COLLABORATIVE TRACKING IN DISTRIBUTED MULTI-SENSORS VIDEO-SURVEILLANCE SYSTEMS

Marion Sbai[1] and Samy Meftali[2] and Djamel Aouali[2]

[1]Laboratoire CRISTAL, University of Lille, France
[2]Decarte Engineering, Paris, France

ABSTRACT

Video processing applications are becoming more complex and greedy in terms of computing resources. Thus, the designers of video surveillance systems are moving more and more towards distributed systems, comprising several video sensors collaboratively working to carry out tracking tasks in particular. However, there is a plethora of collaborative tracking algorithms, in the literature, each with its own advantages and disadvantages.

The purpose of this paper is to present the most common collaborative tracking algorithms and discuss the strengths and weaknesses of each.

KEYWORDS

Video surveillance – Tracking – Camera – Multi-sensor – performances

1. INTRODUCTION

Nowadays, video surveillance becomes an absolute imperative for the security of goods and people. Thus, such systems are increasingly found in all kinds of companies and administrations, to more or less automatically perform video surveillance tasks in general and target tracking in particular.

The common feature of most detection and tracking algorithms is their high computational complexity due to amount of data that needs to be processed. This is especially important for live applications, such as video surveillance systems for threats detection or traffic monitoring, thus using one single camera in a complete and efficient video-surveillance system become almost impossible.

Recent trends in surveillance systems and the appearance of high speed wireless network protocols make possible today the use of hundreds of smart cameras in the form of a multiple sensors distributed system that perform video analysis on site on a collaborative way.

The purpose of a tracker is to associate target objects in consecutive video frames to determine their identities and locations. Multi-object tracking is one of the most fundamental tasks of high-level automated video content analysis due to its extensive application: human-machine interaction, security and monitoring, video communication and compression, traffic control and video montage.

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 59–73, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81006

Multi-view trackers combine data from different camera views to estimate the temporal evolution of objects in a monitored area. The data to be combined can be represented by the characteristics of the object (*such as position, color and silhouette*) or by the object's trajectories in each view.

In this context, we start with a single-target tracking review using *EKCF* and then briefly describe several approaches such as *ECKF*, *JPDA-EKCF*, *GKCF*, *Extended Information Consensus Filter (EICF)* and The *EIWCF*. In summary, we describe in detail the problems of multi-target tracking in a camera network scenario [1] with a detailed comparison between different algorithms.

The rest of this paper is organized as follow. Section 2 introduces the distributed processing in camera networks. A survey of Consensus algorithms for distributed tracking is presented in section 3. Section 4 discusses and analysis different tracking algorithms. Section 5 concludes this paper.

2. Distributed Processing in Camera Networks

2.1. Kalman Filter

A partially distributed target tracking approach using a cluster-based Kalman filter was proposed in [2]. Here, a camera is selected as a cluster head which aggregates all the measurements of a target to estimate its position using a Kalman filter and sends that estimate to a central camera.

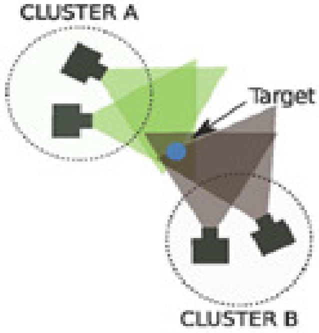


Figure 1: Multiple clusters tracking the same object in a wireless camera network.

2.2. Distributed tracking

In a distributed tracking, each camera node exchanges its estimates with its neighbors until a desired accuracy is achieved. [2]

A Distributed Kalman Consensus filter and subsequent variants have been proposed in [3], [4], [5]. It was a completely distributed solution for estimating the dynamic state of a moving target. However, there are major considerations in applying the method to camera networks due to the nature of video sensors as well as non-linearity, naivety and redundancy. Cameras are directional sensors, each with a limited view of the entire theater of action, with data having high bandwidth and complexity. We will now show how the consensus-based approaches to distributed estimation in multi-agent system literature can be applied to design a consensus-based tracking algorithm in camera networks.

3. Consensus Algorithms For Distributed Tracking

3.1. Mathematical framework

Let C be the set of all cameras in the network. We can then define the subset of all cameras viewing target T_j as $C_j^v \subset C$ and the rest of the cameras as $C_j^{v-¿¿} \subset C$. Each camera C_i will also have its set of *neighboring cameras* $C_i^n \subset C$. Based on the communication constraints due to bandwidth limitations and network connections, we define the set C_i^n as all the cameras with which C_i is able to communicate directly. In other words, C_i can assume that no cameras other than its neighbors, C_i^n, exist as no information flows directly from non-neighboring cameras to C_i. Note that the set of neighbors need not be geographical neighbors. We also define the set of *overlapping cameras* of C_i as $C_i^o \subset C$; since all the cameras can change their *PTZ* parameters and have therefore several possible fields of view, we define the set C_i^o as all the cameras with which C_i can *potentially* have an overlapping field of view. By definition, it becomes clear then that for each $C_i \in C_j^v$, it is true that $C_j^v \subset \{C_i^o \cup C_i\}$. We define $C_i^c \subset C$ as the connected component that C_i is in.We assume $C_i^o \subset C_i^c$, that is to say, C_i is able to exchange information with its overlapping cameras directly or via other cameras. A diagrammatic explanation of the notation is given in ***Fig.2.*** [6] We consider the situation where targets are moving on a ground plane and a homography between each camera's image plane and the ground plane is known. We will show how the state estimation for each target by each camera (*i.e., each camera's estimates based on its individual measurements*) can be combined together through the consensus scheme. This method is independent of the tracking scheme employed in each camera. If the network of cameras is connected, then consensus is achieved across the entire network.

