Distributed Multi-target Tracking in Camera Networks Using Multi-step Consensus

Nemanja Ilić, Khaled Obaid Al Ali, Miloš S. Stanković and Srdjan S. Stanković

Abstract—**In this paper problem of distributed multi-target tracking in large scale camera networks is discussed and a new algorithm based on consensus is proposed. The considered networks are characterized by sparse communication and coverage topologies, as well as the presence of clutter and multiple targets, which make the application of distributed frameworks challenging. The proposed algorithm uses Joint Probabilistic Data Associations (JPDA) methodology to address the clutter and multi-target problem. It applies a novel multi-step consensus strategy which adapts network communication weights to reflect differences between the nodes caused by the adopted network setting. Compared to the existing state of the art algorithms, in spite of having reduced communication requirements, the proposed scheme achieves the same performance quality in terms of the mean estimation error and much better performance in terms of the disagreement between the nodes' estimates due to its specific consensus design.**

Index Terms—**Camera networks, Distributed multi-target tracking, Multi-step consensus, Decentralized adaptation.**

I. Introduction

Development of high performance imaging sensors, together with the increased availability of low-cost cameras in recent years, have potentiated research connected to the application large scale camera networks, such as wide-area surveillance, disaster response, environmental monitoring, etc. One of the problems that have been in focus of researchers is target tracking using such networks, see *e.g.* [1]. Distributed schemes are becoming increasingly popular due to their high fault tolerance and scalability to a large number of sensors. Distributed multi-target tracking using sensor networks assumes that each sensor communicates and exchanges information with a subset of all nodes (neighboring nodes) in order to estimate and track the states of all the targets.

The fact that cameras have limited sensing range, implying that at each time instant there might be a significant share of sensors that do not receive measurements from the targets, makes that most well-known distributed estimation algorithms perform poorly in this situation. In addition, many state of the arts algorithms adopt the setting from the radar tracking community which takes into account the fact that at each time instant the measurements can originate from sources other than target (clutter). This uncertainty in the origin of measurements can be solved using different methodologies, but the most common is probabilistic data association (PDA) [2], together with its extended version that is better suited for the multi-target tracking problem - joint probabilistic data association (JPDA) [3].

The popular distributed tracking algorithm is Kalman-Consensus Filter (KCF), proposed in [4] by introducing a single-step consensus scheme whose dynamics is coupled with the estimation process of the target state. At each time iteration, it assumes that the nodes exchange the so-called information vectors and matrices (depending on local measurements, output matrices and measurement covariances), together with the target state estimates. Further extension of this approach towards its application in sensor networks with limited sensing range has been made by introducing its message-passing version in [5]. Distributed multi-target KCF, proposed in [6], combined PDA with KCF, assuming that all sensors received target-oriented measurements. Since the nodes communicate their estimates regarding multiple targets, [6] also addressed the problem of track to track association, based on a measure of distance.

Another state of the art algorithm, the so-called information-weighted consensus algorithm for distributed maximum a posteriori parameter estimation, and its extension for state estimation - the Information-weighted Consensus Filter (ICF) was proposed in [7]. ICF assumes multi-step consensus on two quantities that are of the same size as information vectors and matrices. Its version taking into account the presence of clutter and multiple targets (MTIC) was proposed in [8] by combining PDA with ICF. It did not consider the problem of track to track association by assuming perfect track to track matchings.

In [9] an algorithm for decentralized state estimation has been proposed in the form of a multi-agent network based on a combination of local estimators of Kalman filtering type and a single-step dynamic consensus strategy which is applied after each estimation step. It assumes the exchange of only information on state estimates between the nodes. Its modification, aimed at dynamically giving more communication weight to the nodes currently receiving the measurements was proposed in [10]. It is applied in situations where coverage topology is sparse, so that at each iteration the algorithm needs to adapt to the fact that only a small portion of the nodes actually observes the target. This adaptive algorithm introduced the exchange of additional zero-one variables between the nodes, reflecting their observation status (whether they observe the target or no), which were later used in calculating communication weights

N. Ilić, who is the corresponding author, is with the College of Technics and Technology, 37000 Kruševac, Serbia and Vlatacom Institute, 11000 Belgrade, Serbia; E-mail: nemili@etf.rs

Khaled Obaid Al Ali is with Etimad, Abu Dhabi, UAE and Vlatacom Institute, 11000 Belgrade, Serbia; E-mail: khaled@etimad.ae

M. S. Stanković is with the Innovation Center, Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia; E-mail: milsta@kth.se

S. S. Stanković is with the Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia and Vlatacom Institute, 11000 Belgrade, Serbia; E-mail: stankovic@etf.rs

for the estimates. It outperforms KCF while having smaller communication bandwidth requirements. The scheme assumes clutter-free scenarios and single target tracking.

