The costs of fusion in smart camera networks

Sandeep Katragadda, Juan C. SanMiguel, Andrea Cavallaro
Centre for Intelligent Sensing, Queen Mary University of London
{s.katragadda, juan.carlos.sanmiguel, a.cavallaro}@qmul.ac.uk

ABSTRACT

The choice of the most suitable fusion scheme for smart camera networks depends on the application as well as on the available computational and communication resources. In this paper we discuss and compare the resource requirements of five fusion schemes, namely centralised fusion, flooding, consensus, token passing and dynamic clustering. The Extended Information Filter is applied to each fusion scheme to perform target tracking. Token passing and dynamic clustering involve negotiation among viewing nodes (cameras observing the same target) to decide which node should perform the fusion process whereas flooding and consensus do not include this negotiation. Negotiation helps limiting the number of participating cameras and reduces the required resources for the fusion process itself but requires additional communication. Consensus has the highest communication and computation costs but it is the only scheme that can be applied when not all viewing nodes are connected directly and routing tables are not available.

Keywords

Information fusion, communication cost, computation cost, target tracking, smart camera networks

1. INTRODUCTION

Fusion schemes are widely used in sensor networks to improve task performance and robustness to failures [13]. These schemes define when and what information to share under specific communication architectures [4]. Resource-limited sensor networks require selecting the most suitable scheme to trade-off performance and resources used. Thus, quantifying the costs of resources helps in choosing the most appropriate scheme for each scenario.

Fusion schemes can share raw data (e.g. measurements) or decisions (e.g. estimations) [30]. In the former case, measurements or features are fused to obtain the global estimate. In the latter case, local estimates at each node are fused to get the global estimate. Fusion can be centralised, decentralised or distributed [28]. In centralised fusion, all nodes send their local information to a fusion centre (FC) via single-hop or multi-hop communications [8]. The decentralised scheme [10] considers various FCs that collect and fuse information from nodes in their neighbourhood. The allocation of nodes to FCs can be static [10] or dynamic [18]. To support topology changes and scalability, dynamic decentralisation (or clustering) is preferred. In distributed fusion [27], each node runs an identical peer-to-peer algorithm to exchange information with other nodes. Flooding [7], consensus [21] and token passing [11] are widely used distributed fusion schemes.

In this paper, we analyse the communication and computation costs of five fusion schemes, namely centralised fusion [8], flooding [7], token passing [11], average consensus [22] and dynamic clustering [18]. We employ the Extended Information Filter (EIF) [19] for all the schemes to perform target tracking in smart camera networks using decision-based fusion. Based on this analysis we discuss which scheme to use based on the communication topology and the available communication and computation resources. The software of the fusion schemes is available at `http://www.eecs.qmul.ac.uk/~andrea/software.htm`.

The paper is organised as follows. Section 2 reviews the EIF-based state estimation. Section 3 discusses the fusion schemes for target tracking and Section 4 compares their costs. Finally, Section 5 concludes the paper.

2. EXTENDED INFORMATION FILTER FOR STATE ESTIMATION

Consider a smart camera network with N_c camera nodes. Each node c_i ($1 \leq i \leq N_c$) consists of an image sensor, a processor and a wireless communication module. Let the target motion model be given by $\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1})$, where $\mathbf{x}_k$ is the estimated target state at time k and $\mathbf{w}_{k-1}$ is a non-additive zero-mean Gaussian noise with covariance $\mathbf{Q}_{k-1}$. $\mathbf{P}_k$ represents the error covariance of the state estimation. The measurement model of a node c_i is $\mathbf{z}^i_k = h^i(\mathbf{x}_k) + \mathbf{v}^i_k$, where $\mathbf{z}^i_k$ is the target measurement and $\mathbf{v}^i_k$ is an additive

Sandeep Katragadda is supported by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA 2010-0012. Juan C. SanMiguel is supported by the EU Crowded Environments monitoring for Activity Understanding and Recognition (CENTAUR, FP7-PEOPLE-2012-IAPP) project under GA number 324359. Andrea Cavallaro acknowledges the support of the Artemis JU and in part by the UK Technology Strategy Board through the COPCAMS Project under Grant 332913.

ICDSC '14, November 4-7, 2014, Venezia Mestre, Italy

http://dx.doi.org/10.1145/2659021.2659032.

zero-mean Gaussian noise with covariance $\mathbf{R}$. Given a posterior at time $k-1$ ($\mathbf{x}^i_{k-1|k-1}$, $\mathbf{P}^i_{k-1|k-1}$) and measurement at time k ($\mathbf{z}^i_k$), filtering at node c_i estimates the posterior at time k ($\mathbf{x}^i_{k|k}$, $\mathbf{P}^i_{k|k}$).