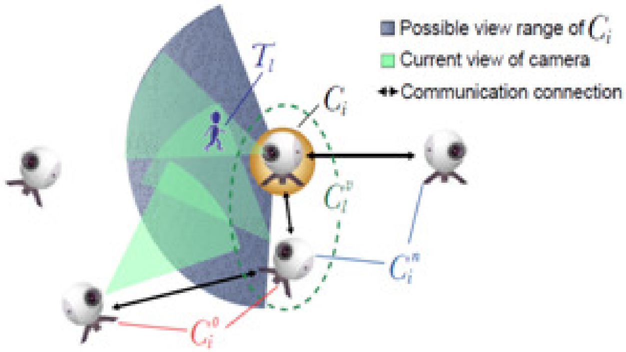


Figure 2: Conceptual illustration of camera network topologies

.

3.2. Algorithm of Extended Kalman-Consensus Filter for a single target

The *Extended Kalman-Consensus Filter algorithm* is developed to solve the problem of *nonlinearity* in the case of *the Kalman-Consensus filter*. This filter is a technique for estimating the state of a noise-disordered nonlinear process using multiple observations from a distributed set of detection nodes. All detection nodes attempt to estimate the same state by determining how their observations affect that state and communicating with neighboring nodes. The algorithm is designed to be more accurate due to measurement diversity, expandable to a large number of nodes, and robust against node loss during operation.

The Extended Kalman-Consensus Filter allows us to track targets on the ground plane using multiple measurements in the image plane taken from various cameras. This allows each camera C_i to have at any time step $\boldsymbol{k}$, a consensus state estimate $\acute{x}_i^j$ and estimate error covariance P_i^j for each target T_j . To model the motion of a target T_j on the ground plane, we consider a linear discrete time dynamical system

$$x^j(k+1) = A^j(k)\,x^j(k) + B^j(k)\,w^j(k),\, x^j(0) \quad \mathcal{N}\left(\mu_i^j, P_0^j\right).$$

And nonlinear observation model for each camera C_i ,

$$z_i^j(k) = h_i\left(x^j(k)\right) + v_i^j(k). \qquad \textbf{\textit{(eq.1)}}$$

Where:

✓ $h_i(.)$: *is the mapping from the ground to the image plane for camera* C_i.

✓ $w^j(k) \wedge v_i^j(k)$: *are zero mean white Gaussian noise.*

✓ $x^j(0)$: *is the initial state of the target.*

Due to the nonlinear nature of the observation model, the linear Kalman-Consensus Filter proposed in [5] cannot be applied as is. An extension to deal with the non-linearity of the observation model is required. Taking into account the nonlinear nature of our dynamical model, we propose an Extended Kalman-Consensus distributed tracking algorithm on the basis of the Kalman-Consensus Filter detailed in [5]. The following are our basic Kalman Filter iterations, as implemented in each camera.

✓ ***Prediction***:

$$P(k+1) = A(k)M(k)A(k)^T + B(k)Q(k)B(k)^T, \qquad \textbf{\textit{(eq.2)}}$$

$$\acute{x}(k+1) = A(k)\hat{x}(k). \qquad \textbf{\textit{(eq.3)}}$$

✓ ***Correction***:

$$\begin{aligned} M(k)^{-1} &= P(k)^{-1} + H(k)^T R(k)^{-1} H(k), \\ K(k) &= M(k)H(k)^T R(k)^{-1}, \qquad \textbf{\textit{(eq.4)}} \\ \hat{x}(k) &= \acute{x}(k) + K(k)\left(z(k) - h\left(\acute{x}(k)\right)\right). \end{aligned}$$

Here, $\boldsymbol{P}$ and $\boldsymbol{M}$ are the *a priori* and *a posteriori* estimate error covariance matrix, respectively, and $\boldsymbol{H}$ is the *Jacobian* matrix of partial derivatives of $\boldsymbol{h}$ with respect to $\boldsymbol{x}$, i.e.

$$H_{[m,n]} = \left.\frac{\partial h_{[m]}}{\partial x_{[n]}}\right|_{x=\acute{x}(k)}, \qquad \textbf{\textit{(eq.5)}}$$

This algorithm is performed at each camera node C_i. At each time step k and for each target T_j, we assume that we are given the estimated prior target state $\acute{x}_i^j(k)$ and the error covariance matrix P_i^j. At time step $k=0$, the Extended Kalman-Consensus filter is initialized with $P_i^j = P_0$ and $\acute{x}_i^j = E\langle x_i^j(0)\rangle$. The consensus algorithm is shown in *Algorithm1*.