In [11] the scheme from [10] was further refined by communicating real numbers instead of binaries between the nodes, reflecting both their current observation status and their communication history (the nodes previously receiving information from the observing nodes were given relatively higher communication weights compared to the nodes previously receiving information from the non-observing nodes). The obtained algorithm was generalized in [12] by considering the problem of single target tracking in cluttered environment. In scenarios with clutter one can have multiple measurements while tracking the single target, so using PDA each of these measurements is connected to the posterior probability that it originated from a target. Additionally, probability of a zero event corresponding to the case in which none of the measurements originated from a target is introduced. Obviously, these probabilities sum up to one. The underlying assumption in [12] is that when a target is being successfully tracked nodes that observe it will have probability corresponding to the target originated measurement close to one and that probability corresponding to clutter originated measurements will generally be close to zero. Those nodes that do not receive any measurements will have probability of zero event equal to one. In that way, for each node, one can take the probability of an event that is complement to the zero event as a measure of the weight that should be given to the node's target state estimate in the consensus scheme. The algorithm also included general setting that extends the PDA approach, based on the introduction of probabilities of target perceivability and target existence, in accordance with the Integrated Probabilistic Data Association (IPDA) methodology [13], aimed at increasing robustness of the whole scheme in cases of occulted targets. It outperformed single-step single-target version of MTIC, while preserving smaller communication requirements.

This paper builds up on the aforementioned results that started from [9] in two ways: it uses Joint Probabilistic Data Association (JPDA) methodology in order to appropriately address the problem of multi-target tracking [3], and it uses a novel multi-step consensus protocol, in line with the developments [7], [14], in order to achieve an even better performance in terms of the estimation error and the disagreement between the estimates. In multiple target tracking scenarios one deals with sets of probabilities for each target obtained via JPDA and with consensus scheme being applied on each of the sets of target state estimates. Track to track association problem has not been addressed but the scheme resembling the one from [6] can be applied in a straightforward way, which represents a topic for further work. It is to be noticed that this problem cannot be solved in a similar way when applying the MTIC algorithm, due to the fact that the nodes do not exchange actual estimates, so that the question of correspondence of a distance measure of the estimates to a distance measure of the quantities being exchanged arises.

It will be shown that the proposed design of a multi-step communication scheme enables successful network-wide tracking of all the targets and that the resulting algorithm achieves the same level of mean estimation error as MTIC, as well as significantly smaller disagreement between the nodes' estimates, while having smaller communication requirements. This is a result of an appropriately designed weighted consensus scheme which provides a faster and more robust algorithm, in contrast to the average consensus algorithm from MTIC.

The outline of the paper is as follows. Problem setting is discussed in Section II. The proposed distributed multi-target tracking algorithm with novel multi-step consensus scheme design is presented in Section III. Section IV gives illustration of the proposed consensus scheme together with characteristic simulation results.

II. Problem Definition

Consider N_T dynamic systems (targets models), where the k-th target, $k = 1, \ldots, N_T$, is represented by the following linear time-invariant discrete-time stochastic model

$$x^k(t+1) = Fx^k(t) + Ge^k(t), \tag{1}$$

where $x^k \in \mathbb{R}^m$ is the target state vector and e^k zero-mean white Gaussian noise with covariance matrix Q^k.

Assume that we have N *intelligent sensors* forming a *network* in which each *node* (camera) is supposed to have, in general, *limited sensing and communication ranges*, which determine at each t the set of nodes that can observe the targets and the sets of neighboring nodes exchanging messages. It is assumed that N_T is known to each node. At time t, each node gets m_i measurements denoted as $z_{i,j}$, $j = 1, \ldots, m_i$. For each measurement, the nodes do not know whether it originated from target or from clutter. Under the hypothesis that the measurement $z_{i,j}$ originated from the k-th target, the *sensing model* for the i-th node is

$$z_{i,j}(t) = H_i^k x^k(t) + v_i^k(t), \tag{2}$$

where $z_{i,j}(t) \in \mathbb{R}^{p_i}$ is the measurement vector, H_i^k a constant output matrix and v_i^k zero-mean white Gaussian measurement noise with covariance matrix R_i^k.

