Distributed Multi-Target Tracking and Data Association in Vision Networks

Ahmed T. Kamal, Jawadul H. Bappy, Jay A. Farrell, *Fellow, IEEE*,
and Amit K. Roy-Chowdhury, *Senior Member, IEEE*

Abstract—Distributed algorithms have recently gained immense popularity. With regards to computer vision applications, distributed multi-target tracking in a camera network is a fundamental problem. The goal is for all cameras to have accurate state estimates for all targets. Distributed estimation algorithms work by exchanging information between sensors that are communication neighbors. Vision-based distributed multi-target state estimation has at least two characteristics that distinguishes it from other applications. First, cameras are directional sensors and often neighboring sensors may not be sensing the same targets, i.e., they are *naive* with respect to that target. Second, in the presence of clutter and multiple targets, each camera must solve a data association problem. This paper presents an information-weighted, consensus-based, distributed multi-target tracking algorithm referred to as the Multi-target Information Consensus (MTIC) algorithm that is designed to address both the naivety and the data association problems. It converges to the centralized minimum mean square error estimate. The proposed MTIC algorithm and its extensions to non-linear camera models, termed as the Extended MTIC (EMTIC), are robust to false measurements and limited resources like power, bandwidth and the real-time operational requirements. Simulation and experimental analysis are provided to support the theoretical results.

Index Terms—Consensus, distributed tracking, data association, camera networks

1 Introduction

Due to the availability of modern low-cost sensors, large-scale camera networks are being used in applications such as wide-area surveillance, disaster response, environmental monitoring, etc. Multiple sensors can cover more of an area and provide views from different angles so that the fusion of all their measurements may lead to robust scene understanding. Among different information fusion approaches, distributed schemes are often chosen over centralized or hierarchical approaches due to their scalability to a large number of sensors, ease of installation and high tolerance to node failure. In this paper, we focus on the problem of *distributed multi-target tracking in a camera network*. We use the term *distributed* to mean that each camera processes its own data and arrives at a final solution through negotiations with its neighbors; there is no central processor.[1]

While a number of distributed estimation strategies have been developed (detailed review provided later), vision networks pose some unique and interesting challenges. These arise from the fact that most cameras are directional sensors, and the data they collect is expensive to process and transmit. Cameras that are viewing the same targets or may potentially view the same target over time may not be in direct communication range. In addition, each camera views very few of the full set of targets, even though the full set of cameras views them all. This can lead to sets of nodes that are *naive* or uninformed about the states of some of the targets. Without resource constraints at each node (highly unrealistic in a distributed setting), this issue could be resolved over multiple communication steps. However, this may not be possible in situations with limited power, bandwidth and the real-time operational requirements. Thus the problem of estimating the state vector (i.e., the concatenated state of all the targets), becomes very challenging in a distributed environment due to the limited local observability at each agent, especially when combined with constrained resources. Most well-known distributed estimation problems fare poorly in this situation; yet it is a fundamental issue if decentralized operation with vision networks is to be feasible.

Fig. 1 depicts an application scenario where the objective is for the network of cameras to collaboratively track all persons. Successful collaboration is built on each camera maintaining an estimate of the state of all targets, even when neither the camera or its neighbors directly detect some targets. We call a node '*naive*' about a target when it has no measurements of that target available in its *local neighborhood* (consisting of the node and its immediate network neighbors) as defined by the communication graph. When an agent is naive about a target, the target state is not (directly) observable by that agent. As long as some agent in the network has a measurement of the target, the centralized problem is still observable, which will enable a decentralized solution.

1.1 Problem Description and Solution Overview

Most of the work in distributed tracking has been in the multi-agent systems community. Methods that work very well in applications where each agent has local observability

1. The term distributed has also been used in computer vision for a camera network that is distributed over a wide area, but where the processing is centralized.

• *The authors are with the University of California, Riverside. E-mail: ahmed.kamal@email.ucr.edu, {mbappy, farrell, amitrc}@ece.ucr.edu.*

Manuscript received 16 Mar. 2014; revised 24 Aug. 2015; accepted 9 Sept. 2015. Date of publication 30 Sept. 2015; date of current version 10 June 2016. Recommended for acceptance by I. Reid.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2484339

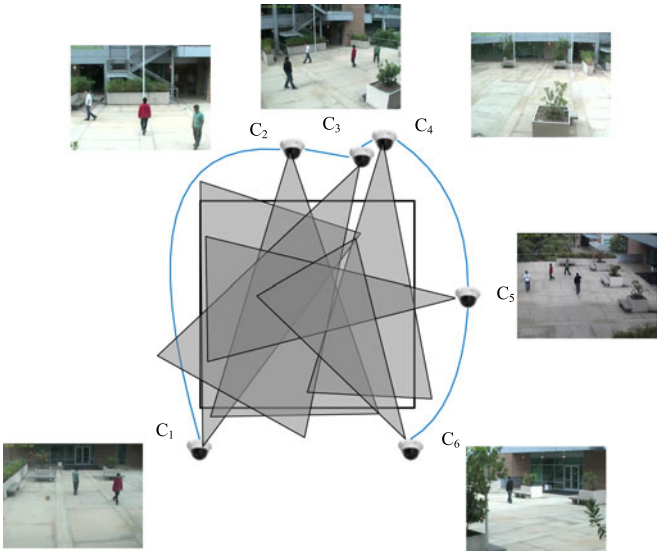

Fig. 1. In this figure, there are six sensing nodes, $C_1, C_2, \ldots, C_6$ observing an area (black rectangle) consisting of four targets. The solid blue lines show the communication channels between different nodes. This figure also depicts the presence of "naïve" nodes. For example, C_3, C_5, C_6 get direct measurements about the black target which it shares with its immediate network neighbors. However, C_1 does not have *direct* access to measurements of that target and thus is naïve w.r.t. that target's state.

of all targets, e.g., [1], do not work as well in camera network applications (see Fig. 1) where each agent directly observes only a small subset of the targets [2], [3]. This limits the observability at each agent to a subset of all the target states, even though a centralized solution, if implemented, would have full observability. In this paper, our goal is to design a distributed multi-target tracking scheme appropriate for applications where each sensor has a limited field-of-view (FOV), local communications, and data association challenges.

A distributed multi-target tracking problem can be divided into three sub-problems: data association (measurement to track association), information fusion, and dynamic state estimation. *Consensus algorithms* [1], [4] solve the distributed information fusion and state estimation problems at each sensor node using information only from that sensor and its communication network neighbors. Iteratively, each node can individually compute a global function of the prior state and measurement information of all the nodes (e.g., average). The important fact is that consensus is reached without all-to-all communication; thus consensus based frameworks do not require any specific communication network topology and are generally applicable to any arbitrary, connected network. The consensus estimates asymptotically converge to the global result. Due to the simplicity and robustness of consensus algorithms, they have been used in many applications, including estimation problems in sensor networks (e.g., [5], [6], [7]).

The Kalman Consensus Filter (KCF) [8] is a well-known consensus-based distributed state estimator. The KCF was originally designed for the scenario where each node has an observation of each target. The quality of neighboring node's prior information is not taken into account in the KCF. Thus, naive nodes may adversely affect the overall performance of the network. Moreover, the cross-covariance terms between the state estimates at different nodes were not incorporated in the KCF estimation process, because they are usually hard to compute in a distributed environment. Due to these reasons, the KCF performance deteriorates when applied within a camera network. Recently, the Information-weighted Consensus Filter (ICF) [2] was proposed to obtain optimal distributed state estimation performance in the presence of naivety.

The KCF and ICF assume that there is a single target, or for multiple targets that the measurement-to-track association is provided. For a multi-target tracking problem, the data association and the tracking steps are highly interdependent. The performance of tracking will affect the performance of data association and vice-versa. Thus, an integrated distributed tracking and data association solution is required where the uncertainty from the tracker can be incorporated in the data association process and vice-versa. Among many single-sensor multi-target data association frameworks, the Multiple Hypothesis Tracking (MHT) [9] and the Joint Probabilistic Data Association Filter JPDAF [10] are two dominant approaches. MHT usually achieves higher accuracy at the cost of higher computational load. Alternatively, JPDAF achieves reasonable results at lower computation cost. As distributed solutions are usually applied within low-power wireless sensor networks where the computational and communication power is limited, the JPDAF scheme will be utilized in the proposed distributed multi-target tracking framework herein. The proposed distributed tracking and data association framework is termed as the Multi-Target Information Consensus (MTIC).

The methods mentioned above are derived under the assumption that the observation model is linear. However, camera observation models are non-linear. Thus, to apply these distributed tracking algorithms in a realistic camera network, the nonlinearity must be addressed. In this paper, we show that the ICF and MTIC algorithm can be extended to these scenarios.

The main contribution of this paper is the tight integration of data association and distributed target tracking methods, taking special care of the issues of naivety and non-linearity in the observation model. Section 2 provides the problem formulation, along with a review of different consensus-based estimation methods. Section 3, reviews the JPDAF approach and extends it to a multi-sensor framework. Then the Multi-Target Information Consensus tracker is proposed. Section 4 proposes non-linear extensions to the ICF and MTIC algorithms. These algorithms are theoretically compared with each other and other algorithms. Finally, Section 6 provides simulation and testbed experimental evaluation.

1.2 Related Work

The purely decentralized nature of the fusion algorithm differentiates it from the majority of multi-camera tracking approaches in the computer vision literature. For example, in [11], a centralized approach for tracking in a multi-camera setup was proposed where the cameras were distributed spatially over a large area. In [12], an efficient target hand-off scheme was proposed, but no multi-camera

information fusion was involved. In [13], a multi-objective optimization based tracking framework was proposed which computes the optimal association of the measurements of each target over time. However, in this paper, the proposed data association strategy is probabilistic i.e., all measurements contribute to all the tracks depending on its association probability. This paper discusses the distributed multi-target tracking problem, where there is no centralized server, the processing is distributed over all the camera nodes and no target hand-off strategy is required. Distributed multi-target tracking in a network of cameras may be useful to various applications including collaborative tracking and camera control strategies as [14], [15].

Various distributed multi-target tracking methods [5], [16], [17], [18], [19] have been proposed in the sensor-networks literature, but they do not deal with special situations associated with camera networks, e.g., naivety and data association. In [16], a solution to the distributed data association problem was proposed by means of the message passing algorithm based on graphical models in which iterative, parallel exchange of information among the nodes viewing the same target was required. In the framework proposed herein, direct information exchange between nodes observing the same target is not required. In [17], the authors proposed a framework for querying a distributed database of video surveillance data to retrieve a set of likely paths of a person moving in the area under surveillance using a dynamic Bayesian Model. Unlike our proposed methods, this method does not deal with the distributed fusion of the information in the network.

In [5], [18], [19], the distributed multi-target tracking schemes did not account for naivety or the presence of cross-correlation between the estimates at different nodes. The method proposed herein build on the ICF [2], [3], which deals with both naivety and cross-correlation.