The consensus process (*Algorithm1*) is performed at each C_i for each T_j that is in the scene viewed by the camera network. C_i^n is the neighboring camera set of C_i and defined as all cameras with which C_i can directly communicate. If C_i is viewing a target T_j, it obtains T_j's position on its image plane z_i^j, and calculates the Jacobian matrix H_i^j of its observation model and consensus state estimate $\acute{x}_i^j$. After that, the corresponding information vector u_i^j and matrix U_i^j are computed with the given measurement covariance matrix R_i^j and H_i^j. Next, the predicted measurement g_i^j and corresponding residue r_i^j are calculated. C_i then sends a message m_i^j to its neighbors which includes the computed information matrix, residue and its estimated target state $\acute{x}_i^j$ at previous time step ***(k−1).*** Similar to [5], we define the information matrix and vector of $C_i \in C_j^{v-¿¿}$ as $U_i^j = 0$ and $u_i^j = 0$ by assuming that their output matrices are zero, i.e., $H_i^j = 0$ for all $C_i \in C_j^{v-¿¿}$ to avoid any ambiguity arising from the lack of measurements in these cameras. C_i then receives similar messages m_l from the cameras in its neighborhood. The information matrices and residues received from these messages are then fused by C_i with its own information matrix and residue and the Extended Kalman-Consensus state estimate is computed. Finally, the ground plane state $\acute{x}_i^j$ and error covariance matrix P_i^j are updated according to the assumed linear dynamical system.

3.3. Algorithm of JPDA-EKCF for tracking multiple targets

This algorithm is designed to solve the problem of data association by local measurements especially in the case of intra-camera data association i.e., to associate measurements observed by a camera with the targets as well as allows to track joint targets and track maintenance under a probability of detection and an unknown clutter rate. The *Joint Probability Data Association (JPDA)* is coupled to the *Extended Kalman Consensus Filter* (EKCF), which manages nonlinearity.

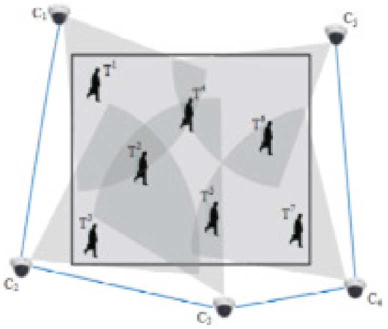


Figure 3: Tracking multiple targets with camera network.

3.3.1. Intra-Camera Association

Due to the weakness of low level video processing methods. Some targets may not be detected because of an occlusion or similar appearance in the background. A direct assignment of the measurement target can lead to poor performance *(problem of naive nodes).* The possibility of false assignment and missed target detection should be considered.

Joint Probability Data Association (*JPDA*) [7] computes an estimate over the various possibilities of measurement-to-track associations. Assume that at time step ***k***, there are $\boldsymbol{N_T}$ targets in the scene and camera C_i obtains $N^i_M(k)$ measurements, $Z_i(k)=\{z_i^1(k),\dots,z_i^{N^i_M}(k)\}$.The history of measurements at camera C_i is denoted as $\mathbf{z}\,(\mathbf{k})\text{¿}\{Z_i(1),\dots,Z_i(k)\}$. Let x^j denote the state of target T_j . Its *a posteriori* state estimate and *a prior* state estimate by camera C_i are denoted as $\hat{x}_i^j$ and $\acute{x}_i^j$, respectively. The state estimate of target T_j at camera C_i is:

$$\hat{x}_i^j=E\left[x^j\middle|\mathbf{z}\,i\right]=\sum_{n=1}^{N^i_M}E\text{¿¿}$$

Where X_i^{jn} denotes the event that measurement z_i^n associates to target T_j at camera C_i. As an extension to standard *Joint Probability Data Association Filter* (JPDAF) **[8]**, the *Extended Kalman Filter* can be used to estimate ¿. Let us denote $\beta_i^{jn}=P\left(\chi_i^{jn}\middle|\mathbf{z}\,i\right)$ and

$$\beta_i^{j0}=1-\sum_{n=1}^{N^i_M}\beta_i^{jn}$$

to represent the probability that target T_j has no measurement associated with it. Then the state estimate can be written as

$$\hat{x}_i^j=\beta_i^{j0}\acute{x}_i^j+\sum_{n=1}^{N^i_M}\beta_i^{jn}\left(\acute{x}_i^j+K_i^j\left(z_i^n-h_i\left(\acute{x}_i^j\right)\right)\right),$$

$$\hat{x}_i^j=\acute{x}_i^j+K_i^j\left(\sum_{n=1}^{N^i_M}\beta_i^{jn}z_i^n-\left(1-\beta_i^{j0}\right)h_i\left(\acute{x}_i^j\right)\right)\quad\textbf{\textit{(eq.6)}}$$

$$\hat{x}_i^j=\acute{x}_i^j+K_i^j\left(\tilde{z}_i^j-\left(1-\beta_i^{j0}\right)h_i\left(\acute{x}_i^j\right)\right),$$