Let $Z_i(t) = \{z_{i,1}(t), \ldots, z_{i,m_i}(t)\}$ denote the set of $m_i(t)$ measurements obtained by node i at time t and $Z_i^t = \{Z_i(1), \ldots, Z_i(t)\}$. Let $\theta_{i,j}^k$ be the event that the j-th measurement of the i-th node originated from the k-th target, and $\theta_{i,0}^k$ the event that none of the measurements originated from the k-th target. The probabilities defined as

$$\beta_{i,j}^k(t) = P\{\theta_{i,j}^k | Z_i^t\}, \quad j = 0, 1, \ldots, m_i(t). \tag{3}$$

represent the core of the targets tracking filters. The well known PDA (Probabilistic Data Association) approach [2] computes these probabilities separately for each target k, under the assumption that all measurements not associated with target k are false (*i.e.*, they come from clutter). The Joint Probabilistic Data Association (JPDA) approach [3] computes these probabilities jointly across all targets and clutter, so that, for each target, measurements not associated with it may come from both other targets and clutter.

III. Adaptive Consensus Filter

A. Tracking Algorithm

For each target k, we use the algorithm [10], which is obtained from [9] after assuming that all subsystem models are equal to the target model (1), combined with the methodology from [2], [3]:

$$\begin{aligned} \xi_i^k(t|t) &= \xi_i^k(t|t-1) + L_i^k(t)\tilde{z}_i^k(t) \\ \xi_i^k(t+1|t) &= F\mathcal{C}^k(\xi_i^k(t|t)), \end{aligned} \tag{4}$$

$i = 1, \ldots, N$, where ξ_i^k is an estimate of x^k generated by the i-th node,

$$\tilde{z}_i^k(t) = \sum_{j=1}^{m_i(t)} \beta_{i,j}^k(t)\tilde{z}_{i,j}^k(t), \tag{5}$$

$\tilde{z}_{i,j}^k(t) = z_{i,j}(t) - H_i^k \xi_i^k(t|t-1)$, $\mathcal{C}^k(\cdot)$ is a consensus operator

$$\mathcal{C}^k(\xi_i^k(t|t)) = \sum_{j \in \mathcal{J}_i} c_{ij}^k(t)\xi_j^k(t|t), \tag{6}$$

$c_{ij}^k(t)$, $i, j = 1, \ldots, N$, are time varying weights, such that $N \times N$ matrix $C^k(t) = [c_{ij}^k(t)]$ (*consensus matrix*) is *row-stochastic* for all t, $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$, where $\mathcal{N}_i$ is the set of in-neighbors of the i-th node, *e.g.*, [15], [16], [17], and

$$L_i^k(t) = P_i^k(t|t-1)(H_i^k)^T S_i^k(t)^{-1}, \tag{7}$$

is the local Kalman gain obtained from (1) and (2) according to [18], using

$$\begin{aligned} P_i^k(t|t) =& P_i^k(t|t-1) + [\beta_{i,0}^k(t) - 1]L_i^k(t)S_i^k(t)L_i^k(t)^T \\ &+ \tilde{P}_i^k(t,t), \\ \tilde{P}_i^k(t,t) =& L_i^k(t)[\sum_{j=1}^{m_i(t)} \beta_{i,j}^k(t)\tilde{z}_{i,j}^k(t)\tilde{z}_{i,j}^k(t)^T \\ &- \tilde{z}_i^k(t)\tilde{z}_i^k(t)^T]L_i^k(t)^T, \\ P_i^k(t+1|t) =& FP_i^k(t|t)F^T + GQ^kG^T, \\ S_i^k(t) =& H_i^k P_i^k(t|t-1)(H_i^k)^T + R_i^k. \end{aligned} \tag{8}$$

The algorithm requires the exchange of state estimates only (size $m \times 1$) between the nodes (compare with [6], [8]). It contains two parts: 1) the filtering part, in which the local measurements are processed, and 2) the prediction part, in which the agreement between the nodes is enforced by forming a convex combination of the communicated local estimations, which are then included in the prediction step, *e.g.*, [15], [19], [16].

B. Multi-step adaptive consensus scheme

The following discussion will omit the target indexes since the same consensus strategy will be applied for each of the targets. We shall start from the set of weights

$$\gamma_i(t) = 1 - \beta_{i,0}(t), \tag{9}$$

$i = 1, \ldots, N$, as in [12]. When a target is being successfully tracked these weights should be close to one for the nodes observing the target and close to zero for the nodes that do not observe the target. Our aim is to design such a communication scheme that will asymptotically result in all the nodes having target state estimates equal to the weighted sum of the initial target state estimates (before the start of the consensus scheme), with weights corresponding to $\gamma_i(t)/\sum_{i=1}^N \gamma_i(t)$. Consequently, after the communication scheme is applied, all the nodes will have estimations that are mostly influenced by the nodes that observe the target. *E.g.*, if only one node observes the target, all the other nodes will have target state estimates close to that one node's estimate; if two nodes observe the target, all the other nodes will have estimates close to the mean of the two measuring nodes' estimates.