If either $f(\cdot)$ or $h^i(\cdot)$ are mildly non-linear and first-order approximations of their Jacobians are available via Taylor series, Extended Kalman Filter (EKF) can be used for state estimation [19]. The Extended Information Filter (EIF) is an alternative form of EKF that represents the state and covariance as the Fisher information vector $\mathbf{y}_{k|k} = \mathbf{P}^{-1}_{k|k}\mathbf{x}_{k|k}$ and Fisher information matrix $\mathbf{Y}_{k|k} = \mathbf{P}^{-1}_{k|k}$, respectively. The prediction step of EIF is [19]:

$$\begin{aligned}
&\mathbf{x}^i_{k|k-1} = f(\mathbf{x}^i_{k-1|k-1}),\\
&\mathbf{P}^i_{k|k-1} = \mathbf{J}_{f,k-1}\mathbf{P}^i_{k-1|k-1}\mathbf{J}^T_{f,k-1} + \mathbf{W}_{f,k-1}\mathbf{Q}_{k-1}\mathbf{W}^T_{f,k-1},
\end{aligned} \tag{1}$$

where $\mathbf{J}_{f,k-1} = \frac{\partial f}{\partial \mathbf{x}}|_{\mathbf{x}^i_{k-1|k-1},0}$ and $\mathbf{W}_{f,k-1} = \frac{\partial f}{\partial \mathbf{w}}|_{\mathbf{x}^i_{k-1|k-1},0}$ are the Jacobians of $f(\cdot)$ w.r.t. $\mathbf{x}$ and $\mathbf{w}$, respectively. The results are converted to the information form:

$$\mathbf{y}^i_{k|k-1} = {\mathbf{P}^i_{k|k-1}}^{-1}\mathbf{x}^i_{k|k-1},\ \mathbf{Y}^i_{k|k-1} = {\mathbf{P}^i_{k|k-1}}^{-1}. \tag{2}$$

The update (correction) step in EIF [19] is:

$$\begin{aligned}
&\mathbf{y}^i_{k|k} = \mathbf{y}^i_{k|k-1} + \mathbf{J}^T_{h,k}\mathbf{R}^{-1}[\mathbf{z}^i_k - h^i(\mathbf{x}_{k|k-1}) + \mathbf{J}_{h,k}\mathbf{x}^i_{k|k-1}],\\
&\mathbf{Y}^i_{k|k} = \mathbf{Y}^i_{k|k-1} + \mathbf{J}^T_{h,k}\mathbf{R}^{-1}\mathbf{J}_{h,k},
\end{aligned} \tag{3}$$

where $\mathbf{J}_{h,k} = \frac{\partial h}{\partial x}|_{x^i_{k|k-1},0}$ is the Jacobian of $h^i(\cdot)$ w.r.t. $\mathbf{x}$ (approximately linearised measurement model $h^i(\cdot)$). EIF has a simpler update step than EKF.

The posteriors of each c_i are fused using one of the five fusion schemes (centralised fusion, flooding, token passing, consensus and dynamic clustering) to compute:

$$\mathbf{y}^g_{k|k} = \sum_{i\in C^v_k}\mathbf{y}^i_{k|k},\ \mathbf{Y}^g_{k|k} = \sum_{i\in C^v_k}\mathbf{Y}^i_{k|k}, \tag{4}$$

where $[\mathbf{y}^g_{k|k}\ \mathbf{Y}^g_{k|k}]$ is the global posterior and C^v_k is the set of all viewing nodes (cameras observing the same target) at time k and $N^v_k = |C^v_k|$. The global state estimate and corresponding error covariance can be calculated using:

$$\mathbf{x}^g_{k|k} = {\mathbf{Y}^g_{k|k}}^{-1}\mathbf{y}^g_{k|k},\ \mathbf{P}^g_{k|k} = {\mathbf{Y}^g_{k|k}}^{-1}. \tag{5}$$

3. FUSION SCHEMES

We perform decision fusion to combine the posteriors of EIF-based nodes using centralised fusion, flooding, token passing, average consensus and dynamic clustering. This section describes each scheme for target tracking in smart camera networks.

In *centralised* fusion [8, 25], all viewing nodes send their local posteriors ($\mathbf{y}^i_{k|k}$ and $\mathbf{Y}^i_{k|k}$) to a FC for computing the global posterior ($\mathbf{y}^F_{k|k}$ and $\mathbf{Y}^F_{k|k}$). Centralised fusion is suitable for small-scale networks as it has high communication cost near the FC. Other drawbacks of centralised fusion are the vulnerability to FC failures and limited robustness to topology changes.

In *flooding* (or dissemination) [29, 13] all viewing nodes broadcast their local posterior ($\mathbf{y}^i_{k|k}$ and $\mathbf{Y}^i_{k|k}$) to all or to subsets of nodes (e.g. viewing nodes) in the network. Information can be distributed in a single iteration if the network is fully connected [17]. Otherwise, flooding requires multi-hop or multiple iterations of communications. In each iteration, each node sends its own and the previously received information to its neighbours. Eventually all participating nodes have the same set of posteriors [7, 6]. Then, each participating node performs fusion, updates its local posterior. Note that when we aim to flood information only to viewing nodes, non-viewing nodes might hold less accurate results as they do not receive the posterior of all viewing nodes. In such cases, non-viewing nodes do not perform fusion to save computation. For large and sparse networks, flooding has high communication cost, high processing cost and high memory requirements [23]. This scheme is therefore suitable for sharing low amounts of information when high connectivity exists among the nodes.