Where

$$\tilde{z}_i^j=\sum_{n=1}^{N^i_M}\beta_i^{jn}z_i^n,$$

and

$$K_i^j=P_i^jH_i^{j^T}\left(H_i^jP_i^jH_i^{j^T}+R_i\right)^{-1}=P_i^jH_i^{j^T}\left(W_i^j\right)^{-1}.$$

H_i^j is the *Jacobian matrix* of partial derivatives of h_i with respect to x_i^j. The error covariance of the estimate is given by

$$M_i^j = P_i^j - \left(1 - \beta_i^{j0}\right) K_i^j W_i^j \left(K_i^j\right)^T + K_i^j \widetilde{P}_i^j \left(K_i^j\right)^T ,$$

Where

$$\widetilde{P}_i^j = \sum_{n=1}^{N_M^i} \beta_i^{jn} \left(z_i^n - h_i\left(\acute{x}_i^j\right)\right)\left(z_i^n - h_i\left(\acute{x}_i^j\right)\right)^T - \left(\widetilde{z}_i^j - h_i\left(\acute{x}_i^j\right)\right)\left(\widetilde{z}_i^j - h_i\left(\acute{x}_i^j\right)\right)^T ,$$

While tracking target in clutter, validation gates are usually used to filter out measurements from clutter within the environment. A validation gate is a metric of "*acceptability*", i.e., within the gate, it is treated as a valid measurement, otherwise it is rejected. Let P_D be the probability that the correct measurement is detected, and P_G be the probability that the correct measurement, if detected, lies within the gate. As shown in [9], by assuming a *Poisson distribution* for false measurements lying in the gate and a *Gaussian distribution* for associating a measurement with a target, using *Bayes rule*, the β_i^j '*s* can be calculated as:

$$\beta_i^{jn} = \frac{\exp\left(\frac{-\left(v_i^{jn}\right)^T \Lambda^{-1} v_i^{jn}}{2}\right)}{b + \sum_{n=1}^{N_M^i} \exp\left(\frac{-\left(v_i^{jn}\right)^T \Lambda^{-1} v_i^{jn}}{2}\right)}$$

$$\beta_i^{j0} = \frac{b}{b + \sum_{n=1}^{N_M^i} \exp\left(\frac{-\left(v_i^{jn}\right)^T \Lambda^{-1} v_i^{jn}}{2}\right)}$$

$$b = \left(2\pi\right)^{\frac{d}{2}} \lambda_f \, det\left(\Lambda\right)^{\frac{1}{2}} \left(1 - P_D P_G\right) / P_D ,$$

where $v_i^{jn} = z_i^n - h_i\left(\acute{x}_i^j\right)$, Λ *is the covariance of the distribution of* ***v****,* ***d*** *is the dimension of measurement vector and* λ_f *is the expected number of occurrences of the Poisson distribution.*

3.3.2. Inter-Camera Association

In distributed tracking of multiple targets, each camera has its own set of estimated tracks and also receives track estimates from its neighbors. Therefore, it is necessary to establish an association between these tracks. This can be formulated as a maximum matching problem in a weighted bipartite graph [10] which minimizes the matching cost. The Hungarian algorithm [11] can be used to find the maximum matching. Different distance metrics can be used to find the matching cost between two track estimates from different cameras.

3.3.3. JPDA-EKCF algorithm

We now show that distributed multiple target tracking can be achieved by integrating data association with a distributed single target tracker. In [12], Joint Probability Data Association (*JPDA*) is coupled with Kalman-Consensus Filter (*KCF*) estimator, where *JPDA* is used to perform local measurement to track associations. This algorithm is referred as *JPDAKCF*. Due to the nonlinear nature of the observation model in the camera network, an extension to deal with the non-linearity is required. Here, we describe an *Extended Kalman-Consensus* Filter coupled with *Joint Probability Data Association* along the lines of the *JPDA-KCF* detailed in [12]. The entire process is shown in *Algorithm2*.

The *JPDA-EKCF algorithm* is performed at each C_i for each T_j that is in the scene under surveillance, where C_i^n is the neighboring camera set of C_i and defined as all cameras with which C_i can directly communicate. Camera C_i computes the assignment of the measurements to targets using *JPDA*. Then C_i calculates the *Jacobian matrix* H_i^j of its observation model with respect to the consensus state estimate $\acute{x}_i^j$ of last time step. After that, the corresponding information vector u_i^j and matrix U_i^j are computed with the given measurement covariance matrix R_i^j and H_i^j. Next, predicted measurements and its corresponding residues are calculated. C_i then sends a message M_i to its neighbors which includes the computed information matrices, residues and its estimated target state $\acute{x}_i^j$ and error covariance P_i^j at previous time step **(*k*−*1*).** C_i then receives similar messages M_l only from the cameras in its neighborhood. Based on the received information, C_i finds the inter-camera track-to-track matching's. The information matrices and residues received from these messages are then fused by C_i with its own information matrices and residues according to the cross camera track matching results and the *Extended Kalman-Consensus* state estimate is computed. Finally, the ground plane state $\acute{x}_i^j$ and error covariance matrix P_i^j are updated according to the assumed linear dynamical system.