To this end, at each time instant t, we shall apply K consensus steps. In each consensus step the nodes exchange the variables based on $\gamma_i(t)$ as well as the target state estimates. The variables will be used for weighting the communicated estimates. Let A represents the network adjacency matrix with its diagonal elements set to 1. Before the consensus scheme is applied, we shall obtain A_c such that

$$\lim_{n \to \infty} A_c^n = \mathbf{1}\mathbf{1}^T/N, \tag{10}$$

where $\mathbf{1}$ is a vector of ones of appropriate size and A_c is a consensus matrix based on A which will be used in communicating weights between the nodes. Such matrix satisfies $\mathbf{1}^T A_c = \mathbf{1}^T$. It can be shown that this equation has, in principle, infinitely many solutions, using the procedure analogous to the one from [20]. Solving requires the knowledge of network topology; it is performed only once for a fixed topology. It is not a necessary prerequisite for the success of the whole consensus scheme but allows us to know the asymptotic behavior of the consensus algorithm. As a replacement, one can use a matrix with equal elements in each row summing up to one (see the following section).

Let $\gamma_i^{[\kappa]}(t)$ and $\xi_i^{[\kappa]}(t|t)$, $\kappa = 1, \ldots, K$, be the weight and the estimate of the i-th node connected to the κ-th consensus step, respectively. We shall start from

$$\gamma_i^{[1]}(t) = \gamma_i(t), \quad \xi_i^{[1]}(t|t) = \xi_i(t|t).$$

In this first consensus step, the nodes exchange $\gamma_i^{[1]}(t)$ and their estimates $\xi_i^{[1]}(t|t)$. The weights $\gamma_i^{[1]}(t)$ are being exchanged through A_c, so that the corresponding consensus matrix that defines the weights for the communicated estimates is obtained by

$$C^{[\kappa]}(t) = \left(A_c \cdot diag\left(\gamma_1^{[\kappa]}(t), \ldots, \gamma_N^{[\kappa]}(t)\right)\right)_{rs}, \tag{11}$$

$\kappa = 1$, where $(\cdot)_{rs}$ denotes an operator making the resulting matrix row-stochastic - it divides elements of each row of the argument matrix by the corresponding row sums.

Now that we have the consensus matrix $C^{[\kappa]}(t) = [c_{ij}^{[\kappa]}(t)]$, $i, j = 1, \ldots, N$, the target state estimates are obtained by

$$\xi_i^{*[\kappa]}(t|t) = \sum_{j \in \mathcal{J}_i} c_{ij}^{[\kappa]}(t)\xi_i^{[\kappa]}(t|t), \tag{12}$$

where star denotes the estimates obtained after the consensus step is applied.

For the next consensus step, our design requires that the weight of each node corresponds to the sum of the weights it received previously, so that $\gamma^{[\kappa+1]}(t) = [\gamma_1^{[\kappa+1]}(t) \quad \cdots \quad \gamma_N^{[\kappa+1]}(t)]^T$ becomes:

$$\gamma^{[\kappa+1]}(t) = A_c \cdot diag\left(\gamma_1^{[\kappa]}(t), \ldots, \gamma_N^{[\kappa]}(t)\right) \cdot \mathbf{1} = A_c \cdot \gamma^{[\kappa]}(t). \tag{13}$$

Also,

$$\xi_i^{[\kappa+1]}(t|t) = \xi_i^{*[\kappa]}(t|t). \tag{14}$$

At this point one can proceed with (11) and (12) for $\kappa + 1$ and likewise repeat the described procedure for the total of K consensus steps.

It might not be obvious but it is straightforward to show that this kind of procedure for $K \to \infty$ achieves consensus equivalent to

$$\lim_{K\to\infty} \xi_i^{*[K]}(t|t) = \sum_{j\in\mathcal{J}_i} c_{ij}^{\infty}(t)\xi_i(t|t), \tag{15}$$

where

$$[c_{ij}^{\infty}(t)] = \begin{bmatrix} \frac{\gamma_1(t)}{\gamma_1(t)+\cdots+\gamma_N(t)} & \cdots & \frac{\gamma_N(t)}{\gamma_1(t)+\cdots+\gamma_N(t)} \\ \vdots & \ddots & \vdots \\ \frac{\gamma_1(t)}{\gamma_1(t)+\cdots+\gamma_N(t)} & \cdots & \frac{\gamma_N(t)}{\gamma_1(t)+\cdots+\gamma_N(t)} \end{bmatrix},$$

which represents the desired result.