The MTIC algorithm, which is one of the methods we present in this paper had been introduced in [20]. In this paper, we show how the MTIC can be extended to handle non-linear models which is necessary for vision applications (Section 4). We also present in-depth proof and comparative analysis of all the state estimation methods in Section 5. Furthermore, detailed experimental results analyzing the performance of the proposed algorithms are presented.

1.3 Organization of the Paper

We start off in Section 2 with a description of the consensus-based distributed estimation framework and how it has been used for distributed state estimation in the Kalman Consensus Filter. This is followed by a review of our previous work on Information Weighted Consensus (ICF) which addresses the issue of naive nodes in the existing consensus approaches. Building upon these fundamentals, Section 3 presents the solution to the consensus problem when there are multiple targets and naive nodes. This calls for data association in the ICF framework. Algorithm 1 summarizes the Multi-Target Information Consensus approach. The approach thus far considered linear models. Section 4 considers the more realistic case of non-linear models and shows how the ICF and MTIC can be extended to these scenarios. Section 5 provides a comparison of the different state estimation approaches developed (i.e., KCF, ICF, MTIC and their non-linear versions). Experimental results in simulation and real-life data are presented in Section 6.

2 Distributed Estimation and Naïvety

2.1 Problem Formulation

Consider a network with N_C sensors/nodes. Communication in the network can be represented using an undirected connected graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$. The set $\mathcal{C} = \{C_1, C_2, \ldots, C_{N_C}\}$ contains the vertices of the graph, which represent the sensor nodes. The set $\mathcal{E}$ contains the edges of the graph, which represent the communication channels between different nodes. The set of nodes each having a direct communication channel with node C_i (sharing an edge with C_i) is represented by $\mathcal{N}_i$. There are $N_T(t)$ targets $\{T^1, T^2, \ldots T^{N_T}\}$ in the area viewable by the sensors at time t. It is assumed that N_C and $N_T(t)$ are known to each node.

The state of the jth target is represented by the vector $\mathbf{x}^j \in \mathcal{R}^p$. For example, for a tracking application, $\mathbf{x}^j$ might be a vector containing the ground plane position and velocity components of T^j. The state dynamics of target T^j are modeled as

$$\mathbf{x}^j(t+1) = \Phi \mathbf{x}^j(t) + \boldsymbol{\gamma}^j(t). \tag{1}$$

Here $\Phi \in \mathcal{R}^{p \times p}$ is the state transition matrix and the white process noise $\boldsymbol{\gamma}^j(t)$ is modeled as $\mathcal{N}(\mathbf{0}, \mathbf{Q}^j)$.

At time t, each node C_i, depending on its FOV and the location of the targets, gets $l_i(t)$ measurements denoted as $\{\mathbf{z}_i^n\}_{n=1}^{l_i(t)}$. The nodes do not know a priori, which measurement was generated from which target. Under the hypothesis that the observation $\mathbf{z}_i^n$ is generated from T^j, it is assumed that $\mathbf{z}_i^n$ was generated by the following observation model

$$\mathbf{z}_i^n = \mathbf{H}_i^j \mathbf{x}^j + \boldsymbol{\nu}_i^j. \tag{2}$$

Here, $\mathbf{H}_i^j \in \mathcal{R}^{m \times p}$ is the observation matrix for node C_i for T^j. The white noise $\boldsymbol{\nu}_i^j \in \mathcal{R}^m$ is modeled as $\mathcal{N}(\mathbf{0}, \mathbf{R}_i^j)$ with covariance $\mathbf{R}_i^j \in \mathcal{R}^{m \times m}$.

Each node also maintains a prior/predicted state estimate $\hat{\mathbf{x}}_i^{j-}(t)$ (and its covariance $\mathbf{P}_i^{j-}(t)$) for each target. The inverse of the state covariance matrix (information/precision matrix) will be denoted as $\mathbf{J}_i^j = (\mathbf{P}_i^j)^{-1}$. We assume that the initial prior state estimate and information matrix is available at each node for each target upon its detection. *Our goal is to track each target at each node, i.e., find the state estimate for each target at each node by using the prior and measurement information available in the entire network in a distributed fashion*. A critical step in this process is association of measurements with targets, which is addressed in this paper. We assume that the set of network measurements $\{\mathbf{z}_i^n\}_{i=1,\ldots,N_C}$ is sufficient that the state of all targets would be observable by a centralized estimator, if it existed. Because only a subset of the measurement information is available to each agent, the full state of all targets may not be locally observable by all agents using only their local information; nonetheless, the consensus-type algorithms that we present converge toward the optimal centralized solution.

2.2 Review of Distributed Estimation

2.2.1 Average Consensus

Average consensus [1] allows a group of distributed nodes to compute the arithmetic mean of some values $\{a_i\}_{i=1}^{N_C}$ using only local information communication between neighbors. Each node i begins with a quantity a_i and is interested in computing the average value of these quantities $\frac{1}{N_C}\sum_{i=1}^{N_C} a_i$ in a distributed manner.

Each node initializes its consensus state as $a_i(0) = a_i$. At the beginning of iteration k, a node C_i sends its previous state $a_i(k-1)$ to its immediate network neighbors $C_{i'} \in \mathcal{N}_i$ and also receives the neighbors' previous states $a_{i'}(k-1)$. Then it updates its own state information using the following equation

$$\begin{aligned} a_i(k) &= a_i(k-1) + \epsilon \sum_{i' \in \mathcal{N}_i} (a_{i'}(k-1) - a_i(k-1)) \\ &= \mathcal{A}(a_i(k-1)). \end{aligned} \tag{3}$$

Here $\mathcal{A}(a_i)$ is a shorthand mathematical operator for a single step of average consensus (defined by Eq. (3)). Iteration of $\mathcal{A}(a_i)$ causes all values of $a_i(k)$ to converge toward the average of the initial values. The average consensus algorithm can be used to compute the average of vectors and matrices by applying it to their individual elements separately. The rate parameter ϵ should be chosen in $(0, \frac{1}{\Delta_{max}})$, where Δ_{max} is the maximum degree of the network graph $\mathcal{G}$. Choosing larger values of ϵ will result in faster convergence, but choosing $\epsilon \geq \Delta_{max}$ will render the algorithm unstable. Average consensus assumes all agents have an estimate for all elements of a and that all estimates are of equal accuracy and uncorrelated. These assumptions do not apply within camera networks.

Consensus algorithms have been extended to perform various tasks in a network of agents such as linear algebraic operations: SVD, least squares, PCA, GPCA [7]. They have been applied in camera networks for distributed implementations of 3-D point triangulation, pose estimation [6], and action recognition [5]. The average consensus algorithm is applicable only for a static parameter estimation problem. For a dynamic state estimation problem, a predictor-corrector solution approach is necessary.

2.2.2 Kalman Consensus Filter

The Kalman Consensus Filter [8] uses the average consensus algorithm to achieve distributed dynamic state estimation. The KCF state estimation equations at the ith node are

$$\begin{aligned} \hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- &+ \left(\mathbf{J}_i^- + \mathbf{B}_i\right)^{-1}\left(\mathbf{b}_i - \mathbf{B}_i\hat{\mathbf{x}}_i^-\right) \\ &+ \frac{\epsilon}{1 + ||(\mathbf{J}_i^-)^{-1}||}(\mathbf{J}_i^-)^{-1} \sum_{i' \in \mathcal{N}_i}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right) \end{aligned} \tag{4}$$

$$\mathbf{J}_i^+ = \mathbf{J}_i^- + \mathbf{B}_i \tag{5}$$

where,

$$\mathbf{b}_i = \sum_{i' \in \mathcal{N}_i \cup i} \mathbf{H}_{i'}^T {\mathbf{R}_{i'}}^{-1} \mathbf{z}_{i'}, \quad \mathbf{B}_i = \sum_{i' \in \mathcal{N}_i \cup i} \mathbf{H}_{i'}^T {\mathbf{R}_{i'}}^{-1} \mathbf{H}_{i'} \tag{6}$$

$\hat{\mathbf{x}}_i^+$ is the posterior state estimate and $\mathbf{J}_i^+$ is the state information matrix. In Eqn. (4), the first term is the prior state estimate, the second term is the innovation from the measurements in the local neighborhood of the node and the third term is an averaging term over the priors in the local neighborhood. Note that each neighbor's prior $\hat{\mathbf{x}}_{i'}^-$ is considered equally informative in the estimation process. The KCF works well in situations where the state is observable at each node. When naive nodes are present, this equal weighting can lead to poor estimation performance, due to the difference in information content.

2.2.3 Information Weighted Consensus

The Information-weighted Consensus Filter algorithm [2], [3] is a distributed state estimation framework that accounts for the naivety issue and can achieve optimal performance equivalent to a centralized solution. The prior information $\{\hat{\mathbf{x}}_i^-, \mathbf{J}_i^-\}$ and measurement information $\{\mathbf{z}_i, \mathbf{R}_i\}$ are first fused into an information vector $\mathbf{v}_i \in \mathcal{R}^p$ and an information matrix $\mathbf{V}_i \in \mathcal{R}^{p \times p}$ at each node:

$$\mathbf{v}_i = \frac{1}{N_C}\mathbf{J}_i^-(t)\hat{\mathbf{x}}_i^-(t) + \mathbf{u}_i \tag{7}$$

$$\mathbf{V}_i = \frac{1}{N_C}\mathbf{J}_i^-(t) + \mathbf{U}_i, \tag{8}$$

where $\mathbf{u}_i = \mathbf{H}_i^T {\mathbf{R}_i}^{-1}\mathbf{z}_i$ and $\mathbf{U}_i = \mathbf{H}_i^T {\mathbf{R}_i}^{-1}\mathbf{H}_i$. Next, using the average consensus algorithm, the average of these vectors and matrices are computed at each node over the network as $\bar{\mathbf{v}}$ and $\bar{\mathbf{V}}$. Finally, the optimal state estimate and its information matrix are computed as,

$$\hat{\mathbf{x}}_i^+(t) = \bar{\mathbf{V}}^{-1}\bar{\mathbf{v}}, \qquad \mathbf{J}_i^+(t) = N_C\bar{\mathbf{V}}. \tag{9}$$

One reason that the ICF performance is not affected by naivety is that the prior information state of each node is appropriately weighted by the prior information matrix $\mathbf{J}_i^-$ at that node before sending it to the neighbors. Thus a node which has less information about a target's state is given less weight in the overall estimation process. As proved in [2], [3], the ICF converges to the maximum a posteriori estimate of the centralized solution. However, the ICF algorithm in [2], [3] assumes that the data association problem is solved when there are multiple targets in the scene.

3 Distributed Multi-Target Tracking

The goal of this section is to extend the ICF approach to a distributed multi-target tracking scheme. Distributed multi-target tracking problems contain three sub-problems: data association (measurement to track association), distributed information fusion, and dynamic state estimation. Because the tracking and data association problems are highly interdependent, an integrated distributed solution is required where the uncertainty from the tracker can be incorporated in the data association process and vice-versa.