3.4. Generalized Kalman Consensus Filter Algorithm

This approach solves the problem of naive nodes. A naive node can associate an observation with a bad target. This can affect the tracking performance of the nodes that actually observe the target by causing them to move away from their estimates. The proposed GKCF is presented in Algorithm 3. Here we first introduce the weighted mean consensus. Then we show how to integrate this consensus pattern into our framework. We then implement the Distributed Kalman Filter (DKF[1]) with the weighted mean consensus results and show how to propagate our covariance and state estimates. For the purpose of easy representation, we use W to denote the information matrix, or inverse covariance matrix, i.e., $W^{-1} = P$. In this section, we will use W^{-1} to replace P as in sections III.2 and III.3.

3.4.1. Weighted Average Consensus

Let the initial state estimate of all N_c agents be $x_i(0)$ with information matrix $W_i(0)$. As we use this information matrix term as weights in the *weighted average consensus algorithm*, the terms weight and *information matrix* will be used interchangeably. Also,

1[1] DKF: (*Distributed Kalman Filter*): Helps to reduce the disagreement of estimates by different nodes.

let $W(0)=\sum_{i=1}^{N_c} W_i(0)$.

So, the global weighted average of the initial states is

$$x^{\text{¿}}=W(0)^{-1}\sum_{i=1:N_c} W_i(0)\,x_i(0). \qquad \textbf{\textit{(eq.7)}}$$

Define the weighted initial state of each agent as

$$\tilde{x}_i(0)=W_i(0)\,x_i(0). \qquad \textbf{\textit{(eq.8)}}$$

Weighted average consensus [3] states that if the iterative update in Equations *(eq.12)* and *(eq.13)* is performed for all $i=1,\dots,N_c$, then each of the terms $W_i(\kappa)^{-1}\tilde{x}_i(\kappa)$ tends to the global weighted average $x^{\text{¿}}$ as $\kappa \to \infty$. As a by-product, the weights also converge to the average of the initial weights. Both these properties of the weighted average consensus will be utilized in our approach.

We assume that the initial information matrix $W_i(0)$, is provided at the initial time step by the target detection mechanism. It would ideally be zero for nodes that are not detecting the target. For nodes that are detecting the target, the initial value would be $W_i(k-1)=H_i^T R^{-1} H_i$.

At the k^{th} iteration, the agents communicate with each other with the $W_i(k-1)$ and $\tilde{x}_i(k-1)$ information. Then, using the previously discussed average consensus scheme, they get an updated prior state estimate $\hat{x}_i^{-\text{¿}|k|\text{¿}}$ and weight estimate $W_i(k)$ (*see eqns. (eq.12), (eq.13) and (eq.14)*). This prior estimate tends towards the global normalized weighted average as stated before.

This approach solves the problem of *naive nodes*. A naive node can associate an observation with a bad target. This can affect the tracking performance of the nodes that actually observe the target by causing them to move away from their estimates. The proposed GKCF is presented in *Algorithm 3*. Here we first introduce the weighted mean consensus. Then we show how to integrate this consensus pattern into our framework. We then implement the Distributed Kalman Filter with the weighted mean consensus results and show how to propagate our covariance and state estimates. For the purpose of easy representation, we use W to denote the information matrix, or inverse covariance matrix, i.e., $W^{-1}=P$ In this section, we will use W^{-1} to replace P as in sections *III.2* and *III.3.*

3.4.2. Weighted Average Consensus

Let the initial state estimate of all N_c agents be $x_i(0)$with information matrix $W_i(0)$. As we use this information matrix term as weights in the *weighted average consensus* *algorithm*, the terms weight and *information matrix* will be used interchangeably. Also, let $W(0)=\sum_{i=1}^{N_c} W_i(0)$. So, the global weighted average of the initial states is

$$x^{¿}=W(0)^{-1}\sum_{i=1:N_c} W_i(0)\,x_i(0). \qquad \textit{(eq.7)}$$

Define the weighted initial state of each agent as

$$\tilde{x}_i(0)=W_i(0)\,x_i(0). \qquad \textit{(eq.8)}$$

Weighted average consensus [3] states that if the iterative update in Equations *(eq.12)* and *(eq.13)* is performed for all $i=1,\ldots,N_c$, then each of the terms $W_i(\kappa)^{-1}\tilde{x}_i(\kappa)$ tends to the global weighted average $x^{¿}$ as $\kappa \rightarrow \infty$. As a by-product, the weights also converge to the average of the initial weights. Both these properties of the weighted average consensus will be utilized in our approach.

We assume that the initial information matrix $W_i(0)$, is provided at the initial time step by the target detection mechanism. It would ideally be zero for nodes that are not detecting the target. For nodes that are detecting the target, the initial value would be $W_i(k-1)=H_i^T R^{-1} H_i$

At the k^{th} iteration, the agents communicate with each other with the $W_i(k-1)$ and $\tilde{x}_i(k-1)$ information. Then, using the previously discussed average consensus scheme, they get an updated prior state estimate $\hat{x}_i^{-¿|k|¿}$ and weight estimate $W_i(k)$ (*see eqns. (eq.12), (eq.13) and (eq.14)*). This prior estimate tends towards the global normalized weighted average as stated before.