Consensus parameters $c_{ij}(t)$ in (6) are obtained by applying K consensus steps (target mark is omitted):

$$[c_{ij}(t)] = C(t) = C^{[K]}(t) \cdot C^{[K-1]}(t) \cdot \ldots \cdot C^{[1]}(t). \tag{16}$$

IV. Examples

A. Consensus scheme illustration

As an illustration of the proposed algorithm, let us consider a sensor network with $N = 3$ nodes represented by the following adjacency matrix (diagonal elements are set to 1):

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Notice that the underlying communication graph is directed, rendering the existing consensus based distributed estimation schemes inapplicable in this situation [4], [5], [7], [14].

First, we shall solve $\mathbf{1}^T A_c = \mathbf{1}^T$, so that A_c corresponding to the adjacency matrix A satisfies the asymptotic equation (10), and obtain:

$$A_c = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 0 & 3/4 & 1/4 \\ 1/2 & 0 & 1/2 \end{bmatrix}.$$

Herein, we have chosen a numerically attractive and simple solution from the set of possible solutions.

It can be easily verified that

$$\lim_{n\to\infty} A_c^n = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}.$$

At each time iteration of the estimation algorithm, for each target being tracked, K consensus steps are applied. In the first consensus step, assuming that we have the weight vector at a given time for a given target $\gamma^{[1]} = \gamma = \begin{bmatrix}\gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}^T$, our proposed algorithm gives the following consensus matrix:

$$C^{[1]} = \begin{bmatrix} \frac{\gamma_1/2}{\gamma_1/2+\gamma_2/4+\gamma_3/4} & \frac{\gamma_2/4}{\gamma_1/2+\gamma_2/4+\gamma_3/4} & \frac{\gamma_3/4}{\gamma_1/2+\gamma_2/4+\gamma_3/4} \\ 0 & \frac{3\gamma_2/4}{3\gamma_2/4+\gamma_3/4} & \frac{\gamma_3/4}{3\gamma_2/4+\gamma_3/4} \\ \frac{\gamma_1/2}{\gamma_1/2+\gamma_3/2} & 0 & \frac{\gamma_3/2}{\gamma_1/2+\gamma_3/2} \end{bmatrix}.$$

In the second consensus step, we use the following weight vector, obtained locally by each node through communication of the original weight vector γ in the first consensus step via the communication weights defined in A_c:

$$\gamma^{[2]} = \begin{bmatrix} \gamma_1^{[2]} \\ \gamma_2^{[2]} \\ \gamma_3^{[2]} \end{bmatrix} = \begin{bmatrix} \gamma_1/2 + \gamma_2/4 + \gamma_3/4 \\ 3\gamma_2/4 + \gamma_3/4 \\ \gamma_1/2 + \gamma_3/2 \end{bmatrix}.$$

The corresponding consensus matrix is $C^{[2]} =$

$$\begin{bmatrix} \frac{\gamma_1^{[2]}/2}{\gamma_1^{[2]}/2+\gamma_2^{[2]}/4+\gamma_3^{[2]}/4} & \frac{\gamma_2^{[2]}/4}{\gamma_1^{[2]}/2+\gamma_2^{[2]}/4+\gamma_3^{[2]}/4} & \frac{\gamma_3^{[2]}/4}{\gamma_1^{[2]}/2+\gamma_2^{[2]}/4+\gamma_3^{[2]}/4} \\ 0 & \frac{3\gamma_2^{[2]}/4}{3\gamma_2^{[2]}/4+\gamma_3^{[2]}/4} & \frac{\gamma_3^{[2]}/4}{3\gamma_2^{[2]}/4+\gamma_3^{[2]}/4} \\ \frac{\gamma_1^{[2]}/2}{\gamma_1^{[2]}/2+\gamma_3^{[2]}/2} & 0 & \frac{\gamma_3^{[2]}/2}{\gamma_1^{[2]}/2+\gamma_3^{[2]}/2} \end{bmatrix}.$$

By multiplying consensus matrices we obtain that the equivalent consensus matrix for the first two consensus steps is $C^{[2]} \cdot C^{[1]} =$

$$\begin{bmatrix} \frac{3\gamma_1/8}{3\gamma_1/8+5\gamma_2/16+5\gamma_3/16} & \frac{5\gamma_2/16}{3\gamma_1/8+5\gamma_2/16+5\gamma_3/16} & \frac{5\gamma_3/16}{3\gamma_1/8+5\gamma_2/16+5\gamma_3/16} \\ \frac{\gamma_1/8}{\gamma_1/8+9\gamma_2/16+5\gamma_3/16} & \frac{9\gamma_2/16}{\gamma_1/8+9\gamma_2/16+5\gamma_3/16} & \frac{5\gamma_3/16}{\gamma_1/8+9\gamma_2/16+5\gamma_3/16} \\ \frac{\gamma_1/2}{\gamma_1/2+\gamma_2/8+3\gamma_3/8} & \frac{\gamma_2/8}{\gamma_1/2+\gamma_2/8+3\gamma_3/8} & \frac{3\gamma_3/8}{\gamma_1/2+\gamma_2/8+3\gamma_3/8} \end{bmatrix}.$$