3.1 Review of JPDAF

Realistic multi-target state estimation problems requires solving of the data association, which is itself a challenging

problem, even in the centralized case. JPDAF [10] is a centralized processing algorithm, thus the agent index i is unnecessary and will be dropped. A double superscript is required for the hypothesis that measurement $\mathbf{z}^n$ is associated with target T^j. At time t, the measurement innovation $\tilde{\mathbf{z}}^{jn}$ and the innovation covariance $\mathbf{S}^j$ of measurement $\mathbf{z}^n$ for target T^j is computed as,

$$\tilde{\mathbf{z}}^{jn} = \mathbf{z}^n - \mathbf{H}^j\hat{\mathbf{x}}^{j-} \tag{10}$$

$$\mathbf{S}^j = \mathbf{H}^j(\mathbf{P}^{j-})(\mathbf{H}^j)^T + \mathbf{R}^j. \tag{11}$$

Because the JPDAF algorithm computes a probability weighted mean measurement from all the available measurements for each target, the index in Eq. (11) is j instead of n. The probability that T^j is the correct target to associate with $\mathbf{z}^n$ is β^{jn} and the probability that none of the measurements belong to T^j is β^{j0}. See [10] for details about computing these probabilities.

The Kalman gain $\mathbf{K}^j$, mean measurement $\mathbf{y}^j$ and mean measurement innovation $\tilde{\mathbf{y}}^j$ for target T^j are

$$\mathbf{K}^j = (\mathbf{P}^{j-})(\mathbf{H}^j)^T(\mathbf{S}^j)^{-1}, \tag{12}$$

$$\mathbf{y}^j = \sum_{n=1}^{l}\beta^{jn}\mathbf{z}^n, \tag{13}$$

$$\tilde{\mathbf{y}}^j = \sum_{n=1}^{l}\beta^{jn}\tilde{\mathbf{z}}^{jn} = \mathbf{y}^j - (1-\beta^{j0})\mathbf{H}^j\hat{\mathbf{x}}^{j-}. \tag{14}$$

The JPDAF state estimate and its covariance are

$$\hat{\mathbf{x}}^{j+}(t) = \hat{\mathbf{x}}^{j-}(t) + \mathbf{K}^j\tilde{\mathbf{y}}^j \tag{15}$$

$$\mathbf{P}^{j+}(t) = \mathbf{P}^{j-}(t) - (1-\beta^{j0})\mathbf{K}^j\mathbf{S}^j\left(\mathbf{K}^j\right)^T + \mathbf{K}^j\tilde{\mathbf{P}}^j\left(\mathbf{K}^j\right)^T, \tag{16}$$

where

$$\tilde{\mathbf{P}}^j = \left(\sum_{n=1}^{l}\beta^{jn}\tilde{\mathbf{z}}^{jn}\left(\tilde{\mathbf{z}}^{jn}\right)^T\right) - \tilde{\mathbf{y}}^j\left(\tilde{\mathbf{y}}^j\right)^T. \tag{17}$$

3.2 Information Form JPDAF

The JPDAF algorithm in the information form will be useful in the next section to derive the distributed multi-target tracking algorithm. For multiple sensors, the JPDAF algorithm described in Eqs. (15)-(16) can be written in the information form as (see Section 1 in supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2484339):

$$\begin{aligned}\hat{\mathbf{x}}^{j+} = &\left(\mathbf{J}^{j-} + \mathbf{U}^j\right)^{-1}\\ &\left(\mathbf{J}^{j-}\hat{\mathbf{x}}^{j-} + \mathbf{u}^j + \beta^{j0}\mathbf{U}^j\hat{\mathbf{x}}^{j-}\right)\end{aligned} \tag{18}$$

$$\mathbf{J}^{j+} = \mathbf{J}^{j-} + \mathbf{G}^j \tag{19}$$

where,

$$\mathbf{G}^j = \mathbf{J}^{j-}\mathbf{K}^j\left(\left(\mathbf{C}^j\right)^{-1} - \mathbf{K}^{jT}\mathbf{J}^{j-}\mathbf{K}^j\right)^{-1}\mathbf{K}^{jT}\mathbf{J}^{j-} \tag{20}$$

$$\mathbf{C}^j = (1-\beta^{j0})\mathbf{S}^j - \tilde{\mathbf{P}}^j \tag{21}$$

$$\mathbf{u}^j = \mathbf{H}^{jT}\mathbf{R}^{j-1}\mathbf{y}^j \quad \text{and} \quad \mathbf{U}^j = \mathbf{H}^{jT}\mathbf{R}^{j-1}\mathbf{H}^j. \tag{22}$$

Note that in Eq. (18), $\mathbf{J}^{j-}\hat{\mathbf{x}}^{j-}$ is the weighted prior information and $\mathbf{u}^j + \beta^{j0}\mathbf{U}^j\hat{\mathbf{x}}^{j-}$ is the weighted measurement information (taking data association uncertainty β^{j0} into account). The sum of these two terms represents the total available information for the single agent case. To incorporate independent measurement information from an additional agent, the weighted measurement information from that agent has to be added to this summation. Similarly, the information matrices ($\mathbf{U}_i^j$ and $\mathbf{G}_i^j$) from additional agents must also be added to the appropriate terms. This gives us the multi-agent centralized estimate in the information form

$$\begin{aligned}\hat{\mathbf{x}}^{j+} = &\left(\mathbf{J}^{j-} + \sum_{i=1}^{N_C}\mathbf{U}_i^j\right)^{-1}\\ &\left(\mathbf{J}^{j-}\hat{\mathbf{x}}^{j-} + \sum_{i=1}^{N_C}\left(\mathbf{u}_i^j + \beta_i^{j0}\mathbf{U}_i^j\hat{\mathbf{x}}^{j-}\right)\right),\end{aligned} \tag{23}$$

$$\mathbf{J}^{j+} = \mathbf{J}^{j-} + \sum_{i=1}^{N_C}\mathbf{G}_i^j. \tag{24}$$

Our goal is to compute the quantities in Eq. (23) and (24) in a distributed manner.

3.3 Multi-Target Information Consensus

This section builds on the ICF and JPDAF to derive a distributed multi-target tracking algorithm that will be referred to as the Multi Target Information Consensus tracker. As we have multiple agents, we will bring back the agent index in the subscripts.

Eqs. (23) and (24) can be manipulated as follows:

$$\begin{aligned}\hat{\mathbf{x}}_i^{j+} &= \left(\sum_{i=1}^{N_C}\left(\frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{U}_i^j\right)\right)^{-1}\\ &\quad\sum_{i=1}^{N_C}\left(\mathbf{u}_i^j + \left(\frac{\mathbf{J}_i^{j-}}{N_C} + \beta_i^{j0}\mathbf{U}_i^j\right)\hat{\mathbf{x}}_i^{j-}\right)\\ &= \left(\sum_{i=1}^{N_C}\mathbf{V}_i^j\right)^{-1}\sum_{i=1}^{N_C}\mathbf{v}_i^j\\ &= \left(\frac{\sum_{i=1}^{N_C}\mathbf{V}_i^j}{N_C}\right)^{-1}\frac{\sum_{i=1}^{N_C}\mathbf{v}_i^j}{N_C}\end{aligned} \tag{25}$$

$$\begin{aligned}\mathbf{J}_i^{j+} &= \sum_{i=1}^{N_C}\left(\frac{\mathbf{J}^{j-}}{N_C} + \mathbf{G}_i^j\right)\\ &= \sum_{i=1}^{N_C}\mathbf{W}_i^j\\ &= N_C\frac{\sum_{i=1}^{N_C}\mathbf{W}_i^j}{N_C},\end{aligned} \tag{26}$$

where,

$$\mathbf{V}_i^j = \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{U}_i^j, \qquad \mathbf{W}_i^j = \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{G}_i^j$$
$$\text{and} \qquad \mathbf{v}_i^j = \mathbf{u}_i^j + \left(\frac{\mathbf{J}_i^{j-}}{N_C} + \beta_i^{j0}\mathbf{U}_i^j\right)\hat{\mathbf{x}}_i^{j-}. \tag{27}$$

In a distributed system, each node C_i will have its own prior information $\{\hat{\mathbf{x}}_i^{j-}, \mathbf{J}_i^{j-}\}$ for T^j; therefore, the averages $\bar{\mathbf{v}}^j$, $\bar{\mathbf{V}}^j$, $\bar{\mathbf{W}}^j$ of each of the three quantities defined in Eq. (27) can be computed, in a distributed manner, using the average consensus algorithm [1]. This allows each agent to compute: $\hat{\mathbf{x}}_i^{j+} = \left(\bar{\mathbf{V}}^j\right)^{-1}\bar{\mathbf{v}}^j$ and $\mathbf{J}_i^{j+} = N_C\bar{\mathbf{W}}^j$ in a distributed manner.

This distributed implementation of the centralized JPDAF algorithm is the MTIC algorithm which is summarized in Algorithm 1. Note that if a node does not get any measurement for T^j, i.e., $\beta_i^{j0} = 1$, $\mathbf{u}_i^j$, $\mathbf{U}_i^j$ and $\mathbf{G}_i^j$ are set to zero vectors and matrices respectively.

Algorithm 1. MTIC for Target T^j at Node C_i at Time Step t

Input: $\hat{\mathbf{x}}_i^{j-}(t)$, $\mathbf{J}_i^{j-}(t)$, $\mathbf{H}_i^j$, $\mathbf{R}_i^j$.

1) Get measurements: $\{\mathbf{z}_i^n\}_{n=1}^{l_i(t)}$

2) Compute $\mathbf{S}_i^j$, $\mathbf{y}_i^j$, β_i^{j0}, $\mathbf{K}_i^j$ and $\mathbf{C}_i^j$ (Eqn. (21))

3) Compute information vector and matrices:

$$\mathbf{u}_i^j \leftarrow \mathbf{H}_i^{jT}\mathbf{R}_i^{j^{-1}}\mathbf{y}_i^j$$
$$\mathbf{U}_i^j \leftarrow \mathbf{H}_i^{jT}\mathbf{R}_i^{j^{-1}}\mathbf{H}_i^j$$
$$\mathbf{G}_i^j \leftarrow \mathbf{J}_i^{j-}\mathbf{K}_i^j\left(\mathbf{C}_i^{j^{-1}} - \mathbf{K}_i^{jT}\mathbf{J}_i^{j-}\mathbf{K}_i^j\right)^{-1}\mathbf{K}_i^{jT}\mathbf{J}_i^{j-}$$

4) Initialize consensus data

$$\mathbf{v}_i^j[0] \leftarrow \mathbf{u}_i^j + \left(\frac{\mathbf{J}_i^{j-}}{N_C} + \beta_i^{j0}\mathbf{U}_i^j\right)\hat{\mathbf{x}}_i^{j-}$$
$$\mathbf{V}_i^j[0] \leftarrow \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{U}_i^j$$
$$\mathbf{W}_i^j[0] \leftarrow \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{G}_i^j$$

5) Perform average consensus (Section 2.2.1) on $\mathbf{v}_i^j[0]$, $\mathbf{V}_i^j[0]$ and $\mathbf{W}_i^j[0]$ independently for K iterations.