3.5. Extended Information Consensus Filter (EICF)

This approach allows the effect of *naivety* and *nonlinearity* to be managed without requiring knowledge of other nodes in the network.

We propose two distributed filters for tracking targets in wireless camera networks, ECF1 and EIF2, which compute the local information, $y^i_{k\vee k},$ and $Y^i_{k\vee k},$ differently. *EICF1* runs at each node c_i and computes the local information values, $y^i_{k\vee k}$ and $Y^i_{k\vee k},$ based on their own respective measurement information, i^i_k and I^i_k:

$$\begin{aligned} Y^i_{k\vee k} &= Y^i_{k\vee k-1}+I^i_k \\ y^i_{k\vee k} &= y^i_{k\vee k-1}+i^i_k \end{aligned} \qquad \textit{(eq.17)}$$

and then exchange the values $y^i_{k\vee k}$ and $Y^i_{k\vee k}$ with neighbours to achieve average consensus.

EICF2 computes local information values, $y^i_{k\vee k}$ and $Y^i_{k\vee k},$ based on its own measurement information and also that of neighbouring nodes:

$$Y^{i}_{k\vee k}=Y^{i}_{k\vee k-1}+\sum_{j\in\left|c_i\cup C^{n}_{i}\right|} I^{j}_{k} \qquad \textbf{(eq.18)}$$

$$y^{i}_{k\vee k}=y^{i}_{k\vee k-1}+\sum_{j\in\left|c_i\cup C^{n}_{i}\right|} i^{j}_{k}$$

EICF2 reaches convergence faster than *EICF1*, at the cost of additional communication to send the measurement information terms. Hence, *EICF2* is recommended when sufficient communication resources are available.

The iterative information exchange between neighbours results in redundancy which causes correlation among the nodes' estimates. Hence, the *EICF* results are sub-optimal because of such correlation among the individual node estimates. In the update step (*see Eq. 10*) of a filter, the two terms involved are the priors, $y^{i}_{k\vee k-1}$ and $Y^{i}_{k\vee k-1}$, and the measurement information about the target, i^{i}_{k} and I^{i}_{k}. The prior information is the result of the prediction on previous estimates, $y^{i}_{k\vee k}$ and $Y^{i}_{k\vee k}$,, which are computed after consensus. Hence, the redundancy always lies in the prior information terms, $y^{i}_{k\vee k-1}$ and $Y^{i}_{k\vee k-1}$.

3.6. Extended Information Weighted Consensus Filter (EIWCF)

This algorithm uses the EIWCF to handle the three main problems (*naivety, redundancy and non-linearity*). However requires knowledge of the number of cameras N_c, the basic principle of these algorithms is to weight the estimates of the nodes according to their covariance information. When N_c is not available, EICF can be used at the cost of not managing the redundancy problem.

Via proper weighting of prior and measurement information, IWCF mitigates the problem of redundancy [13]. By applying the concept of *IWCF* to *EIF*, we propose a non-linear distributed filter called the Extended Information Weighted Consensus Filter *(EIWCF).* Here the prior information is weighted by $1/N_c$ and the consensus proposals are prepared as:

$$v^{i}_{k\vee k}=¿\,\frac{1}{N_c}\,y^{i}_{k\vee k-1}+i^{i}_{k}$$

$$V^{i}_{k\vee k}=¿\,\frac{1}{N_c}\,Y^{i}_{k\vee k-1}+I^{i}_{k} \qquad \textbf{(eq.19)}$$

After achieving consensus on the v and V terms, the results are multiplied by N_c :

$$y^{i}_{k\vee k}=N_c\,v^{i}_{k\vee k}$$

$$Y^{i}_{k\vee k}=N_c\,V^{i}_{k\vee k} \qquad \textbf{(eq.20)}$$

These estimates are not affected by non-linearity, naivety and redundancy. However, *EIWCF* requires the knowledge of the number of nodes in the network (*see Eqs. 12 and 13*). Thus, *EIWCF* cannot be applied when such knowledge is not available whereas *EICF1* or *EICF2* can be used. If sufficient communication resources to receive neighbours' measurement information, $\boldsymbol{i}_k^j$ and $\boldsymbol{I}_k^j$, is available, *EICF2* achieves faster convergence than *EICF1*. Hence the choice depends on the available communication resources.

4. ANALYSIS AND COMPARISON

Based on our proposed approaches, there have been three major issues in consensual distributed tracking for camera networks that are as follows:

✓ Non-linearity.

