Neighbour consensus for distributed visual tracking

Sandeep Katragadda, Andrea Cavallaro
Centre for Intelligent Sensing, Queen Mary University of London
Email: {s.katragadda, a.cavallaro}@qmul.ac.uk

***Abstract*—We propose N-consensus, an algorithm that reduces the cost of the consensus process for distributed visual target tracking without compromising on tracking accuracy. N-consensus fuses target posteriors computed by *viewing nodes* (i.e. the cameras viewing the same target) only and limits the number of nodes participating in consensus to those within a specified number of hops from the viewing nodes. The number of hops is computed based on viewing and communication ranges to identify all nodes within twice the viewing range from the viewing nodes. Unlike average consensus, the proposed N-consensus does not require prior knowledge of node connectivity because we employ an improved fast covariance intersection algorithm during consensus update.**

I. Introduction

Distributed information fusion is desirable for target tracking in wireless camera networks (WCN) to achieve scalability and robustness to node or link failures [1]. Information to be fused includes the estimated target states and the corresponding error covariances. Due to the limited communication ranges and directional field of views (FOV), the wireless cameras (nodes) viewing simultaneously the same target (*viewing nodes*) might be multi-hop neighbours when appropriate lenses are used. In such cases, distributed fusion of the target's information from multiple viewing nodes can be done through consensus [2].

Consensus algorithms are distributed protocols that aim at reaching an agreement on a decision variable using iterative peer-to-peer communication among the nodes [3]. An advantage of consensus algorithms is that, unlike other distributed approaches such as token-passing, they require neither full connectivity among the viewing nodes nor routing tables. Each consensus iteration involves two main steps, namely information exchange with neighbouring nodes and consensus update (i.e. weighted fusion of local and received information). Average consensus (A-consensus) is a widely used consensus algorithm for target tracking in wireless sensor networks [4], [5] and WCNs [6]–[8]. A-consensus aims at having decisions at all nodes (e.g. target state) to converge to their average. The weights used in the consensus update of A-consensus depend on the prior knowledge of connectivity (communication topology) [9]. Iterative Covariance Intersection (ICI) [10] is a consensus algorithm that does not need prior knowledge of connectivity. ICI computes the weights using either the trace [11] or the determinant [12] of the covariance information and achieves better accuracy than A-consensus. Both A-consensus and ICI achieve consensus among all network nodes, and therefore the total energy consumption for communication and computation increases with the number of nodes (for a given number of viewing nodes) [2].

In this paper, we propose Neighbour consensus, N-consensus, a distributed algorithm that identifies dynamically a reduced set of nodes in the neighbourhood of the viewing nodes and achieves consensus only among these nodes. The set includes viewing nodes and their neighbourhood up to a certain number of hops so that all nodes that are viewing the target at the current time step and the nodes that might view the target at the next time step are included. N-consensus reduces the number of nodes involved in the consensus process. The software of the proposed method is available at *http://www.eecs.qmul.ac.uk/~andrea/software.htm*.

The remainder of the paper is organised as follows. Section II presents the proposed N-consensus algorithm. Section III compares the performance of the proposed algorithm with A-consensus and ICI. Finally, Section IV concludes the paper.

II. Neighbour consensus

A. Preliminaries

Consider a WCN, $C = \{c_1, c_2, ...\}$, consisting of homogeneous nodes monitoring an area to track a target. Each node c_i has a directional FOV with viewing range r_v and communication range r_c. Let N_i be the communication neighbourhood of c_i. At time step k, only a subset of cameras, C_k^V, can view the target because of the directionality and limited viewing range. The nodes do not have information about the network or its neighbours such as communication topology, routing tables, and vision graph (graph representing node pairs having overlapping FOVs). The network is so large that the distance between farthest nodes $>> 2r_v$. We assume ideal communication, no false detections, calibrated cameras, that the target is visible in at least one camera at any time step k and that it does not move more than $2r_v$ between consecutive time steps.

