (12). We have chosen this conservative solution because in the first iterations of the algorithm the estimation of $\mathbf{P}_i(t)$ is highly unreliable. Usually, at these times $\mathbf{P}_i(t)$ is multiplied by weights very close to zero, resulting in large covariances due to the inverse $\mathbf{P}_i^{-1}(t)$. When this happens, the Mahalanobis distances are close to zero and spurious votes appear in the algorithm. On the other hand, by doing this, some false negative votes may be introduced in the voting. However, this is not very problematic if there are still enough inliers to compute a good solution.

Proposition 4.1: If $\mathcal{T}_i^+$ and $\mathcal{T}_i^-$ are finite for all $i \in \mathcal{V}$ then the rule (11) with control inputs (14) converges to

$$\lim_{t\to\infty} \boldsymbol{\theta}_i(t) = (\sum_{j\in\mathcal{V}_{\text{con}}} \mathbf{\Lambda}_j^{-1})^{-1} \sum_{j\in\mathcal{V}_{\text{con}}} \mathbf{\Lambda}_j^{-1}\mathbf{x}_j, \tag{15}$$

$$\lim_{t\to\infty} \mathbf{v}_i(t) = \frac{|\mathcal{V}_{\text{con}}|}{N}. \tag{16}$$

where $\mathcal{V}_{\text{con}} = \{i \in \mathcal{V} \mid d_i(t) \leq \chi^2_{d,p},\ t > t_{\max}\}$ and $t_{\max}$ is a time instant such that $t_{\max} > t, \forall t \in \mathcal{T}_i^+, \mathcal{T}_i^-, \forall i \in \mathcal{V}$. ■

If the robots are not indefinitely changing their vote, then the algorithm will achieve convergence to the ML of the subset of robots that have voted for it. At the end the hypothesis with the larger value of $\mathbf{v}_i^h$ will be the one selected by all the robots as the good one. Note that with this approach, once the hypothesis is selected there is no need to compute additional ML estimates.

What remains to be done now is to determine conditions that guarantee the covergence to the ML of the inliers.

C. Conditions to reach the ML *of* $\mathcal{V}_{\text{in}}$

We derive a set of reasonable conditions such that, if $\mathcal{V}_h \subseteq \mathcal{V}_{\text{in}}$ for one hypothesis, then the assumptions in Proposition 4.1 are met and $\mathcal{V}_{\text{con}} = \mathcal{V}_{\text{in}}$ for that hypothesis.

First, we impose a condition on the inlier observations. Since they are different measurements of the same vector, they have to be close to each other.

Condition 1: It holds that $d(\mathbf{x}_i, \mathbf{x}_j, \mathbf{\Lambda}_i) \leq \chi^2_{d,p}$ for any pair of robots, $i, j \in \mathcal{V}_{\text{in}}$.

Lemma 4.1: Let $CH(\mathcal{V}_{\text{in}})$ be the convex hull of the inlier observations. If Condition 1 is satisfied, then, for any robot $i \in \mathcal{V}_{\text{in}}$ and any $\mathbf{x} \in CH(\mathcal{V}_{\text{in}})$, we have $d(\mathbf{x}_i, \mathbf{x}, \mathbf{\Lambda}_i) \leq \chi^2_{d,p}$. This means that we have a set of points voted for by all the inliers, which leads to a second condition

Condition 2: For any $\mathcal{V}_h \subseteq \mathcal{V}_{\text{in}}$, $\boldsymbol{\theta}^h_{\text{ML}} \in CH(\mathcal{V}_{\text{in}})$.

The lemma also suggests a restriction to impose to the outlier observations.

Condition 3: For all $\mathbf{x} \in CH(\mathcal{V}_{\text{in}})$ and $k \notin \mathcal{V}_{\text{in}}$ it holds that $d(\mathbf{x}_k, \mathbf{x}, \mathbf{\Lambda}_k) > \chi^2_{d,p}$.

However, let us note that because our algorithm is not convex, the above conditions are not enough to ensure that one hypothesis instantiated with inliers will end up with all the inliers voting for it and all the outliers rejecting it. It could be possible that some outliers vote for it at some intermediate estimation or that one or more inliers constantly change their vote and convergence does not occur.