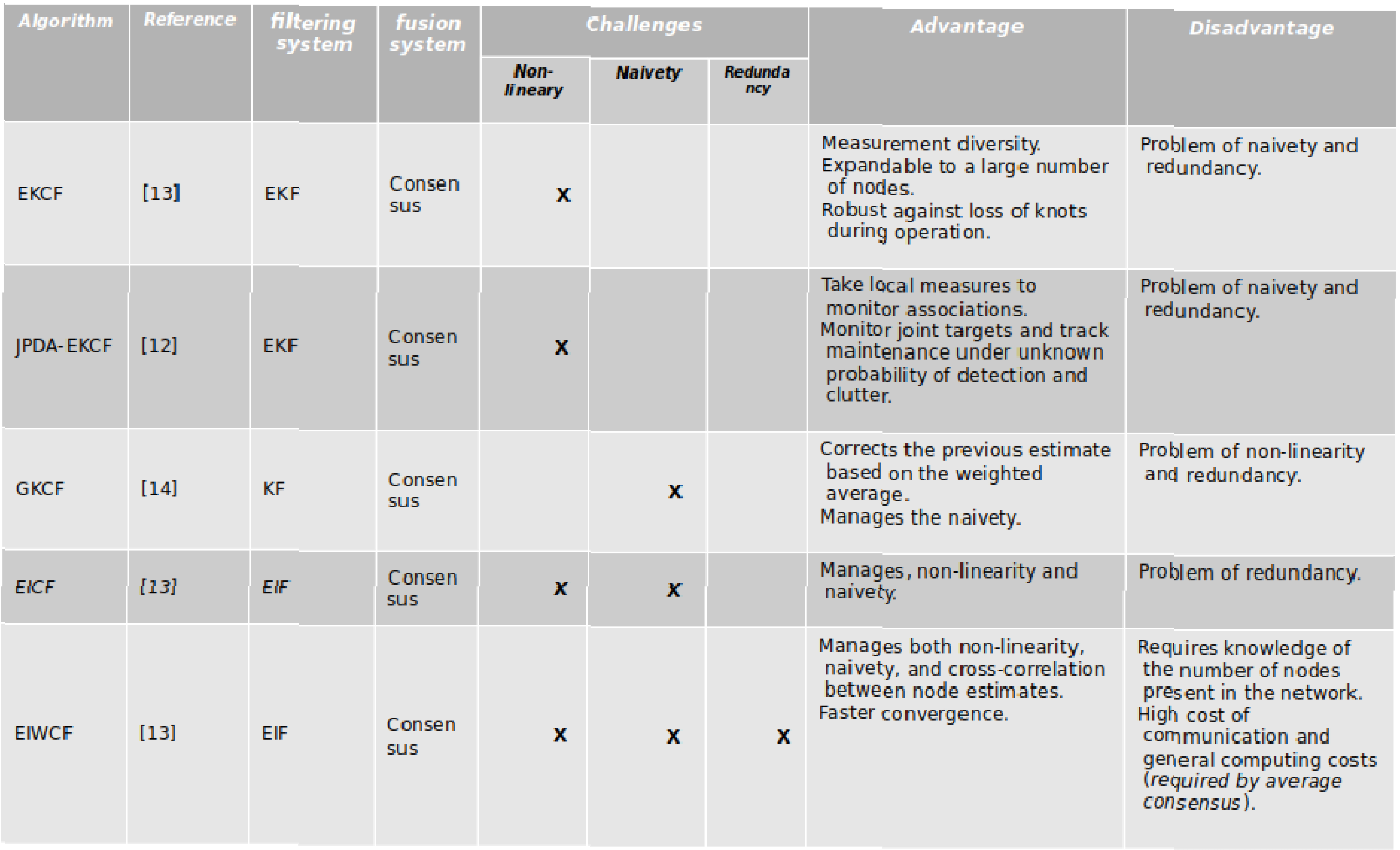
✓ Naivety.

✓ Redundancy.

The *Kalman Extended Consensus Filter* (EKCF) algorithm is extensible to a large number of nodes, robust against the loss of nodes during operation, and has measurement diversity. The JPDA-EKCF makes local measurements to track associations, track joint targets, and track maintenance under unknown probability of detection and clutter. However, these filters do not deal with naivety and redundancy in a camera network, but the GKCF handles naivety and corrects the previous estimate according to the weighted average but it does not deal with nonlinearity and redundancy.

Afterwards, we proposed an *Extended Information Consensus Filter* (EICF). This filter performs a weighted averaging while addressing the problem of *naive nodes* and *nonlinearity*. To overcome the *redundancy problem*, we have also proposed a weighted consensus extended information filter (EIWCF). The EIWCF handles *naivety*, *redundancy* and *non-linearity*, and achieves faster convergence by correctly weighting past and measurement information. However, it requires knowledge of the number of nodes in the network.

The table below (*see Fig 4*), summarizes a detailed comparison between the different algorithms based on references.



Algorithm	Reference	filtering system	fusion system	Challenges			Advantage	Disadvantage
				Non-lineary	Naivety	Redundancy		
EKCF	[13]	EKF	Consensus	x			Measurement diversity. Expandable to a large number of nodes. Robust against loss of knots during operation.	Problem of naivety and redundancy.
JPDA-EKCF	[12]	EKF	Consensus	x			Take local measures to monitor associations. Monitor joint targets and track maintenance under unknown probability of detection and clutter.	Problem of naivety and redundancy.
GKCF	[14]	KF	Consensus		x		Corrects the previous estimate based on the weighted average. Manages the naivety.	Problem of non-linearity and redundancy.
EICF	*[13]*	*EIF*	Consensus	x	x		Manages, non-linearity and naivety.	Problem of redundancy.
EIWCF	[13]	EIF	Consensus	x	x	x	Manages both non-linearity, naivety, and cross-correlation between node estimates. Faster convergence.	Requires knowledge of the number of nodes present in the network. High cost of communication and general computing costs (*required by average consensus*).