Token passing [24, 11, 12] is a sequential estimator in which viewing nodes form an aggregation chain (AC). Each node in the AC receives a partial posterior from the previous one, updates this posterior using its local posterior and sends the result to the next node. The process finishes when all AC nodes are visited once. The most informative node (decided based on the local posterior and the global knowledge of the network) is selected as the *next* node [13]. The *last* AC node provides the global posterior at the current time step. Then, this node initiates the AC for the next time step (often also becoming the *first* AC node). The sequential estimation and the transmission of high dimensional estimations such as Particle Filter (PF) posteriors cause latency [13]. Nastasi and Cavallaro [20] applied such a fusion scheme to smart camera networks using distributed PFs assuming that viewing nodes can communicate with each other. The scheme is suitable when cameras with overlapping FOVs are connected or routing tables are provided.

Reaching *consensus* means that all nodes have the same value for the considered variable(s) such as the target state [29, 22]. Consensus schemes operate at two time scales: collecting measurements and performing iterations between consecutive measurement collections [23]. In each iteration, nodes exchange information with neighbours and perform fusion using the average [2], gossip [1, 3], maximum or minimum [9] approaches. Average consensus is widely used in wireless sensor networks [22] and smart camera networks [27, 26, 28]. The distributed Kalman Consensus Filter (KCF) [22] computes local estimates ($\mathbf{x}^i_{k|k}$) via Kalman Filters (KF). Non-linear measurement models or motion models require other filters such as EIF [16] or PF [23]. In *average consensus*, each node c_i exchanges its posterior ($\mathbf{y}^i_{k|k}$ and $\mathbf{Y}^i_{k|k}$) with neighbours where non-viewing nodes send either zeros or predicted posterior [14] as information. Each node c_i executes a consensus step as:

$$\mathbf{y}^{i,l}_{k|k} = \mathbf{y}^{i,l-1}_{k|k} + \mathbf{w}_{ij}\sum_{j\in C^N_i}(\mathbf{y}^{j,l-1}_{k|k} - \mathbf{y}^{i,l-1}_{k|k}), \tag{6}$$

where $\mathbf{y}^{i,l}_{k|k}$ is the consensus achieved after the l^{th} iteration and C^N_i is the neighbourhood of c_i. The same process is applied to $\mathbf{Y}^i_{k|k}$. The values $\mathbf{w}_{ij}$ can be set to guarantee the convergence to the average of the initial estimates of all nodes after L iterations [21]. The speed of convergence to the posterior average depends on the number of nodes. By multiplying the average with the total number of nodes in the network, N_c, the sum (global posterior) can be calculated as in Equation 4 [25]. The advantages of this scheme are the

Algorithm 1 Decision based fusion in a camera network

Input:
N_c : number of nodes in the network
C/C_k^v : set of all/viewing cameras at time k
C_i^N : set of communication neighbours of node c_i
c_F, c_H : fusion centre, cluster head
c_f, c_p, c_n : first, previous, next node in aggregation chain (AC)
$[\mathbf{y}_{k|k}^i\ \mathbf{Y}_{k|k}^i]$: posterior information of c_i at time k
$\mathbf{m}_{k|k}^i$: message containing $[\mathbf{y}_{k|k}^i\ \mathbf{Y}_{k|k}^i]$
$\mathtt{send}(\mathtt{c_i}, \mathtt{c_j}, \mathbf{m})$: node c_i sends message $\mathbf{m}$ to node c_j
z_k^i: measurement of node c_i at time k
$[y_{k|k}^g\ Y_{k|k}^g]$: global posterior
$\mathtt{EIF}(\mathtt{y^i_{k-1|k-1}}, \mathtt{Y^i_{k-1|k-1}}, \mathbf{z}^i_k)$: computes posterior using Extended Information Filter
$\mathtt{clustering}(\mathtt{C^v_k})$: forms a cluster of C_k^v and returns its head
$\mathtt{findNextNode}(\mathtt{c_i})$: identifies the next node of node c_i in AC
$\mathtt{avgConsensus}(\mathbf{m}^i_{k|k})$: performs average consensus on $\mathbf{m}_{k|k}^i$ (Equation 6)
L: Number of consensus iterations
For each $c_i \in C$
  If $c_i \in C_k^v$, $[y_{k|k}^i\ \ Y_{k|k}^i] = \mathtt{EIF}(\mathtt{y^i_{k-1|k-1}}, \mathtt{Y^i_{k-1|k-1}}, \mathbf{z}^i_k)$
  Else $[y_{k|k}^i\ \ Y_{k|k}^i] = [\mathbf{0}\ \mathbf{0}]$
  endIf
endFor
Switch(Algorithm **)**
  Case Centralized **:**
    For each $c_i \in C_k^v$, $\mathtt{send}(\mathtt{c_i}, \mathtt{c_F}, \mathbf{m}^i_{k|k})$ endFor
    c_F performs fusion:

$$[\mathbf{y}_{k|k}^g\ \mathbf{Y}_{k|k}^g] = [\mathbf{y}_{k|k}^F\ \mathbf{Y}_{k|k}^F] = \left[\sum_{j\in C_k^v} \mathbf{y}_{k|k}^j \quad \sum_{j\in C_k^v} \mathbf{Y}_{k|k}^j\right]$$