An additional condition to ensure convergence to the desired result is imposed. First let us notice that $\boldsymbol{\theta}_i(t)$ and the Mahalanobis distance, $d_i(t)$, can be written as functions of a vector $\mathbf{w} = (w_1, \ldots, w_N) \in [0,1]^N$, which represents the weights of the linear combination (not necessarily convex) of the different observations.

$$\boldsymbol{\theta}_i(\mathbf{w}) = \mathbf{P}_i^{-1}(\mathbf{w})\mathbf{q}_i(\mathbf{w}) = (\sum_{i\in\mathcal{V}} w_i \mathbf{\Lambda}_i^{-1})^{-1}(\sum_{i\in\mathcal{V}} w_i \mathbf{\Lambda}_i^{-1}\mathbf{x}_i),$$

$$d_i(\mathbf{w}) = d(\mathbf{x}_i, \boldsymbol{\theta}_i(\mathbf{w}), \mathbf{\Lambda}_i) = \sqrt{(\mathbf{x}_i - \boldsymbol{\theta}_i(\mathbf{w}))^T \mathbf{\Lambda}_i^{-1}(\mathbf{x}_i - \boldsymbol{\theta}_i(\mathbf{w}))},$$

Both functions are well defined for any $\mathbf{w} \neq \mathbf{0}$. However, let us recall that if $\mathbf{w} = \mathbf{0}$, we have defined $d_i(\mathbf{w}) > \chi^2_{d,p}$. The values of the different w_i are hard to compute as a function of t because they depend on the weights $a_{ij}(t)$ in eq. (11) and the network topology at each time instant. However, we can analyze the behavior of $d_i(\mathbf{w})$ over a compact subset of $[0,1]^N$. If the behavior in the set is the desired one, we will be able to ensure that for any t the algorithm will return the desired results.

Without loss of generality, let us assume that the robots are ordered so that we can separate the different elements of $\mathbf{w}$ in $\mathbf{w}_{\text{in}} \in [0,1]^{|\mathcal{V}_{\text{in}}|}$, the components corresponding to the inliers and $\mathbf{w}_{\text{out}}$ the components of the outliers, $\mathbf{w} = [\mathbf{w}_{\text{in}}, \mathbf{w}_{\text{out}}]$.

Condition 4: For any $i \in \mathcal{V}$, the partial derivatives of $d_i(\mathbf{w})$,

$$\frac{\partial d_i(\mathbf{w})}{\partial w_j} = \frac{(\mathbf{x}_i - \boldsymbol{\theta}_i(\mathbf{w}))^T \mathbf{\Lambda}_i^{-1}\mathbf{P}_i^{-1}(\mathbf{w})\mathbf{\Lambda}_j^{-1}(\mathbf{x}_j - \boldsymbol{\theta}_i(\mathbf{w}))}{d_i(\mathbf{w})}, \tag{17}$$

satisfy, $\forall j \in \mathcal{V}$ and $\mathbf{w}_{\text{out}} = \mathbf{0}$, that

$$\frac{\partial d_i(\mathbf{w})}{\partial w_j} = 0 \Leftrightarrow \begin{cases} \boldsymbol{\theta}_i(\mathbf{w}) = \mathbf{x}_i, \mathbf{x}_j, & \text{if } i, j \in \mathcal{V}_{\text{in}}, \\ \boldsymbol{\theta}_i(\mathbf{w}) = \mathbf{x}_i, & \text{if } i \in \mathcal{V}_{\text{in}}, j \notin \mathcal{V}_{\text{in}}, \\ \boldsymbol{\theta}_i(\mathbf{w}) = \mathbf{x}_j, & \text{if } i \notin \mathcal{V}_{\text{in}}, j \in \mathcal{V}_{\text{in}}. \end{cases} \tag{18}$$

Theorem 4.1: Under Conditions 1-4, for any $\mathcal{V}_h \subseteq \mathcal{V}_{\text{in}}$, the following holds:

- All the inliers eventually vote for the hypothesis

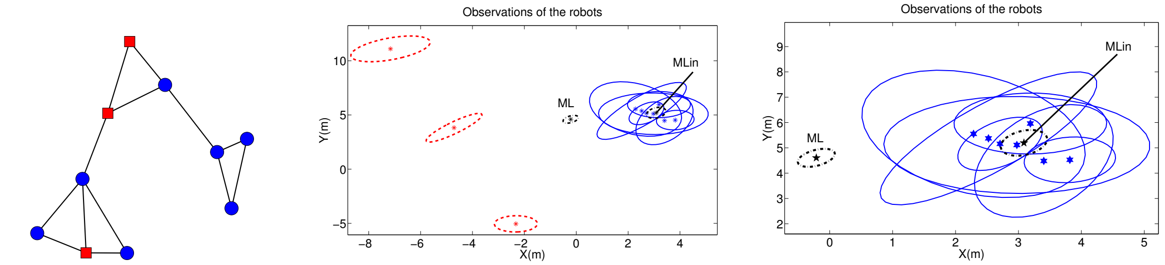
$$\exists\, t^+ \mid \forall t > t^+, \forall i \in \mathcal{V}_{\text{in}},\ d_i(t) \leq \chi^2_{d,p}. \tag{19}$$

- The outliers do not vote for the hypothesis at any time

$$d_k(t) > \chi^2_{d,p},\ \forall t > 0, \forall k \notin \mathcal{V}_{\text{in}}. \tag{20}$$

This means, by Proposition 4.1 that (11) will converge and that $\mathcal{V}_{\text{con}} = \mathcal{V}_{\text{in}}$. ■

Let us note that the conditions to know with certainty that the algorithm will converge to the desired result are relatively easy to occur in real scenarios. The first condition requires that the inlier observations are close to each other, which is easy to happen because they are good measurements of the same vector.



(a) Communication Graph (b) Observations of the robots (c) Observations of the inliers (zoom)

Fig. 2: Robust sensor fusion of a bi-dimensional feature observed by ten robots. (a) Communication network. Blue circles represent the robots with inlier information and red squares the outliers. (b) Observations of the ten robots. Seven robots have a good observation of the feature (blue crosses and solid ellipses) whereas three robots have observed another features (red crosses and dashed ellipses). If all the measurements are considered in the computation of the Maximum Likelihood, the obtained result is the black cross and dash-dot ellipse with the ML mark. The good maximum likelihood is the one in the middle of the inlier observations (MLin). (c) Zoom of the inlier observations and both ML.

The second condition is that the maximum likelihood of partial sets of inliers falls in the convex hull of the inliers. For a very small number of inliers (2 or 3) this will be hard to be true, but for a larger number of inliers this condition is almost sure to happen because the ML is similar to a weighted average.

The third condition requires that the outliers are far away from the convex hull of the inliers. This makes sense, otherwise one may argue that they are not real outliers as they would be posing a good observation of the feature.

The last condition is that the derivatives of the Mahalanobis distance only vanish at the points that correspond to the inlier observations. For the observation of any robot i, a global minimum is obtained when $\boldsymbol{\theta}_i(\mathbf{w}) = \mathbf{x}_i$, with Mahalanobis distance equal to zero. For the inliers this is not a problem because they must vote the hypotheses. For the outliers, it would be a problem that this happened, but it is almost impossible that a combination of inlier observations satisfying Conditions 1-3 returns the outlier. A global extreme can also appear if there exists $\mathbf{w}$ such that $(\mathbf{x}_k - \boldsymbol{\theta}(\mathbf{w}))$ is orthogonal to all $(\mathbf{x}_i - \boldsymbol{\theta}(\mathbf{w}))$ with respect to $\boldsymbol{\Lambda}_k^{-1}\mathbf{P}_k^{-1}(\mathbf{w})\boldsymbol{\Lambda}_i^{-1}$. Nevertheless, we have not encountered this situation in any of the simulations we have carried out and, provided that this extreme satisfies that $d_k(\mathbf{w}) > \chi^2_{d,p}$, the algorithm would still converge.

Finally, let us remark that the algorithm may still converge to the desired result if some of these conditions are not met. In section V we analyze this situation.