6) Estimate:

$$\hat{\mathbf{x}}_i^{j+} \leftarrow \left(\mathbf{V}_i^j[K]\right)^{-1}\mathbf{v}_i^j[K]$$
$$\mathbf{J}_i^{j+} \leftarrow N_C\mathbf{W}_i^j[K] \tag{28}$$

7) Predict:

$$\hat{\mathbf{x}}_i^{j-}(t+1) \leftarrow \mathbf{\Phi}\hat{\mathbf{x}}_i^{j+}(t)$$
$$\mathbf{J}_i^{j-}(t+1) \leftarrow \left(\mathbf{\Phi}\left(\mathbf{J}_i^{j+}(t)\right)^{-1}\mathbf{\Phi}^T + \mathbf{Q}\right)^{-1}$$

Output: $\hat{\mathbf{x}}_i^{j+}(t)$, $\mathbf{J}_i^{j+}(t)$, $\hat{\mathbf{x}}_i^{j-}(t+1)$, $\mathbf{J}_i^{j-}(t+1)$.

Note that at each time step, the nodes iteratively converge toward the averages $\bar{\mathbf{v}}^j$, $\bar{\mathbf{V}}^j$, $\bar{\mathbf{W}}^j$ and only compute the posterior estimate and information matrix at the conclusion of the consensus iterations for that time step. If convergence was complete, the converged quantities would be equal to the centralized estimate and information matrix. Two special situations are of interest.

1) First, when a new target is detected, many or most agents will have zero information while those (few) agents that detect the new target have only their new measurement information. The cross-correlation between the estimates of the different agents is zero (correlation coefficient of 0). The information weighting in MTIC correctly accounts for this quite common occurrence.

2) Second, after consensus is achieved, the total information in the network about T^j is $\mathbf{J}^{j+}$, which MTIC accounts for as $\mathbf{W}^{j+}$ information stored by each of N_C agents. Because the agents all have the same estimate and information, the correlation is perfect (correlation coefficient of 1). Completely accounting for the cross correlation matrices for all target estimates across all agents would be a very large computational and communication burden, which is unnecessary because the consensus algorithm rapidly drives the correlation coefficient from zero to one. Simulations in [3] evaluate ICF performance relative to that of a centralized estimator versus the number of consensus iterations (per time step), which demonstrate these issues.

4 ICF with Non-Linear Models

The ICF and MTIC assumed a linear model. In many applications, the observation is a non-linear function of the target state. This is true for cameras because the position of a point in the camera pixel coordinate system is a non-linear function of the position of the point in the world coordinate system [21]. For the algorithms derived in the previous sections to be applicable in such scenarios, we need to extend them to non-linear observation models. We call these algorithms, the Extended ICF (EICF) and the Extended MTIC (EMTIC).

4.1 Extended ICF

Considering the data association is known, the measurement at each node can be expressed using the non-linear relation as,

$$\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \boldsymbol{\nu}_i. \tag{29}$$

Using first-order Taylor series approximation, the linearized observation matrix can be written as,

$$\mathbf{H}_i^j = \nabla_{\mathbf{x}^j}\mathbf{h}_i(\mathbf{x}^j)|_{\mathbf{x}^j=\hat{\mathbf{x}}_i^{j-}}. \tag{30}$$

The collection of all measurements from all nodes can be expressed as,

$$\mathcal{Z} = \mathbf{h}_c(\mathbf{x}) + \boldsymbol{\nu}, \tag{31}$$

where $\mathcal{Z} = [\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots \mathbf{z}_{N_C}^T]^T \in \mathcal{R}^m$ is the concatenation of all measurements in the network, $m = \sum_{i=1}^{N_c} m_i$ is the individual measurement vector length, and m_i is the dimension of $\mathbf{z}_i$. The central non-linear observation function $\mathbf{h}_c()$ is a stack of all the functions from individual sensors such that $\mathbf{h}_c(\mathbf{x}) = \{\mathbf{h}_1(\mathbf{x})^T, \mathbf{h}_2(\mathbf{x})^T, \ldots \mathbf{h}_{N_C}(\mathbf{x})^T\}^T$ (a vector of vector valued functions). We will represent the stack of linearized

observation matrices as $\mathcal{H} = [\mathbf{H}_1^T, \mathbf{H}_2^T, \ldots \mathbf{H}_N^T]^T \in \mathcal{R}^{mN_C \times p}$. For the measurement noise vector, $\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T, \boldsymbol{\nu}_2^T, \ldots \boldsymbol{\nu}_N^T]^T \in \mathcal{R}^{mN_C}$, we denote its covariance as $\mathcal{R} \in \mathcal{R}^{mN_C \times mN_C}$. We assume the measurement noise to be uncorrelated across nodes. Thus, the measurement covariance matrix is $\mathcal{R} = diag(\mathbf{R}_1, \mathbf{R}_2, \ldots \mathbf{R}_N)$. Let us denote $\mathbf{U}_c = \mathcal{H}^T\mathcal{R}^{-1}\mathcal{H}$ and $\mathbf{u}_c = \mathcal{H}^T\mathcal{R}^{-1}\mathcal{Z}$. The subscript c stands for "centralized". Let us also denote the centralized predicted measurement, $\mathbf{h}_c(\hat{\mathbf{x}}_c^-) = \{\mathbf{h}_1(\hat{\mathbf{x}}_c^-)^T, \mathbf{h}_2(\hat{\mathbf{x}}_c^-)^T, \ldots \mathbf{h}_{N_C}(\hat{\mathbf{x}}_c^-)^T\}^T$.

The state estimation equations from the centralized Extended Kalman Filter (EKF) [22] algorithm can be written as (see Section 2 in supplementary materials, available online),

$$\begin{aligned}\hat{\mathbf{x}}_c^+ = &\left(\mathbf{J}_c^- + \mathbf{U}_c\right)^{-1} \\ &\left(\mathbf{J}_c^-\hat{\mathbf{x}}_c^- + \mathbf{u}_c + \mathcal{H}^T\mathcal{R}^{-1}(\mathcal{H}\hat{\mathbf{x}}_c^- - \mathbf{h}_c(\hat{\mathbf{x}}_c^-))\right)\end{aligned} \tag{32}$$

$$\mathbf{J}_c^+ = \left(\mathbf{J}_c^- + \mathbf{U}_c\right). \tag{33}$$

Given that all the nodes have reached consensus on the previous time step, we have, $\hat{\mathbf{x}}_i^- = \hat{\mathbf{x}}_c^-$ and $\mathbf{J}_i^- = \mathbf{J}_c^-$ for all i. This implies,

$$\mathbf{J}_c^- = \sum_{i=1}^{N_C} \frac{\mathbf{J}_i^-}{N_C} \quad \text{and} \quad \mathbf{J}_c^-\hat{\mathbf{x}}_c^- = \sum_{i=1}^{N_C} \frac{\mathbf{J}_i^-}{N_C}\hat{\mathbf{x}}_i^-.$$

Also, as $\mathbf{R}_c$ is block diagonal, we have,

$$\mathbf{U}_c = \sum_{i=1}^{N_C} \mathbf{U}_i, \quad \text{and} \quad \mathbf{u}_c = \sum_{i=1}^{N_C} \mathbf{u}_i,$$

and

$$\mathcal{H}^T\mathcal{R}^{-1}(\mathcal{H}\hat{\mathbf{x}}_c^- - \mathbf{h}_c(\hat{\mathbf{x}}_c^-)) = \sum_{i=1}^{N_C} \mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{H}_i\hat{\mathbf{x}}_i^- - \mathbf{h}_i(\hat{\mathbf{x}}_i^-)).$$

Thus, from Eqs. (32) and (33) we get,

$$\begin{aligned}\hat{\mathbf{x}}_c^+ = &\left(\sum_{i=1}^{N_C}\left(\frac{\mathbf{J}_i^-}{N_C} + \mathbf{U}_i\right)\right)^{-1} \\ &\sum_{i=1}^{N_C}\left(\mathbf{J}_i^-\hat{\mathbf{x}}_i^- + \mathbf{u}_i + \mathbf{H}_i^T\mathbf{R}_i^{-1}\left(\mathbf{H}_i\hat{\mathbf{x}}_i^- - \mathbf{h}_i(\hat{\mathbf{x}}_i^-)\right)\right)\end{aligned} \tag{34}$$

$$\mathbf{J}_c^+ = \sum_{i=1}^{N_C}\left(\frac{\mathbf{J}_i^-}{N_C} + \mathbf{U}_i\right). \tag{35}$$

The above equations can be computed in a distributed manner by initializing $\mathbf{v}_i[0]$ and $\mathbf{V}_i[0]$ as the following at each node and running average consensus algorithm on them:

$$\begin{aligned}\mathbf{v}_i[0] &= \frac{\mathbf{J}_i^-}{N_C}\hat{\mathbf{x}}_i^- + \mathbf{u}_i \\ &\quad + \mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{H}_i\hat{\mathbf{x}}_i^- - \mathbf{h}_i(\hat{\mathbf{x}}_i^-)) \\ \mathbf{V}_i[0] &= \frac{\mathbf{J}_i^-}{N_C} + \mathbf{U}_i.\end{aligned} \tag{36}$$

This leads to the EICF algorithm and is summarized in Algorithm 2.

Algorithm 2. EICF at Node C_i at Time Step t

Input: prior state estimate $\hat{\mathbf{x}}_i^-(t)$, prior information matrix $\mathbf{J}_i^-(t)$, consensus speed factor ϵ and total number of consensus iterations K.

1) Linearize $\mathbf{h}_i$ at $\hat{\mathbf{x}}_i^-(t)$ to compute $\mathbf{H}_i$

2) Get measurement $\mathbf{z}_i$ with covariance $\mathbf{R}_i$

3) Compute consensus proposals,

$$\begin{aligned}\mathbf{V}_i[0] &\leftarrow \frac{1}{N_C}\mathbf{J}_i^-(t) + \mathbf{U}_i \\ \mathbf{v}_i[0] &\leftarrow \frac{1}{N_C}\mathbf{J}_i^-(t)\hat{\mathbf{x}}_i^-(t) + \mathbf{u}_i \\ &\quad + \mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{H}_i\hat{\mathbf{x}}_i^- - \mathbf{h}_i(\hat{\mathbf{x}}_i^-))\end{aligned}$$

4) Perform average consensus (Section 2.2.1) on $\mathbf{v}_i[0]$, $\mathbf{V}_i[0]$ independently for K iterations.