    $\mathtt{send}(\mathtt{c_F}, \mathtt{C}, \mathbf{m}^F_{k|k})$
    break;
  Case Flooding **:**
    For each $c_i \in C_k^v$, $\mathtt{send}(\mathtt{c_i}, \mathtt{C^N_i}, \mathbf{m}^i_{k|k})$ endFor
    For each c_i performs fusion:

$$[\mathbf{y}_{k|k}^g\ \mathbf{Y}_{k|k}^g] = [\mathbf{y}_{k|k}^i\ \mathbf{Y}_{k|k}^i] = \left[\sum_{j\in C_k^v} \mathbf{y}_{k|k}^j \quad \sum_{j\in C_k^v} \mathbf{Y}_{k|k}^j\right]$$

    endFor
    For each $c_i \in C_k^v$, $\mathtt{send}(\mathtt{c_i}, \mathtt{C^N_i}, \mathbf{m}^i_{k|k})$ endFor
    break;
  Case Token Passing **:**
    $c_p = c_f$
    While($c_n = \mathtt{findNextNode}(\mathtt{c_p})$ exists)
      $send(c_p, c_n, \mathbf{m}_{k|k}^p)$
      c_n performs local update:

$$[\mathbf{y}_{k|k}^n\ \mathbf{Y}_{k|k}^n] = [\mathbf{y}_{k|k}^n + \mathbf{y}_{k|k}^p\ \ \mathbf{Y}_{k|k}^n + \mathbf{Y}_{k|k}^p]$$

      $c_p = c_n$
    endWhile
    $[\mathbf{y}_{k|k}^g\ \mathbf{Y}_{k|k}^g] = [\mathbf{y}_{k|k}^p\ \mathbf{Y}_{k|k}^p]$
    $\mathtt{send}(\mathtt{c_p}, \mathtt{C^N_p}, \mathbf{m}^i_{k|k})$
    $c_f = c_p$
    break;
  Case Consensus **:**
    For $l = 1 : L$
      For each $c_i \in C$, $\mathtt{send}(\mathtt{c_i}, \mathtt{C^N_i}, \mathbf{m}^i_{k|k})$ endFor
      For each $c_i \in C$, $\mathtt{avgConsensus}(\mathbf{m}^i_{k|k})$ endFor
    endFor
    For each $c_i \in C$, c_i computes posterior:

$$[\mathbf{y}_{k|k}^g\ \mathbf{Y}_{k|k}^g] = [\mathbf{y}_{k|k}^i\ \mathbf{Y}_{k|k}^i] = [N_c\mathbf{y}_{k|k}^i\ \ N_c\mathbf{Y}_{k|k}^i]$$

    endFor
    break;
  Case Dynamic Clustering **:**
    $c_H = \mathtt{clustering}(\mathtt{C^v_k})$
    For each $c_i \in C_k^v$, $\mathtt{send}(\mathtt{c_i}, \mathtt{c_H}, \mathbf{m}^i_{k|k})$ endFor
    c_H performs fusion:

$$[\mathbf{y}_{k|k}^g\ \mathbf{Y}_{k|k}^g] = [\mathbf{y}_{k|k}^H\ \mathbf{Y}_{k|k}^H] = \left[\sum_{j\in C_k^v} \mathbf{y}_{k|k}^j \quad \sum_{j\in C_k^v} \mathbf{Y}_{k|k}^j\right]$$

    $\mathtt{send}(\mathtt{c_H}, \mathtt{C^N_H}, \mathbf{m}^H_{k|k})$
    break;
return $\mathbf{y}_{k|k}^g$ and $\mathbf{Y}_{k|k}^g$

Table 1: Summary of selected characteristics of fusion schemes. Key. CE: Centralised. FL: Flooding. CO: Consensus. TP: Token Passing. DC: Dynamic Clustering. CH: Cluster Head. DIS: Distributed. DEC: Decentralised.

Scheme	Type	Fusion centres		Involves negotiation?	Network knowledge needed?
		Nodes	Dynamic selection		
CE	CE	Single	No	No	Yes
FL	DIS	Viewing	Yes	No	No
TP	DIS	Viewing	Yes	During fusion	Yes
CO	DIS	All	No	No	No
DC	DEC	CH	Yes	Prior to fusion	Yes

availability of global posterior at all nodes and robustness to node failures. Moreover, this scheme does not require routing protocols or knowledge about nodes (e.g. observation models or FOVs) and the network (e.g. communication graph), thus coping with topology changes and link failures.