On the other side,

$$A_c^2 = \begin{bmatrix} 3/8 & 5/16 & 5/16 \\ 1/8 & 9/16 & 5/16 \\ 1/2 & 1/8 & 3/8 \end{bmatrix}.$$

It can be seen that the coefficients that multiply the initial weights in the equivalent consensus matrix are in fact equal to the corresponding parameters of the power of A_c.

By applying further consensus steps, it is clear that for K large enough we would obtain $C^{[K]} \cdot C^{[K-1]} \cdot \ldots \cdot C^{[1]} \approx$

$$\begin{bmatrix} \frac{\gamma_1/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} & \frac{\gamma_2/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} & \frac{\gamma_3/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} \\ \frac{\gamma_1/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} & \frac{\gamma_2/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} & \frac{\gamma_3/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} \\ \frac{\gamma_1/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} & \frac{\gamma_2/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} & \frac{\gamma_3/3}{\gamma_1/3+\gamma_2/3+\gamma_3/3} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{\gamma_1}{\gamma_1+\gamma_2+\gamma_3} & \frac{\gamma_2}{\gamma_1+\gamma_2+\gamma_3} & \frac{\gamma_3}{\gamma_1+\gamma_2+\gamma_3} \\ \frac{\gamma_1}{\gamma_1+\gamma_2+\gamma_3} & \frac{\gamma_2}{\gamma_1+\gamma_2+\gamma_3} & \frac{\gamma_3}{\gamma_1+\gamma_2+\gamma_3} \\ \frac{\gamma_1}{\gamma_1+\gamma_2+\gamma_3} & \frac{\gamma_2}{\gamma_1+\gamma_2+\gamma_3} & \frac{\gamma_3}{\gamma_1+\gamma_2+\gamma_3} \end{bmatrix}.$$

Therefore, by applying our algorithm we have obtained the desired asymptotic behavior of a communication network using only locally obtained communication weights.

In order to avoid divide by zero situations in practical scenarios where in some neighborhoods all nodes have zero weights, a simple fix can be applied by adding some small number ϵ to all γ_i, $i = 1, \ldots, N$.

If we didn't solve for A_c that satisfies (10) in the beginning, but use equal elements in rows summing up to 1:

$$A_c = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{bmatrix}, \quad \lim_{n\to\infty} A_c^n = \begin{bmatrix} 1/3 & 2/9 & 4/9 \\ 1/3 & 2/9 & 4/9 \\ 1/3 & 2/9 & 4/9 \end{bmatrix},$$

we would obtain for K large enough $C^{[K]} \cdot C^{[K-1]} \cdot \ldots \cdot C^{[1]} \approx$

$$\begin{bmatrix} \frac{\gamma_1/3}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} & \frac{2\gamma_2/9}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} & \frac{4\gamma_3/9}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} \\ \frac{\gamma_1/3}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} & \frac{2\gamma_2/9}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} & \frac{4\gamma_3/9}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} \\ \frac{\gamma_1/3}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} & \frac{2\gamma_2/9}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} & \frac{4\gamma_3/9}{\gamma_1/3+2\gamma_2/9+4\gamma_3/9} \end{bmatrix},$$

which would still exhibit good results in the distributed estimation scheme, since in situations where $\gamma_i \approx 1$ for measuring nodes and $\gamma_i \approx 0$ for non-measuring nodes a clear emphasis is given on the estimates of measuring nodes. The only difference is in non-equal weights connected to the estimates in neighborhoods with multiple measuring nodes.

B. Simulations

The considered problem is distributed tracking of $N_T = 4$ targets via a network of $N = 15$ cameras (sensors, nodes), each target moving within a 500×500 space [7], [8]. The cameras are randomly distributed in the space with random orientations resulting in overlapping field-of-views (FOVs). Communication range is set to 200 units. Targets' initial positions are randomly selected within the square area, with initial speeds set to 2 units per time step with random directions. The dynamics of all targets is modeled using a constant speed model with time increment of 1. The process covariance Q is set to $diag(10, 10, 1, 1)$. Only the targets that remained in the simulation area after the duration of the experiment (20 time iterations) were considered. Cameras observe targets positions according to their FOVs, which are represented by equilateral triangles with the height of 300 units. Measurement noise covariances are set to $100I_2$. The initial prior state estimates are randomly selected around initial target states with noise covariance of $5Q$. The initial error covariance matrices are set to $diag(100, 100, 10, 10)$ for all the nodes. Number of consensus steps is set to $K = 10$, as in [7], [8]. Regarding the parameters included in calculating $\beta_{i,j}^k(t)$ (see [3] for details), false measurements (clutter) were assumed Poisson distributed with the spatial density of $1/32$, gate probability was set to 0.99 and probability of detection was calculated individually for each node at each time instant by integrating the probability density function of the predicted estimate over the triangular area visible to the camera.