Figure 4 :Algorithms for distributed tracking in camera network.

5. CONCLUSION

Video tracking can be defined as a problem of locating a moving object (*or multiple objects*) over time based on the observations of the object in the images. In other words, the purpose of a tracker is to associate target objects in consecutive video frames to determine their identities and locations. Multi-object tracking is one of the most fundamental tasks of high-level automated video content analysis through its extensive applications. Maintaining the stability of tracks on multiple video targets over extended periods and extended areas remains a difficult problem. Among the most basic monitoring methods are the *Kalman filter* and the *JPPAF filters*, we presented a distributed state estimation method based on the *Generalized Kalman Consensus Filter* (GKCF), which has exceeded the KCF approach under such conditions. However, in themselves, these methods are generally not able to follow extended spatial horizons. Since the measurement model of a camera is non-linear, algorithms based on the *Kalman filter* can not be used. Nonlinear filters such as the *Kalman Extended Consensus Filter* (EKCF) do not deal with naivety and redundancy. To overcome the redundancy problem, we have also proposed the *Extended Information Weighted Consensus* (EIWCF) filter by combining the *Extended Information Filter* (EIF) and the *Information-weighted Consensus Filter* (IWCF). The EIWCF handles naïveté, redundancy and non-linearity, and achieves faster convergence by correctly weighting past and measurement information. However, it requires knowledge of the number of nodes in the network.

As future work, we will explore the reduction of communication and computational overhead required by average consensus. The management of dynamic link structure and asynchronous networks are other possible future works.

REFERENCES

[1] Samuel Davey, Neil Gordon, Ian Holland, Mark Rutten, Jason Williams. “Bayesian Methods in the Search for MH370”, Commonwealth of Australia 2016.

[2] W. Li, W. Zhang “Multiple target localization in wireless visual sensor networks” Front. Comput. Sci., pp. 496-504. 7 (4) 2013.

[3] Donato Di Paolaa, Antonio Petittia, Alessandro Rizzob. “Distributed Kalman Filtering via Node Selection in Heterogeneous Sensor Networks”. ISSIA-Bari, Italy; September 2013.

[4] Kar, J. M. F. Moura, and K. Ramanan, “Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication”. IEEE Transactions on Information Theory, vol. 58, no. 6, pp. 3575–3605, 2012.

[5] S. Das and J. M. F. Moura, “Distributed Kalman filtering and network tracking capacity” in 47th Asilomar Conference on Signals, Systems, and Computers, pp. 629–633. 2013.

[6] Amit K. Roy-Chowdhury, Bi Song.“Camera Networks: The Acquisition and Analysis of Videos over Wide Areas”, University of California, Riverside, 133 pages. January 2012.

[7] Xiao Chen, Yaan Li, Yuxing Li, Jing Yu and Xiaohua Li “A Novel Probabilistic Data Association for Target Tracking in a Cluttered Environment”. Northwestern Polytechnical University, China, December 2016.

[8] M. Chandrajit, R. Girisha, T. Vasudev and M. Hemesh. “Data Association and Prediction for Tracking Multiple Objects”, Indian Journal of Science and Technology, Vol 9 (33). September 2016.

[9] Taek Lyul Song ; Hyoung Won Kim ; Darko Musicki. “Iterative joint integrated probabilistic data association for multi-target tracking”, IEEE Transactions on Aerospace and Electronic Systems Vol.51, April 2015.

[10] Abul K. M. Azad, Mohammed Misbahuddin. “Web-Based Object Tracking Using Collaborated Camera Network” . Northern Illinois University, DeKalb, IL, USA. Apr 27, 2018.

[11] Humayra Dil Afroz, Dr.Mohammad Anwar Hossen. ”New Proposed Method for Solving Assignment Problem and Comparative Study with the Existing Methods”. Journal of Mathematics (IOSR-JM). Volume 13, Issue 2 Ver. IV, pp 84-88. Mar. - Apr. 2017.

[12] Subhro Dasy, José.M, F. Mouraz, “Distributed estimation of dynamic fields over multi-agent networks”, NY 10598, USA – Jan. 2017.

[13] Ziren Wang, Guoliang Liu, Guohui Tian, “Human skeleton tracking using information weighted consensus filter in distributed camera networks”, Chinese Automation Congress (CAC), Oct. 2017.

[14] Nemanja Ilié, Khaled Obaid Al Ali, Milos S. Stankovic and Srdjan S. Stankovic, “Distributed Multitarget Tracking in Camera Networks Using Multi-step Consensus”, Proceedings of 4th International Conference on Electrical, Electronics and Computing Engineering, Kladovo, Serbia, June 05-08. 2017.