In *dynamic clustering*, nodes viewing the target negotiate locally and form clusters where a node is selected as cluster head. The node generates the global posterior by fusing its own ($\mathbf{y}_{k|k}^H$ and $\mathbf{Y}_{k|k}^H$) and received posterior from the other cluster members ($\mathbf{y}_{k|k}^i$ and $\mathbf{Y}_{k|k}^i$). Static clusters can be created to track targets based on their overlapping sensing regions [10] and might use more nodes per cluster than needed. In order to cluster only the viewing nodes, dynamic clustering adapts the cluster membership depending on the target location and the network topology [18]. Iterative message exchange (or negotiation) is required to select the cluster head and to propagate the cluster membership over time. In this case cluster formation is a distributed process and fusion is a decentralised process. Cluster formation and cluster-head selection add computation and communication costs and increase latency.

Multiple fusion centres exist in all schemes except centralised fusion. All nodes operate similarly in consensus, whereas only viewing nodes operate in flooding, token passing and dynamic clustering. At the end of each time step, the FCs broadcast the result (global posterior) to their neighbourhood in order to use the global posterior as input for their local EIFs if the target enters their field of view (FOV) in the next time step (Equation 1). Moreover, token passing and dynamic clustering require negotiation among nodes (prior to fusion) to decide whom to pass the token to and to propose cluster-head candidates, respectively. If there is no direct communication among the viewing nodes, dynamic clustering forms multiple single-hop clusters, flooding requires several iterations, and token passing needs routing tables for multi-hop communication.

The implementation of the five target tracking approaches is presented in Algorithm 1. The function `clustering()` is used in dynamic clustering to decide which cluster a camera belongs to [18]. The function returns the id of the cluster head. The function `findNextNode()` is used in the token passing approach to decide the next node in the AC [20]. The function returns the id of the next node.

Finally, Table 1 and 2 summarise the fusion schemes used in sensor networks and the existing approaches in smart camera networks, respectively; whereas Table 3 compares the communication and computation costs for each fusion scheme.

Table 2: Decentralised (DEC) and Distributed (DIS) tracking techniques for smart camera networks. Key. SC: Static Clustering. DC: Dynamic Clustering. TP: Token Passing. CO: Consensus. KF: Kalman Filter. EKF: Extended Kalman Filter. PF: Particle Filter. IF: Information Filter.

Reference	Fusion type		Fusion scheme				Filter
	Data	Decision	DEC		DIS		
			SC	DC	TP	CO	
[10]	✓		✓				KF
[18]	✓			✓			EKF
[31]	✓			✓			KF
[20]		✓			✓		PF
[27, 26]		✓				✓	KF
[5]		✓				✓	EKF
[14, 15]		✓				✓	IF
[16]		✓				✓	EIF

4. QUANTIFYING THE COSTS OF FUSION

We consider a wireless smart camera network of eight cameras (Figure 1(a)) that have overlapping FOVs and are single-hop neighbours. We use the communication graph shown in Figure 1(b) where error-free communications and no false measurements are assumed. We consider the 50 trajectories shown in Figure 1(c). Each target is tracked using the five approaches presented in Section 3. The target follows the motion model given by [18]:

$$\mathbf{x}_k = \begin{bmatrix} x_{k-1} + v_{x,k-1}\delta_k + a_x\delta_k^2/2 \\ y_{k-1} + v_{y,k-1}\delta_k + a_y\delta_k^2/2 \\ v_{x,k-1} + a_x\delta_k \\ v_{y,k-1} + a_y\delta_k \\ \delta_k + \epsilon \end{bmatrix}, \qquad (7)$$

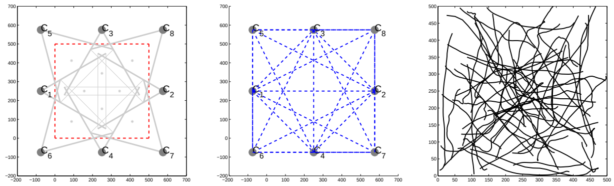
where $\mathbf{x}_k = [x_k \; y_k \; v_{x,k} \; v_{y,k} \; \delta_k]^T$ is the state vector at time k, (x_k, y_k) is the target position on the ground plane; $(v_{x,k}, v_{y,k})$ is the target velocity, (a_x, a_y) is the target acceleration, δ_k is the time step between consecutive measurements and ϵ is the time uncertainty that models synchronisation errors among cameras. We model (a_x, a_y, ϵ) as Gaussian with zero mean and covariance $\mathbf{Q} = diag([1 \;\; 1 \;\; 0.01])$. The measurement model of node c_i is:

$$\mathbf{z}_k^i = \begin{bmatrix} u_k^i \\ v_k^i \end{bmatrix} = \begin{bmatrix} \dfrac{H_{11}^i x_k + H_{12}^i y_k + H_{13}^i}{H_{31}^i x_k + H_{32}^i y_k + H_{33}^i} \\ \dfrac{H_{21}^i x_k + H_{22}^i y_k + H_{23}^i}{H_{31}^i x_k + H_{32}^i y_k + H_{33}^i} \end{bmatrix} + \mathbf{v}_k, \qquad (8)$$

where $\mathbf{z}_k^i = [u_k^i \; v_k^i]^T$ is the measurement vector with the target coordinates in the image plane of the node c_i, H^i is its homography matrix that converts the ground-plane coordinates to the image plane of c_i and $\mathbf{v}_k$ is the measurement noise independently modelled for each camera as a zero-mean Gaussian with covariance $\mathbf{R} = diag([5 \;\; 5])$. The H^i values are taken as

$$H^i = \begin{bmatrix} 397.2508 & 95.2020 & 287280 \\ 51.7437 & 396.9189 & 139100 \\ 0.0927 & 0.1118 & 605.2481 \end{bmatrix}.$$