Each viewing node estimates the target state representing position and velocity on the ground plane. The information is a Gaussian probability density function (posterior) with mean $\mathbf{x}_{i,k}$ and covariance $\mathbf{P}_{i,k}$. Each viewing node c_i at time step k computes the local posterior of the target based on the posterior at $k-1$ (prior) and the measurement at k. The posterior is represented as information vector $\mathbf{y}_{i,k} = \mathbf{P}_{i,k}^{-1}\mathbf{x}_{i,k}$ and information matrix $\mathbf{Y}_{i,k} = \mathbf{P}_{i,k}^{-1}$. The goal is to fuse the local posteriors, $[\mathbf{y}_{i,k} \;\; \mathbf{Y}_{i,k}]$, of all viewing nodes $c_i \in C_k^V$ in a distributed way with minimum energy consumption when the viewing nodes are not communicative neighbours. The distributed algorithm aims at reaching the centralised result:

$$[\mathbf{y}_k^C \;\; \mathbf{Y}_k^C] = \left[\sum_{c_i \in C_k^V} w_{i,k}\mathbf{y}_{i,k} \quad \sum_{c_i \in C_k^V} w_{i,k}\mathbf{Y}_{i,k} \right], \qquad (1)$$

where $w_{i,k}$ represents the weight allotted to the information provided by node c_i and $\sum_{c_i \in C_k^V} w_{i,k} = 1$.

TABLE I: Table of notations

C	:	set of all nodes
k	:	time step
r_v	:	viewing range
r_c	:	communication range
C_k^S	:	set of sink nodes
C_k^I	:	set of inactive nodes
C_k^V	:	set of current viewing nodes
C_{k+}^V	:	set of future viewing nodes
C^N	:	set of neighbouring nodes (N-Nodes)
$N_{i,k}^N$	:	set of all N-Nodes in the communication range of c_i
N_i	:	set of single-hop neighbours of node c_i
$D_{i,k}^H$	:	hop distance of node c_i from the nearest current viewing node
D	:	threshold distance (in hops)
$\texttt{EIF}()$	:	Extended Information Filter
$\texttt{SEND}(\texttt{c}_\texttt{i}, \texttt{c}_\texttt{j}, \mathbf{m})$	:	node c_i sends message $\mathbf{m}$ to node c_j
$\texttt{RECV}(\texttt{c}_\texttt{i}, \texttt{c}_\texttt{j}, \mathbf{m})$	:	node c_i receives message $\mathbf{m}$ from node c_j
l	:	iteration index
L	:	maximum number of consensus iterations
$\Big\lvert \cdot \Big\rvert$	:	determinant function
$[\mathbf{y}_{i,k}^l \ \mathbf{Y}_{i,k}^l]$	:	posterior information (information vector and information matrix) available at c_i after l consensus iterations
$[\mathbf{y}_k^C \ \mathbf{Y}_k^C]$	:	posterior information computed in a centralised way
$\mathbf{z}_{i,k}$	:	measurement of node c_i
$R_{i,k}$	:	measurement noise covariance of node c_i
Q_k	:	process noise covariance

The posteriors, $[\mathbf{y}_{i,k} \ \mathbf{Y}_{i,k}]$, can be estimated using the Extended Information Filter (EIF) [13] when the target motion model, or camera measurement model, or both are non linear, and the noise in target motion (process noise) and the measurement noise are Gaussian with zero mean and covariances $\mathbf{Q}_k$ and $\mathbf{R}_{i,k}$, respectively. EIF computes the target posterior at time step k using $[\mathbf{y}_{i,k-1} \ \mathbf{Y}_{i,k-1}]$, the posterior at $k-1$, and $\mathbf{z}_{i,k}$, the measurement at k.

The weights, $w_{i,k}$, used during the fusion of posteriors vary depending on the fusion algorithm. Improved Fast Covariance Intersection (IFCI) [12] computes the weights based on the determinants of the information matrices and produces better fusion estimates than Fast Covariance Intersection [11].