5) Compute a posteriori state estimate and information matrix for time t

$$\begin{aligned}\hat{\mathbf{x}}_i^+(t) &\leftarrow (\mathbf{V}_i[K])^{-1}\mathbf{v}_i[K] \\ \mathbf{J}_i^+(t) &\leftarrow N_C\mathbf{V}_i[K]\end{aligned} \tag{37}$$

6) Predict for next time step $(t+1)$

$$\begin{aligned}\mathbf{J}_i^-(t+1) &\leftarrow \left(\Phi(\mathbf{J}_i^+(t))^{-1}\Phi^T + \mathbf{Q}\right)^{-1} \\ \hat{\mathbf{x}}_i^-(t+1) &\leftarrow \Phi\hat{\mathbf{x}}_i^+(t)\end{aligned}$$

Output: EICF estimate $\hat{\mathbf{x}}_i^+(t)$, $\mathbf{J}_i^+(t)$, $\hat{\mathbf{x}}_i^-(t+1)$, $\mathbf{J}_i^-(t+1)$.

Note that for a linear observation model, the EICF reduces to the original ICF in Eqs. (7)-(8). Also, the EICF has the additional $\mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{H}_i\hat{\mathbf{x}}_i^- - \mathbf{h}_i(\hat{\mathbf{x}}_i^-))$ term at each node compared to the ICF in the information form just as the EKF has the additional term $\mathcal{H}^T\mathcal{R}^{-1}\left(\mathcal{H}\hat{\mathbf{x}}_c^- - \mathbf{h}_c\left(\hat{\mathbf{x}}_c^-\right)\right)$ compared to the KF.

4.2 Extended Multi-Target Information Consensus

This section extends the MTIC algorithm to handle non-linear sensing models. Equivalently, it extends the EICF to scenarios where the data association is not known a priori.

For a non-linear sensing model $\mathbf{h}()$, the measurement innovation term in Eq. (14) becomes,

$$\tilde{\mathbf{y}}^j = \mathbf{y}^j - (1 - \beta^{j0})\mathbf{h}(\hat{\mathbf{x}}^{j-}). \tag{38}$$

Using this, the state estimation equations for a centralized approach can be written as follows (see Section 3 in supplementary materials, available online):

$$\begin{aligned}\hat{\mathbf{x}}_c^{j+} = &\left(\mathbf{J}_c^{j-} + \sum_{i=1}^{N_C}\mathbf{U}_i^j\right)^{-1}\left(\mathbf{J}_i^{j-}\hat{\mathbf{x}}_i^{j-} + \sum_{i=1}^{N_C}\left[\mathbf{u}_i^j\right.\right. \\ &\left.\left. + \mathbf{H}_i^{jT}\mathbf{R}_i^{j-1}(\mathbf{H}_i^j\hat{\mathbf{x}}_i^{j-} - (1 - \beta_i^{j0})\mathbf{h}_i(\hat{\mathbf{x}}_i^{j-}))\right]\right)\end{aligned} \tag{39}$$

$$\mathbf{J}_c^{j+} = \mathbf{J}_i^{j-} + \sum_{i=1}^{N_C}\mathbf{G}_i^j.$$

Given that all the nodes have reached consensus on the previous time step (discussed in details in Section 3.3), we have, $\hat{\mathbf{x}}_i^{j-} = \hat{\mathbf{x}}_c^{j-}$ and $\mathbf{J}_i^{j-} = \mathbf{J}_c^{j-}$ for all i. This implies, $\mathbf{J}_c^{j-} = \sum_{i=1}^{N_C} \frac{\mathbf{J}_i^{j-}}{N_C}$ and $\mathbf{J}_c^{j-}\hat{\mathbf{x}}_c^{j-} = \sum_{i=1}^{N_C} \frac{\mathbf{J}_i^{j-}}{N_C}\hat{\mathbf{x}}_i^{j-}$. Thus we can write,

$$\begin{aligned}\hat{\mathbf{x}}_c^{j+} &= \left(\sum_{i=1}^{N_C}\left(\frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{U}_i^j\right)\right)^{-1}\sum_{i=1}^{N_C}\left(\frac{\mathbf{J}_i^{j-}}{N_C}\hat{\mathbf{x}}_i^{j-} + \mathbf{u}_i^j \right.\\ &\left. + \mathbf{H}_i^{jT}\mathbf{R}_i^{j-1}\left[\mathbf{H}_i^j\hat{\mathbf{x}}_i^{j-} - (1-\beta_i^{j0})\mathbf{h}_i(\hat{\mathbf{x}}_i^{j-})\right]\right) \\ \mathbf{J}_c^{j+} &= \sum_{i=1}^{N_C}\left(\frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{G}_i^j\right).\end{aligned} \tag{40}$$

Thus, for EMTIC, the initial consensus variables are

$$\begin{aligned}\mathbf{v}_i^j[0] &= \frac{\mathbf{J}_i^{j-}}{N_C}\hat{\mathbf{x}}_i^{j-} + \mathbf{u}_i^j \\ &\quad + \mathbf{H}_i^{jT}\mathbf{R}_i^{j-1}\left(\mathbf{H}_i^j\hat{\mathbf{x}}_i^{j-} - (1-\beta_i^{j0})\mathbf{h}_i(\hat{\mathbf{x}}_i^{j-})\right) \\ \mathbf{V}_i^j[0] &= \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{U}_i^j, \\ \mathbf{W}_i^j[0] &= \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{G}_i^j.\end{aligned} \tag{41}$$

Note that for a linear observation model the non-linear MTIC equations reduce to the original MTIC. The EMTIC algorithm is summarized as Algorithm 3.

5 Comparison of KCF, ICF, MTIC, EKCF, EICF and EMTIC

In Table 1, we compare the state estimation equations of the KCF, ICF, MTIC, EKCF, EICF and EMTIC for one particular target and a single consensus iteration step. The derivation of these particular forms is shown in Section 4 in supplementary materials, available online.

Note that the differences in the prior states between the neighboring nodes, $\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-$ are weighted by the corresponding neighbor's prior information matrix $\mathbf{J}_{i'}^-$ in ICF, MTIC, EICF and EMTIC. This handles the issue with naivety as the innovation from a naive neighbor's prior state will be given less weight. In KCF and EKCF, the innovation from each neighbor's prior is given equal weight which may yield poor performance in the presence of naive nodes.

The term $\mathbf{u}_i$, in Eqs. (45) and (47) are not exactly the same. As ICF assumes perfect data association and computes $\mathbf{u}_i$ from the appropriate measurement $\mathbf{z}_i^j$, whereas, in MTIC, $\mathbf{u}_i$ is computed from the mean measurement $\mathbf{y}_i^j$. The same argument holds for Eq. (51) and Eq. (53) of the EICF and EMTIC, respectively.

In MTIC Eq. (47), the $\mathcal{A}(\mathbf{U}_i\hat{\mathbf{x}}_i^- - (1-\beta_{i0})\mathbf{U}_i\hat{\mathbf{x}}_i^-)$ term arises due to the data association error. In EICF Eq. (51) the $\mathcal{A}(\mathbf{U}_i\hat{\mathbf{x}}_i^- - \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{h}_i(\hat{\mathbf{x}}_i^-))$ term comes from nonlinearity in the observation model. In a similar fashion, in EMTIC Eq. (53), the $\mathcal{A}\big(\mathbf{U}_i\hat{\mathbf{x}}_i^- - (1-\beta_i^0)\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{h}_i(\hat{\mathbf{x}}_i^-)\big)$ term is a result of both data association error and nonlinearity in the observation model. It can be seen that, with perfect data association and a linear observation model, this term is equal to zero, yielding ICF Eq. (45).

Algorithm 3. EMTIC for target T^j at node C_i at time step t

Input: $\hat{\mathbf{x}}_i^{j-}(t)$, $\mathbf{J}_i^{j-}(t)$, $\mathbf{h}_i$, $\mathbf{R}_i^j$.

1) Linearize $\mathbf{h}_i$ at $\hat{\mathbf{x}}_i^{j-}(t)$ to compute $\mathbf{H}_i^j$

2) Get measurements: $\{\mathbf{z}_i^n\}_{n=1}^{l_i(t)}$

3) Compute $\mathbf{S}_i^j$, $\mathbf{y}_i^j$, β_i^{j0}, $\mathbf{K}_i^j$ and $\mathbf{C}_i^j$

4) Compute information vector and matrices:

$$\begin{aligned}\mathbf{u}_i^j &\leftarrow \mathbf{H}_i^{jT}\mathbf{R}_i^{j-1}\mathbf{y}_i^j \\ \mathbf{U}_i^j &\leftarrow \mathbf{H}_i^{jT}\mathbf{R}_i^{j-1}\mathbf{H}_i^j \\ \mathbf{G}_i^j &\leftarrow \mathbf{J}_i^{j-}\mathbf{K}_i^j\left(\mathbf{C}_i^{j-1} - \mathbf{K}_i^{jT}\mathbf{J}_i^{j-}\mathbf{K}_i^j\right)^{-1}\mathbf{K}_i^{jT}\mathbf{J}_i^{j-}\end{aligned}$$

5) Initialize consensus data

$$\begin{aligned}\mathbf{v}_i^j[0] &\leftarrow \frac{\mathbf{J}_i^{j-}}{N_C}\hat{\mathbf{x}}_i^{j-} + \mathbf{u}_i^j \\ &\quad + \mathbf{H}_i^{jT}\mathbf{R}_i^{j-1}\left(\mathbf{H}_i^j\hat{\mathbf{x}}_i^{j-} - (1-\beta_i^{j0})\mathbf{h}_i(\hat{\mathbf{x}}_i^{j-})\right) \\ \mathbf{V}_i^j[0] &\leftarrow \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{U}_i^j \\ \mathbf{W}_i^j[0] &\leftarrow \frac{\mathbf{J}_i^{j-}}{N_C} + \mathbf{G}_i^j\end{aligned}$$

6) Perform average consensus (Section 2.2.1) on $\mathbf{v}_i^j[0]$, $\mathbf{V}_i^j[0]$ and $\mathbf{W}_i^j[0]$ independently for K iterations.

7) Estimate:

$$\begin{aligned}\hat{\mathbf{x}}_i^{j+} &\leftarrow \left(\mathbf{V}_i^j[K]\right)^{-1}\mathbf{v}_i^j[K] \\ \mathbf{J}_i^{j+} &\leftarrow N_C\mathbf{W}_i^j[K]\end{aligned} \tag{42}$$

8) Predict:

$$\begin{aligned}\hat{\mathbf{x}}_i^{j-}(t+1) &\leftarrow \mathbf{\Phi}\hat{\mathbf{x}}_i^{j+}(t) \\ \mathbf{J}_i^{j-}(t+1) &\leftarrow \left(\mathbf{\Phi}\left(\mathbf{J}_i^{j+}(t)\right)^{-1}\mathbf{\Phi}^T + \mathbf{Q}^j\right)^{-1}\end{aligned}$$

Output: $\hat{\mathbf{x}}_i^{j+}(t)$, $\mathbf{J}_i^{j+}(t)$, $\hat{\mathbf{x}}_i^{j-}(t+1)$, $\mathbf{J}_i^{j-}(t+1)$.