We use the result (global posterior estimation) available at FC (centralised), cluster head (dynamic clustering) and the last node (token passing) for accuracy comparison. For flooding, the result is available at all viewing nodes. Similarly, for consensus the result is available at all network nodes and therefore we use their average as the final state estimate for accuracy comparison. Note that the consensus estimate converges to the centralised estimate when enough consensus iterations are used [27]. The accuracy error in all approaches except consensus is 6.48 ± 2.47 meters as they perform the same operations (Equation 4). The difference is *where* and in *what order* the sum is computed. In the case of consensus, the error in the estimate is 7.93 ± 2.76 meters. We use two consensus iterations ($L = 2$, see Equation 6) as an accuracy-cost compromise. The costs of the five fusion schemes are summarised in Table 4 and discussed in detail in the next sections.



(a) (b) (c)

Figure 1: Experimental setup. (a) Camera locations and their field of views (FOV). Dotted red box: 500×500m area where targets move and '*': the FOV centre; (b) communication graph; (c) trajectories used for evaluation.

4.1 Communication cost

During the fusion process cameras transmit their local posteriors $\mathbf{y}_{k|k}^i$ and $\mathbf{Y}_{k|k}^i$. As the information matrix $\mathbf{Y}_{k|k}^i$ is symmetric, only the upper triangular values are sent to reduce communication cost. The size of the local posterior ($[\mathbf{y}_{k|k}^i \; \mathbf{Y}_{k|k}^i]$) being fused is 20 scalars.

In centralised fusion, all viewing nodes send the local posteriors to the FC and then the FC sends the global posterior to all nodes. In flooding, each viewing node sends its local posterior to all other viewing nodes. After fusion each viewing node broadcasts the fusion result (global posterior) to its neighbourhood. Dynamic clustering sends information flags to distinguish information for clustering and information for fusion. Dynamic clustering employs similar communications for fusion to token passing but requires a different additional cost for negotiation among neighbours to select the cluster head. The associated overhead is highly reduced when the set of viewing nodes does not change (i.e. the cluster membership does not need to be updated). Negotiations are necessary when adding or removing a cluster member. Once cluster formation is done, communication involves only the transfer of local posteriors from members to the head. The estimated global posterior is sent back to all neighbours.

Token passing has a higher communication cost than dynamic clustering because each viewing node queries its neighbours at each time step to identify the next node in AC. Token passing involves transmitting queries to neighbours, their replies and partial posteriors to the identified next node. Hence, all viewing cameras, except the last one in the AC, perform a minimum transmission of $size([\mathbf{y}_{k|k}^i \; \mathbf{Y}_{k|k}^i])$ scalars (the posterior information). The maximum number of queries a node in AC transmits depends on its number of

Table 3: The costs of fusion schemes at time k. For simplicity the subscript k is discarded in this table. Key. N_c: Total number of nodes. C^v: Set of viewing nodes. N^v: Number of viewing nodes ($|C^v|$). m: message containing the posterior [y Y]. $\mathcal{C}_{cm}(\cdot)$: Communication cost. $\mathcal{C}_{cp}(\cdot)$: Computation cost. L: Number of consensus iterations.

Fusion scheme	Availability of global posterior	Communication cost		Computation cost	
		Fusion	Negotiation	Fusion	Negotiation
Centralised Fusion	Fusion centre	$(N^v+1)\times size(\mathbf{m})$	–	$\mathcal{C}_{cp}\left(\sum_{j\in C^v}\mathbf{y}^j\right)+\mathcal{C}_{cp}\left(\sum_{j\in C^v}\mathbf{Y}^j\right)$	–
Flooding	All viewing nodes	$2\times N^v\times size(\mathbf{m})$	–	$N^v\times\mathcal{C}_{cp}\left(\sum_{j\in C^v}\mathbf{y}^j\right)+N^v\times\mathcal{C}_{cp}\left(\sum_{j\in C^v}\mathbf{Y}^j\right)$	–
Token Passing	Last node	$N^v\times size(\mathbf{m})$	$\mathcal{C}_{cm}(\texttt{findNextNode}())$	$N^v\times\mathcal{C}_{cp}(\mathbf{y}+\mathbf{y}))+N^v\times\mathcal{C}_{cp}(\mathbf{Y}+\mathbf{Y}))$	$\mathcal{C}_{cp}(\texttt{findNextNode}())$
Consensus	All nodes	$L\times N_c\times size(\mathbf{m})$	–	$L\times N_c\times\mathcal{C}_{cp}(\texttt{consensus}(\mathbf{m}))$	–
Dynamic Clustering	Cluster head	$N^v\times(1+size(\mathbf{m}))$	$\mathcal{C}_{cm}(\texttt{clustering}())$	$\mathcal{C}_{cp}\left(\sum_{j\in C^v}\mathbf{y}^j\right)+\mathcal{C}_{cp}\left(\sum_{j\in C^v}\mathbf{Y}^j\right)$	$\mathcal{C}_{cp}(\texttt{clustering}())$

Table 4: Average costs of fusion schemes when applied to a camera network in which cameras with overlapping FOVs are connected (see Figure 1). L: Number of consensus iterations.