Fig. 1 illustrates performance of the proposed scheme, giving position estimates of all the nodes, together with target trajectories. It shows that the proposed algorithm (Adaptive Multi-step Consensus Filter, AMCF) gives network-wide accurate estimates of the states of all targets. The estimates were also generated via MTIC algorithm. The modified version of JPDA-KCF algorithm taking into account limited sensing range sensors was also considered but it did not perform well, yielding not comparable results. It can be seen that the proposed algorithm achieves the same level of performance

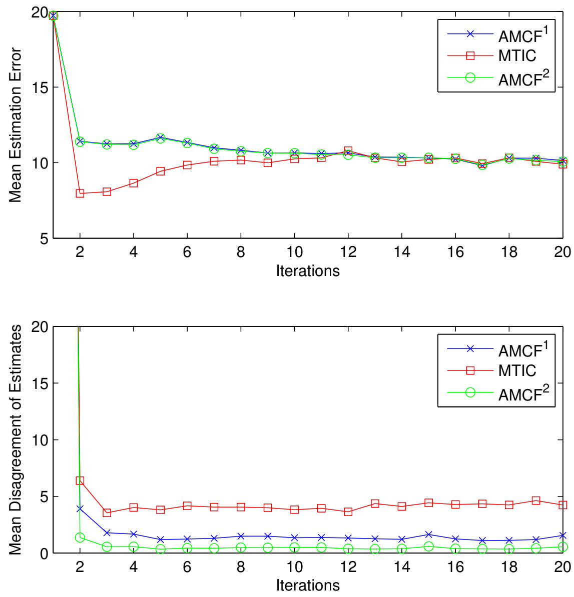

Fig. 2. Average distance per node per target between the true and the estimated positions (top) and average variance of the estimates per target (bottom) versus time

as MTIC in terms of the estimation error and that it reaches greater agreement of the estimates compared to MTIC.

For a more thorough comparison, an experiment was set with 100 different randomly selected networks, each with 4 randomly generated tracks. Estimation error (average distance per node per target between targets positions estimates and the actual positions) and disagreement between the nodes (average variance of the estimates per target) were calculated. The proposed algorithm was simulated in two versions, in line with the discussion from the previous subsection: $AMCF^1$, where consensus schemes used matrices satisfying (10), and $AMCF^2$ where these matrices were replaced by matrices with equal elements in each row summing up to one. It can be seen that the proposed algorithm exhibits similar performance in both versions, with estimation errors at the same level as MTIC, but with significantly smaller disagreement.

Finally, we compared performances of the aforementioned algorithms with respect to the number of consensus steps applied. To this end, the setting above was repeated for $K = 2, \ldots, 10$. It can be seen that the proposed algorithm for lower K outperforms MTIC in terms of disagreement even more than for higher K. This implies that with the careful communication weights design one can apply consensus with as low as $K = 3$ and obtain an algorithm with very good performance (this K would of course be higher for sparser communication topologies).

References

[1] D. E. Maggio and D. A. Cavallaro, *Video Tracking: Theory and Practice*, 1st ed. Wiley Publishing, 2011.

[2] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451 – 460, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109875900217