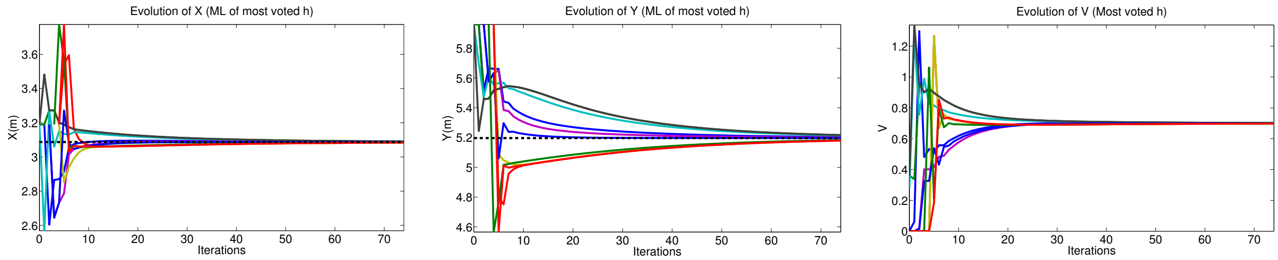
V. Simulation Results

We have tested our robust algorithm in a simulated environment to evaluate its performance. In Fig. 2, we show the observations of a two-dimensional feature by a network composed by 10 robots. We have chosen a two-dimensional feature in order to have a good visualization of the results, however, the algorithm is not restricted to this case and can be used with descriptors of any dimension. The communication network is fixed and we have used the Metropolis Weights [12] to configure $\mathbf{A}(t)$. Seven robots have good observations of the feature (blue crosses and solid ellipses) and 3 robots have outlier information (red crosses and dashed ellipses). If all the measurements are considered in the computation of the Maximum Likelihood, the obtained result is the black cross and dash-dotted ellipse with the ML mark at value $(-0.21, 4.60)$ while the ML of the inlier robots is $(3.08, 5.19)$ (MLin).

In the first step, the robots generate the different subsets $\mathcal{V}_h$ that will initialize the hypotheses. In the experiment we have set the probability of being an inlier to 0.6 and the probability of success in RANSAC to 0.99. As we mentioned in Section III, the value of c can be arbitrarily chosen. Larger values of c make many robots to plug their observations at the beginning, which is good if the number of inliers is large. However, the larger the c, the more hypotheses will be required to succeed, and for each additional hypothesis the amount of information to be communicated among agents grows linearly. For this reason we have set $c = 1$ resulting in a total of 6 hypotheses generated by the algorithm. In this example, the conditions stated in section IV-C are also satisfied, ensuring convergence to the ML of the inliers if one hypothesis is instantiated by robots in $\mathcal{V}_{\text{in}}$.

In Fig. 3 (a), (b), we show the evolution of the two coordinates of $\boldsymbol{\theta}_i(t)$ for the most voted hypothesis. The values of the different robots converge to the value of the ML of all the inliers (depicted in black dashed line in the graphics). In Fig. 3 (c), the evolution of $\mathbf{v}_i(t)$ for the same hypothesis is depicted. Eventually all the nodes reach the value 0.7, which is exactly the fraction of robots with inlier information. It is also remarkable that the number of iterations in which the robots change their opinion is considerably small. In less than 10 iterations, the graphics do not have discontinuities due to the inputs (11). After that point, the algorithm behaves as a static consensus algorithm.

We have also run a Monte Carlo simulation considering more general situations where the conditions of Section IV-C do not always hold. We have compared the results of the new algorithm with the robust consensus based on static voting and with the distributed consensus algorithm proposed in [12] to compute the ML of all the observations. We have run 1000 trials in which 20 robots have been considered. The probability for each robot to have inlier information has been set to 0.8, and only 3 hypotheses were generated in each trial. The inlier robots have measurements of the feature with gaussian error of zero mean and standard deviation



(a) (b) (c)

Fig. 3: Evolution of the Maximum Likelihood and the number of votes of the most voted hypothesis. In figures (a) and (b) we show the evolution of $\boldsymbol{\theta}_i(t)$ for the hypothesis that has obtained the most number of votes in the end. The dashed black line is the value of the ML of the robots with inlier information. It is observed that in both coordinates the values of $\boldsymbol{\theta}_i(t)$ asymptotically converge to it. The value of $\mathbf{v}_i^{h^*}$, depicted in (c), converges to 0.7 for all the robots, which is exactly the fraction of robots with inlier information.