Fusion scheme	Communication cost (no. of scalars transmitted)			Computation cost (no. of scalar operations)		
	Fusion	Negotiation	Total	Fusion	Negotiation	Total
Centralised Fusion	64.52 ± 8.36	–	64.52 ± 8.36	24.52 ± 8.36	–	24.52 ± 8.36
Flooding	89.04 ± 16.71	–	89.04 ± 16.71	299.32 ± 63.09	–	299.32 ± 63.09
Token Passing	45.07 ± 8.35	16.57 ± 1.61	61.65 ± 9.93	44.52 ± 8.36	119.83 ± 25.25	164.35 ± 33.58
Consensus ($L=2$)	320 ± 0	–	320 ± 0	4960 ± 0	–	4960 ± 0
Dynamic Clustering	46.75 ± 8.77	3.57 ± 1.17	50.31 ± 9.39	24.52 ± 8.36	2.46 ± 1.12	26.98 ± 8.98

neighbours. Token passing requires additional communication between the last node of AC in the previous time step and the first node of AC of the current time step when the last node of the previous AC does not have measurements in the current time step. If N_k^v is the number of viewing nodes at time step k, only N_k^v-1 nodes transmit their local posteriors to the next node in the AC in token passing and to the cluster head in dynamic clustering. In consensus, all nodes (viewing and non-viewing) participate in the fusion thus increasing the communication cost. Consensus has the highest cost whereas dynamic clustering the lowest.

The fusion schemes can therefore be sorted by decreasing communication cost as consensus, flooding, centralised fusion, token passing and dynamic clustering.

4.2 Computational cost

In our experiments the average filtering cost of 50 trajectories is 3145.3 ± 590 scalar operations. We do not consider this filtering cost in the comparison as it is the same for all the schemes.

The computation cost of fusion depends on how many nodes are involved and how much information is fused at the nodes (Table 4). The centralised fusion scheme has the lowest cost as only one node, the FC, performs fusion. Consensus is the most expensive as it performs iterative fusion. Even if only a few cameras are viewing the target, all cameras perform the same operations thus resulting in a high number of computations in the network. In flooding, only the viewing nodes perform fusion. Therefore flooding has a smaller computation cost than consensus. In token passing only viewing nodes perform fusion. However, token passing requires a significant number of additional computations to identify the next node (e.g. based on target-to-neighbours distance). Each viewing node only fuses the partial posterior received from its previous node in the AC and its local posterior. Cameras participating in the AC do sequential fusion where viewing cameras update the posterior. Only the first (or last) camera of the AC does not perform fusion. Non-viewing cameras do not perform any tasks. Token passing has a higher computation cost than dynamic clustering because each camera has to identify the next node and perform fusion. In contrast, for dynamic clustering, once the cluster is formed and as long as the target does not leave FOV of at least one of the cluster members, the cluster does not change. For this reason its negotiation cost is significantly smaller compared to that of token passing. Moreover, in dynamic clustering and centralised fusion, only one node performs fusion; whereas in token passing, flooding and consensus multiple nodes do the fusion. Centralised fusion and dynamic clustering differ in the extra computation involved in identifying the cluster head. The computation cost of flooding is higher than that of token passing as each viewing node fuses all received data from other viewing cameras.

The schemes can therefore be sorted by decreasing computation cost as consensus, flooding, token passing, dynamic clustering and centralised fusion.

5. CONCLUSIONS

We discussed five fusion schemes and compared their resource requirements (communication and computation) in a wireless camera network assuming that cameras with overlapping field of views can communicate. We combined the Extended Information Filter with centralised fusion, flooding, token passing, consensus and dynamic clustering for target tracking with non-linear motion and measurement models.

While consensus-based fusion involves high communication and computation costs compared to the other four fusion schemes, consensus provides state estimation at each node in the network and it is more robust to node failures. Token passing and dynamic clustering require negotiation among the nodes. Token passing has significant additional communication and computation costs as it involves sequential fusion when each node updates its local posterior.

Our future work will aim at reducing the communication and computation costs by addressing the stopping criteria of iterative (consensus) and sequential (token passing) analysis.

6. REFERENCES

[1] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. on Signal Processing*, 57(7):2748–2761, 2009.

[2] G. Battistelli, L. Chisci. Kullback-leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50(3):707–718, 2014.

[3] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah. Randomized gossip algorithms. *IEEE Trans. on Inf. Theory*, 52(6):2508–2530, 2006.

[4] M. Cetin, L. Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright, A. S. Willsky. Distributed fusion in sensor networks. *IEEE Signal Processing Magazine*, 23(4):42–55, 2006.