The information matrix update equations, i.e., Eqs. (46) and (48), are different for ICF and MTIC as the data association uncertainty is incorporated in $\mathbf{G}_i$ for MTIC. This shows the tight integration of the data association and tracking steps in MTIC, as the uncertainty of one step is considered in the other. The same argument holds for EICF Eq. (52) and EMTIC Eq. (54).

6 Experiments

In this section, we compare the performance of the proposed MTIC algorithm with other methods in a simulation framework. We also evaluate the performance of the EICF and EMTIC in simulation and with real-life data.

6.1 Simulation Experiments

The simulation experiments are organized so as to be able to compare performance with linear and non-linear models.

TABLE 1
Comparison of the State Estimation Equations of the KCF, ICF, MTIC, EKCF, EICF and EMTIC for One Particular Target and a Single Consensus Iteration Step

Method	Update Equation		Covariance Equation	
Kalman Consensus Filter (KCF)	$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \left(\mathbf{J}_i^- + \mathbf{B}_i\right)^{-1}\left(\mathbf{b}_i - \mathbf{B}_i\hat{\mathbf{x}}_i^-\right) + \frac{\epsilon}{1+\|(\mathbf{J}_i^-)^{-1}\|}(\mathbf{J}_i^-)^{-1}\sum_{i'\in\mathcal{N}_i}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right)$	(43)	$\mathbf{J}_i^+ = \mathbf{J}_i^- + \mathbf{B}_i$	(44)
Information Weighted Consensus (ICF)	$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{U}_i)\right)^{-1}\left(\mathcal{A}(\mathbf{u}_i) - \mathcal{A}(\mathbf{U}_i)\hat{\mathbf{x}}_i^- + \epsilon\sum_{i'\in\mathcal{N}_i}\frac{\mathbf{J}_{i'}^-}{N_C}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right)\right)$	(45)	$\mathbf{J}_i^+ = N_C\left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{U}_i)\right)$	(46)
Multi-Target Information Consensus (MTIC)	$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \left(\mathcal{A}(\frac{\mathbf{J}_i^-}{N_C}) + \mathcal{A}(\mathbf{U}_i)\right)^{-1}\left(\mathcal{A}(\mathbf{u}_i) - \mathcal{A}(\mathbf{U}_i)\hat{\mathbf{x}}_i^- + \mathcal{A}(\mathbf{U}_i\hat{\mathbf{x}}_i^- - (1-\beta_{i0})\mathbf{U}_i\hat{\mathbf{x}}_i^-) + \epsilon\sum_{i'\in\mathcal{N}_i}\frac{\mathbf{J}_{i'}^-}{N_C}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right)\right)$	(47)	$\mathbf{J}_i^+ = N_C\left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{G}_i)\right)$	(48)
Extended Kalman Consensus Filter (EKCF)	$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \left(\mathbf{J}_i^- + \mathbf{B}_i\right)^{-1}\left(\mathbf{b}_i - \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{h}_i(\hat{\mathbf{x}}_i^-) + \gamma\sum_{i'\in\mathcal{N}_i}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right)\right)$	(49)	$\mathbf{J}_i^+ = \mathbf{J}_i^- + \mathbf{B}_i$	(50)
Extended Information Weighted Consensus (EICF)	$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{U}_i)\right)^{-1}\left(\mathcal{A}(\mathbf{u}_i) - \mathcal{A}(\mathbf{U}_i)\hat{\mathbf{x}}_i^- + \mathcal{A}(\mathbf{U}_i\hat{\mathbf{x}}_i^- - \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{h}_i(\hat{\mathbf{x}}_i^-)) + \epsilon\sum_{i'\in\mathcal{N}_i}\frac{\mathbf{J}_{i'}^-}{N_C}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right)\right)$	(51)	$\mathbf{J}_i^+ = N_C\left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{U}_i)\right)$	(52)
Extended Multi-Target Information Consensus (EMTIC)	$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{U}_i)\right)^{-1}\left(\mathcal{A}(\mathbf{u}_i) - \mathcal{A}(\mathbf{U}_i)\hat{\mathbf{x}}_i^- + \mathcal{A}\left(\mathbf{U}_i\hat{\mathbf{x}}_i^- - (1-\beta_i^0)\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{h}_i(\hat{\mathbf{x}}_i^-)\right) + \epsilon\sum_{i'\in\mathcal{N}_i}\frac{\mathbf{J}_{i'}^-}{N_C}\left(\hat{\mathbf{x}}_{i'}^- - \hat{\mathbf{x}}_i^-\right)\right)$	(53)	$\mathbf{J}_i^+ = N_C\left(\mathcal{A}\left(\frac{\mathbf{J}_i^-}{N_C}\right) + \mathcal{A}(\mathbf{G}_i)\right)$	(54)

Linear Model. In Section 6.2, we evaluate in a simulated environment, the relative performance of the following approaches: MTIC JPDA-KCF, ICF with ground truth data association (ICFGT) and a centralized Kalman Filter with ground truth data association (CKFGT). Note that ICFGT and CKFGT require the knowledge of the ground truth data association, whereas MTIC and JPDA-KCF do not. ICFGT converges to CKFGT in several iterations, thus ICFGT will provide a performance bound for the other iterative approaches that have to solve the data association problem.

Non-linear Model. Section 6.2 also presents the results of the algorithms for non-linear models: EICF, Extended Kalman Consensus Filter (EKCF) (distributed) [23] and the Extended Kalman Filter (centralized). In addition, it presents a simulation analysis of the performance of the proposed EMTIC algorithm in comparison with the Extended Joint Probabilistic Data Association Filter (EJPDAF) (centralized), the EICF (with ground truth data association), and the EKF (with ground truth data association). It can be seen that the EMTIC algorithm compares well with other algorithms (EJPDAF, EICF, EKF) that are either centralized or require the knowledge of the data association, whereas the EMTIC solves both the tracking and data association in a completely distributed manner.

6.1.1 Simulation Experimental Setup

Linear model. We simulate a camera network with $N_C = 15$ cameras (there are additional cameras in the varying camera number simulation below) monitoring an area containing $N_T = 3$ targets roaming randomly in a 500×500 area. Each camera has a rectangular FOV of 200×200 that are randomly placed in such a way that together they cover the

entire area. A circulant network topology with a degree of 2 (at each node) was chosen for the network connectivity. Each target was randomly initialized at a different location with random velocity. The target's state vector was a $4D$ vector, with the $2D$ position and $2D$ velocity components. The targets evolved for 40 time steps using the target dynamical model of Eq. (1). The state transition matrix (used both in track generation and estimation) $\mathbf{\Phi}$ and process covariance $\mathbf{Q}$ were chosen as

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The initial prior covariance $\mathbf{P}_i^{j-}(1) = diag(100, 100, 10, 10)$ was used at each node for each target. The initial prior state $\hat{\mathbf{x}}_i^{j-}(1)$ was generated by adding zero-mean Gaussian noise of covariance $\mathbf{P}_i^{-}(1)$ to initial the ground truth state. The observation was generated using Eq. (2) and $\mathbf{H}_i^j$ was set as

$$\mathbf{H}_i^j = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

If the ground truth state is within the FOV of a sensor, a measurement is generated from the ground truth track using the measurement model Eq. (2) with $\mathbf{R}_i = 100\mathbf{I}_2$. The consensus rate parameter ϵ was set to $0.65/\Delta_{max}$ where $\Delta_{max} = 2$. Total number of consensus iterations per measurement step, K, was set to 20. The parameters for computing the association probabilities, β_i^{jn}'s, were set as follows (see [10] for details on the definitions of the parameters). False measurements (clutter) were generated at each node at each measurement step using a Poisson process with $\lambda = \frac{1}{32}$. Here, λ is the average number of false measurements per sensor. Gate probability P_G was set to 0.99. The probability of detecting a target in each camera, P_D was computed by integrating the probability density function of the predicted measurement, (i.e., $\mathcal{N}(\mathbf{H}_i^j\hat{\mathbf{x}}_i^{j-}, \mathbf{S}_i^j)$) over the area visible to the camera.

To measure the performance of different approaches, one of the parameters was varied while keeping the others to their aforementioned values. As a measure of performance, we computed the estimation error, e, defined as the Euclidean distance between the ground truth position and the estimated posterior position. The simulation results were averaged over multiple simulation runs with 100 randomly generated sets of tracks.

Non-linear Model. The simulation setup is similar to the above with a few differences due to the nonlinearity in the observation model. We simulate a camera network with $N_C = 8$ cameras (there are additional cameras in the varying camera number simulation below) monitoring an area with targets randomly roaming in its 500×500 area. Each of the cameras can view approximately 20 percent of the entire area with some overlap with other cameras. Together the FOV's of all cameras cover the entire area. A circulant network topology with degree 2 (at each node) was chosen for the network connectivity. The observations were generated using Eq. (29) for the standard pin hole projection model. The linearized observation matrix $\mathbf{H}_i^j$ was computed for each target at each camera at each time step using the prior state estimate $\hat{\mathbf{x}}_i^j(t)$ in Eq. (30). The probability of detecting a target in each camera, P_D was computed by integrating the probability density function of the predicted measurement, (i.e., $\mathcal{N}(\mathbf{h}_i(\hat{\mathbf{x}}_i^{j-}), \mathbf{S}_i^j)$) over the area visible to the camera.

6.2 Simulation Results

In this section, the mean (μ_e) of the error norm for different methods is shown for different experiments with both linear and non-linear models.

6.2.1 Varying Clutter

Linear Model. The amount of clutter, λ, was varied from $\frac{1}{256}$ to 8. From the results shown in Fig. 2a it can be seen that both MTIC and JPDA-KCF are very robust even to a very high amount of clutter. The amount of clutter was kept at $\lambda = \frac{1}{32}$ for the other experiments.

Non-linear Model. In Fig. 2b, the amount of clutter was varied from $1/32$ to 2. The total number of consensus iterations, K was set to 5. It can be seen from the figure that the EMTIC algorithm performed very close to its centralized counterpart, the EJPDAF. Also note that the EICFGT and the EKFGT algorithms were not affected by clutter as expected, as the ground truth data association is provided for those two algorithms.

6.2.2 Varying Numbers of Sensors and Targets

Linear Model. The total number of sensors N_C and total number of targets N_T were varied and the results are shown in Figs. 2c and 2e. With more sensors, the total number of available measurements increases which should increase estimation performance. However, with an increase in the number of sensors, the total number of false measurements also increases which can adversely affect the performance. Also, more sensors with the same degree of connectivity, would require more consensus iterations to converge. Due to these contradictory issues, the performance remained almost constant with different number of sensors. With the increase in the number of targets, the problem of data association became more challenging which had an adverse effect in the performance of the different algorithms as can be seen in Fig. 2e.