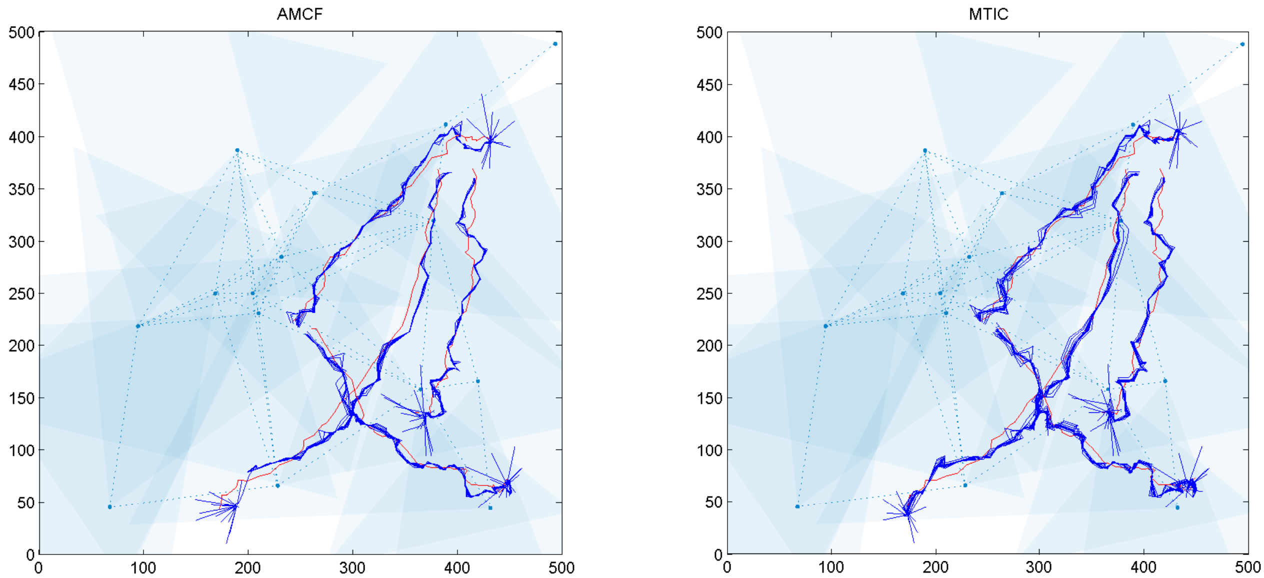

Fig. 1. Targets positions estimates of all nodes (blue lines), together with targets trajectories (red lines). Camera positions and FOVs are also shown (dots and triangles in shade of blue, respectively), as well as the communication network (dotted lines).

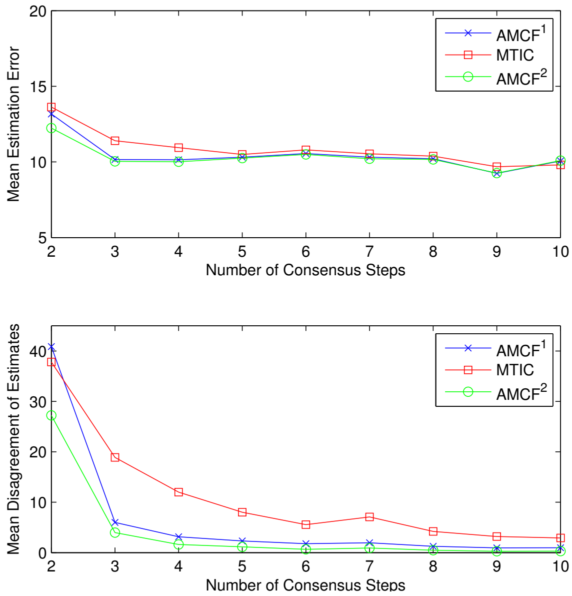

Fig. 3. Average distance per node per target between the true and the estimated positions (top) and average variance of the estimates per target (bottom) at $t = 20$, versus number of consensus steps.

[3] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, Jul 1983.

[4] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE Conf. Decision and Control*, 2007, pp. 5492–5498.

[5] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in *Proc. American Control Conference*, 2008, pp. 3157–3162.

[6] N. F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *2008 47th IEEE Conference on Decision and Control*, Dec 2008, pp. 1085–1090.

[7] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, Dec 2013.

[8] A. T. Kamal, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed multi-target tracking and data association in vision networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1397–1410, July 2016.

[9] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, pp. 1397–1406, 2009.

[10] N. Ilić, M. S. Stanković, and S. S. Stanković, "Adaptive consensus-based distributed target tracking in sensor networks with limited sensing range," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 778–785, March 2014.

[11] S. S. Stanković, N. Ilić, K. O. A. Ali, and M. S. Stanković, "Distributed consensus based ipdaf for tracking in vision networks," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sept 2016, pp. 304–309.

[12] N. Ilić, K. O. A. Ali, M. S. Stanković, and S. S. Stanković, "Adaptive consensus based distributed target tracking in camera networks," in *3rd International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*, June 2016.

[13] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1237–1241, Jun 1994.

[14] 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 *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Dec 2011, pp. 7863–7869.

[15] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, 2007.

[16] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.

[17] R. A. Horn and C. A. Johnson, *Matrix Analysis*. Cambridge Univ. Press, 1985.

[18] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poola, M. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, pp. 1453–1464, 2004.

[19] W. Ren and R. Beard, "Consensus seeking in multi-agent systems using dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, pp. 655–661, 2005.

[20] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4069–4074, Dec 2016.