[5] C. Ding, B. Song, A. Morye, J. A. Farrell, A. K. Roy-Chowdhury. Collaborative sensing in a distributed PTZ camera network. *IEEE Trans. on Image Processing*, 21(7):3282–3295, 2012.

[6] P. M. Djuric, J. Beaudeau, M. F. Bugallo. Non-centralized target tracking with mobile agents. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 5928–5931, 2011.

[7] P. M. Djuric, L. Geng. Non-centralized target tracking in networks of directional sensors. In *Proc. of IEEE Aerospace Conf.*, pages 1–6, 2011.

[8] H. Durrant-Whyte. *Multi Sensor Data Fusion*. Technical report, Australian Centre for Field Robotics, The University of Sydney, Australia, 2001.

[9] S. Farahmand, S. I. Roumeliotis, G. B. Giannakis. Set-membership constrained particle filter: Distributed adaptation for sensor networks. *IEEE Trans. on Signal Processing*, 59(9):4122–4138, 2011.

[10] R. Goshorn, J. Goshorn, D. Goshorn, H. Aghajan. Architecture for cluster-based automated surveillance network for detecting and tracking multiple persons. In *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*, pages 219–226, 2007.

[11] O. Hlinka, P. M. Djuric, F. Hlawatsch. Time-space-sequential distributed particle filtering with low-rate communications. In *Proc. of Asilomar Conf. on Signals, Systems and Computers*, pages 196–200, 2009.

[12] O. Hlinka, F. Hlawatsch. Time-space-sequential algorithms for distributed Bayesian state estimation in serial sensor networks. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, pages 2057–2060, 2009.

[13] O. Hlinka, F. Hlawatsch, P. M. Djuric. Distributed particle filtering in agent networks: A survey, classification, and comparison. *IEEE Signal Processing Magazine*, 30(1):61–81, 2013.

[14] A. T. Kamal, J. A. Farrell, A. Roy-Chowdhury. Information weighted consensus filters and their application in distributed camera networks. *IEEE Trans. on Automatic Control*, 58(12):3112–3125, 2013.

[15] A. T. Kamal, J. A. Farrell, A. K. Roy-Chowdhury. Information consensus for distributed multi-target tracking. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2403–2410, 2013.

[16] S. Katragadda, J. C. SanMiguel, A. Cavallaro. Consensus protocols for distributed tracking in wireless camera networks. In *Proc. of IEEE Conf. on Information Fusion*, 2014.

[17] W. Li, Y. Jia. Distributed consensus filtering for discrete-time nonlinear systems with non-gaussian noise. *Signal Processing*, 92(10):2464–2470, 2012.

[18] H. Medeiros, J. Park, A. C. Kak. Distributed object tracking using a cluster-based Kalman filter in wireless camera networks. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):448–463, 2008.

[19] A. G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1998.

[20] C. Nastasi, A. Cavallaro. Distributed target tracking under realistic network conditions. In *Sensor Signal Processing for Defence*, pages 1–5, 2011.

[21] R. Olfati-Saber, J. A. Fax, R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. of the IEEE*, 95(1):215–233, 2007.

[22] R. Olfati-Saber, N. F. Sandell. Distributed tracking in sensor networks with limited sensing range. *Proc. of American Control Conf.*, pages 3157–3162, 2008.

[23] V. Savic, H. Wymeersch, S. Zazo. Belief consensus algorithms for fast distributed target tracking in wireless sensor networks. *Signal Processing*, 95:149–160, 2014.

[24] X. Sheng, Y.-H. Hu, P. Ramanathan. Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In *Proc. of Int. Symp. on Information Processing in Sensor Networks*, pages 181–188, 2005.

[25] A. Simonetto, T. Keviczky, R. Babuska. Distributed nonlinear estimation for robot localization using weighted consensus. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 3026–3031, 2010.

[26] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, A. K. Roy-Chowdhury. Distributed camera networks. *IEEE Signal Processing Magazine*, 28(3):20–31, 2011.

[27] B. Song, A. T. Kamal, C. Soto, C. Ding, J. A. Farrell, A. K. Roy-Chowdhury. Tracking and activity recognition through consensus in distributed camera networks. *IEEE Trans. on Image Processing*, 19(10):2564–2579, 2010.

[28] M. Taj, A. Cavallaro. Distributed and decentralized multicamera tracking. *IEEE Signal Processing Magazine*, 28(3):46–58, 2011.

[29] L. Xiao. A scheme for robust distributed sensor fusion based on average consensus. In *Proc. of the Int. Conf. on Information Proc. in Sensor Networks*, pages 63–70, 2005.

[30] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, C.-W. Yi. Data fusion improves the coverage of wireless sensor networks. In *Proc. of the Int. Conf. on Mobile Computing and Networking*, pages 157–168, 2009.

[31] J. Yoder, H. Medeiros, J. Park, A. C. Kak. Cluster-based distributed face tracking in camera networks. *IEEE Trans. on Image Processing*, 19(10):2551–2563, 2010.