B. Node types

We define Neighbouring nodes (N-Nodes) of the target at time step k, C_k^N, all nodes that are viewing the target at the current time step, *current viewing nodes*, C_k^V, and the nodes that might view the target at the next time step, *future viewing nodes*, C_{k+}^V. Nodes other than N-Nodes are *inactive nodes*, C_k^I and they do not participate in the consensus process. The inactive nodes that are single-hop neighbours of N-Nodes act as border between the N-Nodes and the remaining inactive nodes by not transmitting any information. We call them *sink nodes*, C_k^S, because they always receive information (from N-Nodes) but they do not send it. The relation between C, C_k^N, C_k^V, C_{k+}^V, C_k^I and C_k^S is as follows:

$$C = \underbrace{C_k^V \bigcup C_{k+}^V}_{C_k^N} \bigcup C_k^I \text{ and } C_k^S \subseteq C_k^I.$$

A constraint for using N-consensus is that the N-Nodes of a target must be connected at any time step. The notations used in this paper are listed in Table I.

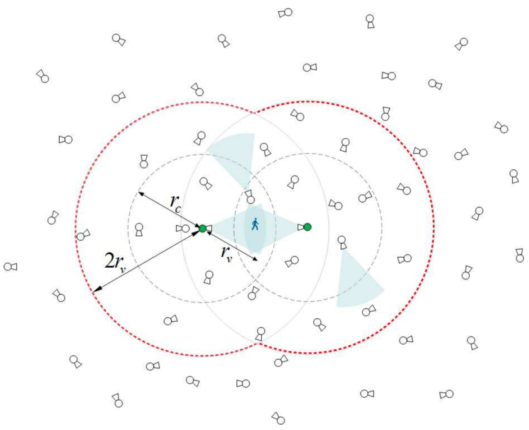


Fig. 1: Illustration of the consensus neighbourhood (red) of a target. A network of wireless camera nodes, field of view (blue) of four nodes, viewing range, r_v, communication range (dotted grey), r_c, of the current viewing nodes (green) are shown.

Ideally, the future viewing nodes include the nodes having overlapping FOVs with all the current viewing nodes. However, nodes are unaware of the FOV information of other nodes so we consider all nodes that are located within twice the viewing range, $2r_v$, distance from each current viewing node as future viewing nodes. We select the value $2r_v$ because the maximum possible physical distance between two nodes with overlapping FOVs is $2r_v$ [14], and a viewing node passing information to all nodes within $2r_v$ guarantees that the information is available at all current and future viewing nodes. Fig. 1 shows a scenario where the number of current viewing nodes is 2. The N-Nodes (nodes with in $2r_v$ distance from the current viewing nodes) are surrounded by a red boundary.

C. Iterative algorithm

Relative distances among the nodes are required to check if a node is within $2r_v$ distance from any of the current viewing nodes. As the physical locations of the nodes are unknown, relative distance between two nodes can be approximated either using hop counts [15] or using radio signal strength [16]. We use the former and perform iterative limited-multi-hop search [17] to identify C_{k+}^V. The maximum possible hop count (hop distance) between a future viewing node and its nearest current viewing node $D = \lceil \dfrac{2r_v}{r_c} \rceil$. We use this value as threshold to identify future viewing nodes, which include 1-hop, 2-hop, ..., D-hop neighbours of each current viewing node. Note that we consider current viewing nodes as 0-hop neighbours. Consensus is achieved among the C_k^N nodes (N-Nodes). Note that, for various r_c and r_v values D varies as:

$$D \begin{cases} > 2, & \text{if } r_c < r_v, \\ = 2, & \text{if } r_v \leq r_c < 2r_v, \\ = 1, & \text{if } r_c \geq 2r_v. \end{cases}$$

At each time step, current viewing nodes initialise their hop distance, $D_{i,k}^H$, to zero and compute the local posterior, $[\mathbf{y}_{i,k} \ \mathbf{Y}_{i,k}]$, using EIF [13] to handle the non-linearities. Each non-viewing node (node with no target measurement) identifies itself as an inactive node and initialises its hop distance to