Non-linear Model. Fig. 2d shows the result with varying number of cameras for non-linear model. Total number of consensus iterations was set to 20. In addition to the issues discussed in the linear case, the error due to the linear approximation of the non-linear model is also present in this case and as a whole it can be seen that the error increases with the number of sensors. From the graph, it can also be seen that with more number of sensors, the error in the distributed algorithms increases with respect to the centralized counterparts as the distributed ones would require more consensus iterations to converge. In Fig. 2f, the number of targets were varied from 1 to 6. The results of the EMTIC and EJPDAF algorithms are very similar. With more targets, the chance of association failure rises, thus the decrease in performance. This clearly does not have any affect in the EICFGT and EKFGT algorithms as the ground truth association is provided.

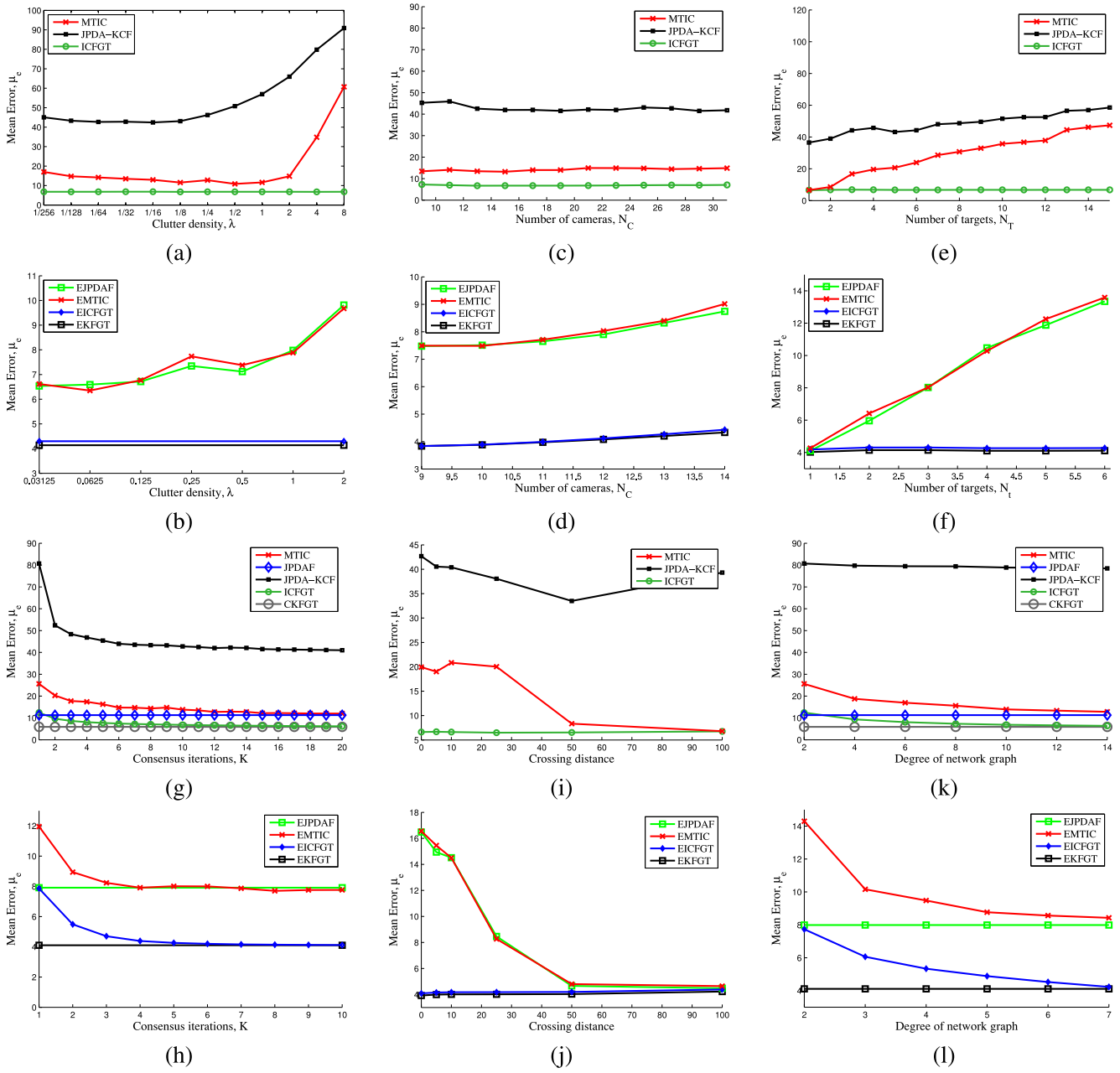

Fig. 2. We show the performance comparison for both MTIC and EMTIC algorithm in a simulation setup. The parameters evaluated are: (a,b) varying amount of clutter, (c,d) varying number of cameras, (e,f) varying number of targets, (g,h) convergence over consensus iterations, (i,j) varying proximity of tracks, and (k,l) varying degree of the network graph. The top and bottom figures are generated from the linear and non-linear model respectively.

6.2.3 *Convergence*

Linear Model. To show the convergence of the different methods, the total number of iterations per measurement step, K was varied. Fig. 2g shows the effect on performance as the number of iterations increases - ICFGT perfprmance approaches that of the CKFGT. It can also be seen that MTIC outperforms JPDA-KCF for any K.

Non-linear Model. The performance of the EMTIC algorithm is shown in Fig. 2h. In the experiments, the results were averaged over 500 randomly generated sets of tracks. At each time step, a set of three tracks was generated. In Fig. 2h, the total number of iterations K, was varied from 1 to 10. As shown theoretically in this paper, as K increased, the performance of the EMTIC algorithm matched its centralized counter-part (i.e., the EJPDAF) algorithm's performance. In the same figure, we also show the performance of the EICF and EKF algorithms where the ground truth data association was provided.

6.2.4 *Varying Target Proximity*

Linear Model. Data association becomes increasingly challenging as target trajectories approach each other. As a measure of proximity, we used a metric called *crossing distance* which is defined as the minimum over all time steps of the distance between all possible pairs of targets for a particular run of the simulation. There were two targets in the scene. In Fig. 2i, when the crossing distance decreases, the performance of different approaches deteriorates. However, MTIC performs better than JPDA-KCF.

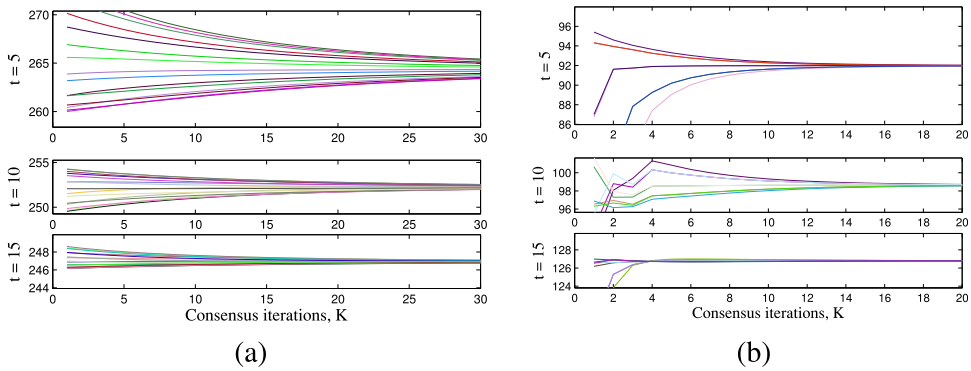

Fig. 3. Difference in convergence speed at different timesteps for (a) linear model and (b) non-linear model.

Non-linear Model. In Fig. 2j, the crossing distance of the targets were varied for the non-linear model. There were two targets in the scene. It can be clearly seen that when the targets come close the performance of the EJPDAF and EMTIC algorithm drops. The performance of these two algorithms was very similar, the underlying reason being that the closer the targets, the higher the chance of wrong data association. This is clearly not an issue for the EICFGT and EKFGT where the ground truth association is given.

6.2.5 Varying Network Topology

Linear Model. Fig. 2k demonstrates the performance with varying degree of the network. The degree was varied from 2 to 14. A degree of d would mean that each node is connected to d other nodes. For each of the randomly generated sequences, a new connected network with a particular degree was randomly generated. Total number of consensus iterations, K, was set to 1. We can observe from the Fig. 2k that with more connection among the nodes, MTIC and ICFGT algorithms converged to JPDAF and KFGT algorithms respectively.

Non-linear Model. In Fig. 2l, the degree of the network graph was varied from 2 to 7. Total number of consensus iterations, K, was set to 1. It can be seen from the figure that with denser connection, the distributed algorithms i.e., EMTIC and EICFGT converged to their centralized counterpart i.e., EJPDAF and EKFGT respectively.

6.2.6 Difference in Convergence Speed

In Fig. 3a, we can see from the timeline diagram that with each iteration, the estimates at different cameras are converging toward a single value. The three plots in Fig. 3a show the nature of convergence at three different time steps for one particular state variable and for one particular target in different sensors. It can be seen that sometimes the initial states at different nodes are very different and it takes more time to converge (i.e., Fig. 3a). On the contrary, sometimes (e.g., center and bottom subplots in Fig. 3a) the initial states are very close and it takes less time to converge. Similar behavior is observed for the non-linear case (Fig. 3b).

6.3 Real-Life Experiments

This section shows the performance of the EMTIC algorithm on two real-life datasets—a multi-camera multi-target dataset captured at EPFL [24] and another that we captured. The algorithms were run on a post-processing basis on the collected datasets, to allow direct comparison of algorithms using exactly the same data inputs. OpenCV library's [25] implementation of histogram of oriented gradients (HOG) based person detector [26] was used to detect the persons in each frame. We show results with only the non-linear models as these were needed for modeling the real-life data.

EPFL Dataset. This dataset [24] provides several sequences with synchronised video streams viewing the same area under different angles. In this experiment, we use laboratory sequences that were shot by four cameras. In this sequence, four to six people sequentially enter the room and walk around for approximately 2.5 minutes. The videos from the EPFL dataset are available at 25 fps. They were used in the algorithms without any subsampling. To calculate the tracking error we use the annotated ground-truth and calibration file provided with the dataset.

Figs. 4a, 4b, 4c, and 4d show the snapshots of the laboratory sequence from EPFL dataset generated at different times. Figs. 5a, 5c, and 5e demonstrate the performance comparison between EMTIC and EJPDAF.

Our Dataset. Six cameras with partially overlapping FOV were chosen for the experiment. The camera network communication topology and FOVs, shown in Fig. 1, was used. The cameras were calibrated with respect to the ground plane. There were 2-4 persons roaming in the area with varying degrees of proximity and occlusion.