TABLE I: Comparison of the different algorithms

Method	ML [12]	Static Voting	Dynamic Voting
Trials	1000	1000	1000
False Positive Votes	4015	375	355
False Negative Votes	0	472	192
Avrg. norm of error	2.063	0.444	0.438
Std. deviation of error	2.502	1.275	1.256
Iterations per trial	100	300	100

of 2 meters. For the outlier robots we have assigned a deviation of 10 meters. The covariance matrices have also been randomly generated with eigenvalues of mean 0.5 and standard deviation 0.5. Regarding the communication topology we have changed it at each iteration of each trial without affecting our results.

The results obtained in the simulation can be seen in Table I. The results using the non robust algorithm [12] are in the first column. In this case all the outliers participate in the different trials (a total of 4015). As a consequence, the average error in the estimation of the ML is large (2.06 meters). If the static voting method is used, the results are clearly improved. Only 375 false positive votes appear and the average error is reduced to 0.444 meters, with only 472 inliers thinking they have outlier information. Finally, if the proposed algorithm is used, 355 false positive votes are counted and the average error is of 0.438 meters. Moreover, since the robots are voting a dynamic observation which tends to the good ML, a fewer number of false negatives is registered (192). Although the results are similar to those of the static voting, the dynamic algorithm is much faster because it requires one third of communication rounds. The dynamic voting algorithm condenses the three steps of RANSAC in only one, requiring 100 iterations per trial. The only drawback of the robust algorithms with respect to [12] is that the size of the messages grows linearly with the number of hypotheses. However, this limitation is also found in the centralized version of RANSAC, where for each hypothesis a different solution must be computed and voted for.

VI. Conclusions and Future Work

In this paper, we have presented a distributed algorithm for sensor fusion robust to outlier observations. The algorithm relies only on local communications and supports changes in the topology of the communication network. Each robot detects in a local way if its observation should be merged in the algorithm or not. Conditions under which the maximum likelihood of the inliers is obtained have been given. Simulation results show the performance of our proposal. The algorithm also obtains good results in situations in which not all the requirements are satisfied. Although real experiments are not provided in the paper, possible applications in which this can be used include surveillance in camera networks, calibration of sensor networks or multi-robot SLAM.

References

[1] R. Aragues, J. Cortes, and C. Sagues. Dynamic consensus for merging visual maps under limited communications. In *IEEE Int. Conf. on Robotics and Automation*, pages 3032–3037, 2010.

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

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

[4] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak. A collaborative approach for in-place sensor calibration. In *2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, volume 2634 of *Lecture Notes in Computer Science*, pages 301–316. Springer-Verlag, 2003.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[6] N. Floudas, A. Polychronopoulos, O. Aycard, J. Burlet, and M. Ahrholdt. High level sensor data fusion approaches for object recognition in road environment. In *IEEE Intelligent Vehicles Symposium*, pages 136–141, 2007.

[7] G. Gate, A. Breheret, and F. Nashashibi. Centralized fusion for fast people detection in dense environment. In *IEEE Int. Conf. on Robotics and Automation*, pages 76–81, 2009.

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

[9] K. LeBlanc and A. Saffiotti. Multirobot object localization: A fuzzy fusion approach. *IEEE Transactions on Systems, Man, and Cybernetics-part B*, 39(5):1259–1276, 2009.

[10] E. Montijano, S. Martínez, and C. Sagues. De-RANSAC: Robust Consensus for Robot Formations. In *Workshop: Network Science and Systems Issues in Multi-Robot Autonomy. ICRA*, 2010.

[11] A. W. Stroupe, M. C. Martin, and T. Balch. Distributed sensor fusion for object position estimation by multi-robot systems. In *IEEE Int. Conf. on Robotics and Automation*, pages 1092–1098, 2001.

[12] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedins of the International Conference on Information Processing in Sensor Networks*, pages 63–70, 2005.