Figs. 4e, 4f, 4g, and 4h show snapshots of the real-life experiment for four different scenarios generated from four different sequences. In Figs. 4e, 4f, and 4g, two, three and four targets were present respectively. In Fig. 4h, four targets were present but in closer proximity than in the other three

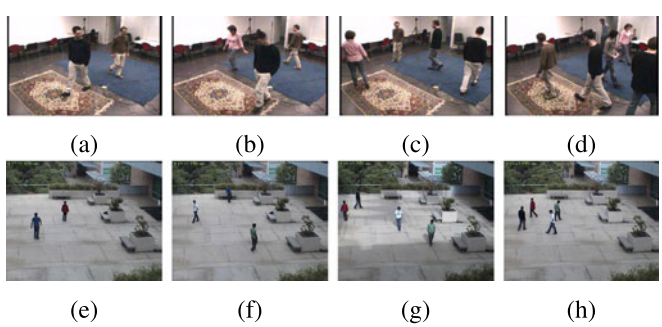

Fig. 4. Example scenarios from EPFL dataset ((a)-(d)) and our dataset ((e)-(h)) on which the proposed EMTIC algorithm were evaluated by varying the number and proximity of targets.

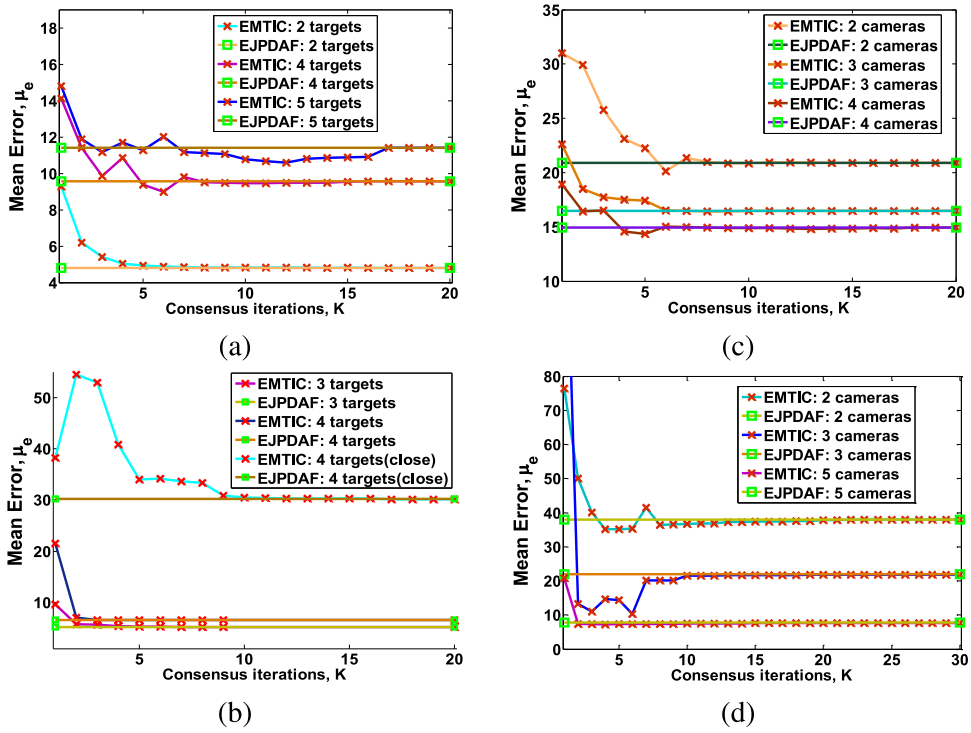

Fig. 5. Performance comparison between EMTIC and EJPDAF on EPFL (top row) and our dataset (bottom row). The subfigures demonstrate the convergence over consensus iterations(K) with (a,b) varying number of targets at fixed number of cameras and (c,d) varying number of cameras at fixed number targets. (Plots are best viewable in color.)

cases. Out of the six cameras, images from only one camera is shown (the one having the best coverage). The data was collected at 30 fps and used without any subsampling in the algorithms. Fig 5b, 5d and 5f show the performance comparison between EMTIC and EJPDAF algorithms on our own dataset on the above four scenarios (Figs. 4e, 4f, 4g, and 4h).

The error metric was defined as the Euclidean distance (in pixels) between the labeled ground truth location on the image plane and the projection of the state estimate onto the camera's image plane. The error was averaged over all the targets and over all time steps for each scenario. Next, we analyze the results in Fig. 5.

Performance with varying number of targets. Figs. 5a and 5b show the convergence over consensus iterations with varying number of targets keeping the number of cameras fixed for EPFL and our dataset respectively. For Fig. 5a, we choose 2, 4 and 5 targets with four cameras to demonstrate the convergence. From the Fig. 5a, we can see that the error increases with the number of targets. Similarly, for our dataset we choose three scenarios from Figs. 4f, 4g, and 4h respectively with six cameras. We can conclude from the Fig. 5b that error increases with the number of targets and the tracking becomes poorer due to data association failure when the targets come close.

Performance with varying number of cameras. In Fig. 5c, the number of cameras varies from 2 to 4 for both EMTIC and EJPDAF with six targets on EPFL dataset. In Fig. 5d, we choose 1, 3 and 5 cameras with four targets (Fig. 4g). As the number of cameras increases, the total number of measurements also increases, which boosts up the estimation performance. From both figures we can see that the results are very similar for both algorithms and error decreases with the increase of the number of cameras.

6.4 Conclusion

In this paper, we proposed the Multi Target Information Consensus algorithm, which is a general consensus-based, distributed, multi-target tracking scheme applicable to a wide-variety of sensor networks. MTIC addresses the issues of naivety and data association in a single efficient algorithm. Addressing naivety (i.e., agents not having measurements for all targets) makes MTIC applicable to sensor networks where the sensors have limited FOV's (which is the case for a camera network). Addressing data association extends the ICF to applications with multiple targets where the data association is not known. Experimental results indicate that MTIC is robust to false measurements/clutter. The article has also extended the ICF and the MTIC algorithms to handle nonlinearity in the observation model, as is the case for most cameras, yielding the EICF and EMTIC algorithms. The properties and structures of the algorithms were compared theoretically and experimentally. Experiments include both simulation and real-life testbed applications.

Acknowledgments

This work was partially supported by US National Science Foundation (NSF) grants IIS-1316934 and CNS-1330110.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[2] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus," in *Proc. IEEE Conf. Decision Control*, 2012, pp. 2732–2737.

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

[4] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed sensor fusion using dynamic consensus," in *Proc. IFAC World Congress*, 2005, pp. 1–6.
[5] B. Song, A. T. Kamal, C. Soto, C. Ding, J. A. Farrell, and A. K. Roy-Chowdhury, "Tracking and activity recognition through consensus in distributed camera networks," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2564–2579, Oct. 2010.
[6] R. Tron and R. Vidal, "Distributed computer vision algorithms," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 32–45, May 2011.
[7] R. Tron and R. Vidal, "Distributed computer vision algorithms through distributed averaging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 57–63.
[8] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. IEEE Conf. Decision Control*, 2009, pp. 7036–7042.
[9] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
[10] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.
[11] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," in *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.
[12] Q. Cai and J. K. Aggarwal, "Tracking human motion in structured environments using a distributed-camera system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1241–1247, Nov. 1999.
[13] B. Song and A. K. Roy-Chowdhury, "Robust tracking in a camera network: A multi-objective optimization framework," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 4, pp. 582–596, 2008.
[14] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed camera networks: Integrated sensing and analysis for wide area scene understanding," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 20–31, May 2011.
[15] C. Ding, A. A. Morye, A. K. Roy-Chowdhury, and J. A. Farrell, "Opportunistic sensing in a distributed PTZ camera network," in *Proc. IEEE/ACM Int. Conf. Distrib. Smart Cameras*, 2012, pp. 1–6.
[16] M. Cetin, L. Chen, J. W. F. Iii, E. T. Ihler, O. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 42–55, Jul. 2006.
[17] L. L. Presti, S. Sclaroff, and M. L. Cascia, "Path modeling and retrieval in distributed video surveillance databases," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 346–360, Apr. 2012.
[18] T. Onel, C. Ersoy, and H. Delic, "On collaboration in a distributed multi-target tracking framework," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 3265–3270.
[19] N. F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *Proc. IEEE Conf. Decision Control*, Dec. 2008, pp. 1085–1090.
[20] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information consensus for distributed multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2403–2410.
[21] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[22] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[23] A. K. Roy-Chowdhury and B. Song, *Camera Networks: The Acquisition and Analysis of Videos over Wide Areas* (series synthesis lectures on computer vision). San Mateo, CA, USA: Morgan, 2012.
[24] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.
[25] G. Bradski, "OpenCV," *Dr. Dobb's Journal of Software Tools*, 2000.
[26] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Int. Conf. Comput. Vis. Pattern Recog.*, Jun. 2005, vol. 2, pp. 886–893.

Ahmed T. Kamal received the BS degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka in 2008 and the MS and PhD degrees in electrical engineering from the University of California, Riverside in 2010 and 2013, respectively. He is currently working as a research and development engineer at Aquifi Inc. at Palo Alto, CA. His main research interests include cyber-physical systems, wide area scene analysis, distributed information fusion, and computer vision.

Jawadul H. Bappy received the BS degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka in 2012. He is currently working toward the PhD degree in electrical and computer engineering at the University of California, Riverside. His main research interests include wide area scene analysis, scene understanding, object recognition, and machine learning.

Jay A. Farrell received the BS degrees in physics and electrical engineering from Iowa State University, and the MS and PhD degrees in electrical engineering from the University of Notre Dame. He is a professor and two time chair of the Department of Electrical Engineering at the University of California, Riverside. He has served the IEEE Control Systems Society (CSS) as a vice president finance and vice president of technical activities, as a general chair of IEEE CDC 2012, and as a president in 2014. He is an author of more than 200 technical publications. He is an author of the book *Aided Navigation: GPS with High Rate Sensors* (McGraw-Hill 2008). He is also co-author of the books *The Global Positioning System and Inertial Navigation* (McGraw-Hill, 1998) and *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches* (John Wiley 2006). He is interested in applied research related to estimation, planning, and control of intelligent autonomous agents. He is a fellow of the IEEE, AAAS, and a distinguished member of IEEE CSS.

Amit K. Roy-Chowdhury received the bachelor's degree in electrical engineering from Jadavpur University, Calcutta, India, the master's degree in systems science and automation from the Indian Institute of Science, Bangalore, India, and the PhD degree in electrical engineering from the University of Maryland, College Park. He is a professor of electrical engineering and a cooperating faculty in the Department of Computer Science, University of California, Riverside. His broad research interests include the areas of image processing and analysis, computer vision, and video communications and statistical methods for signal analysis. His current research projects include intelligent camera networks, wide-area scene analysis, motion analysis in video, activity recognition and search, video-based biometrics (face and gait), biological video analysis, and distributed video compression. He is a coauthor of two books *Camera Networks: The Acquisition and Analysis of Videos over Wide Areas* and *Recognition of Humans and Their Activities Using Video*. He is the editor of the book *Distributed Video Sensor Networks*. He has been on the organizing and program committees of multiple computer vision and image processing conferences and is serving on the editorial boards of multiple journals. He is a senior member of the IEEE.