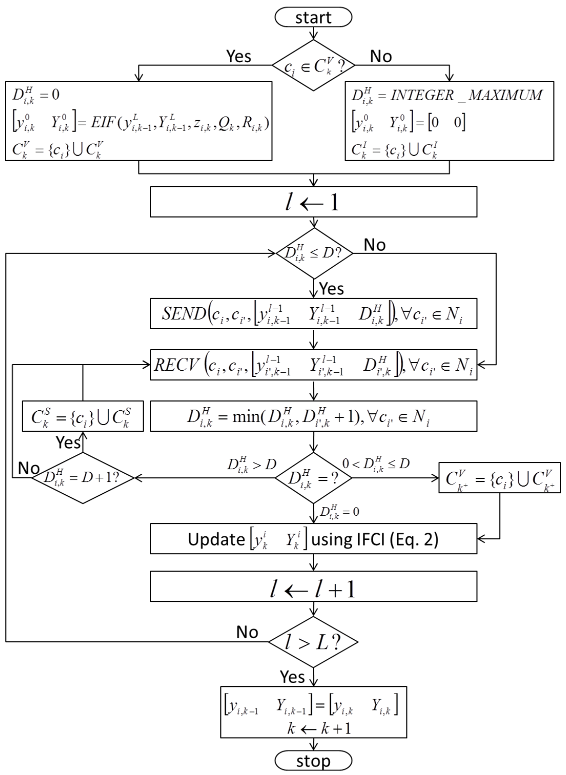

start
$c_i \in C_k^V$?
Yes
No
$D_{i,k}^H = 0$
$[y_{i,k}^0 \quad Y_{i,k}^0] = EIF(y_{i,k-1}^L, Y_{i,k-1}^L, z_{i,k}, Q_k, R_{i,k})$
$C_k^V = \{c_i\} \cup C_k^V$
$D_{i,k}^H = INTEGER\_MAXIMUM$
$[y_{i,k}^0 \quad Y_{i,k}^0] = [0 \quad 0]$
$C_k^I = \{c_i\} \cup C_k^I$
$l \leftarrow 1$
$D_{i,k}^H \le D$?
No
Yes
$SEND(c_i, c_{i'}, [y_{i,k-1}^{l-1} \quad Y_{i,k-1}^{l-1} \quad D_{i,k}^H]), \forall c_{i'} \in N_i$
$RECV(c_i, c_{i'}, [y_{i',k-1}^{l-1} \quad Y_{i',k-1}^{l-1} \quad D_{i',k}^H]), \forall c_{i'} \in N_i$
$C_k^S = \{c_i\} \cup C_k^S$
$D_{i,k}^H = \min(D_{i,k}^H, D_{i',k}^H + 1), \forall c_{i'} \in N_i$
Yes
No
$D_{i,k}^H = D+1$?
$D_{i,k}^H > D$
$D_{i,k}^H = ?$
$0 < D_{i,k}^H \le D$
$C_{k^+}^V = \{c_i\} \cup C_{k^+}^V$
$D_{i,k}^H = 0$
Update $[y_k^i \quad Y_k^i]$ using IFCI (Eq. 2)
$l \leftarrow l+1$
No
$l > L$?
Yes
$[y_{i,k-1} \quad Y_{i,k-1}] = [y_{i,k} \quad Y_{i,k}]$
$k \leftarrow k+1$
stop

Fig. 2: Flow diagram of N-consensus

infinity and its local posterior to $[\mathbf{0}\ \mathbf{0}]$ (Fig. 3a). Current viewing nodes initiate the iterative process of information exchange and consensus update.

During the information exchange step, all identified N-Nodes send messages containing their local posterior, $[\mathbf{y}_{i,k}\ \mathbf{Y}_{i,k}]$, and hop distance, $D^H_{i,k}$, to all their neighbours. Each receiving node within the communication range increments the received hop distance by one (i.e. $D^H_{i,k}+1$) and uses the value as a hop distance proposal made by the sender. The receiving nodes update their hop distances to the minimum of their local hop distance and the hop distance proposals. C^V_{k+} and C^S_k are updated based on the new hop distances.

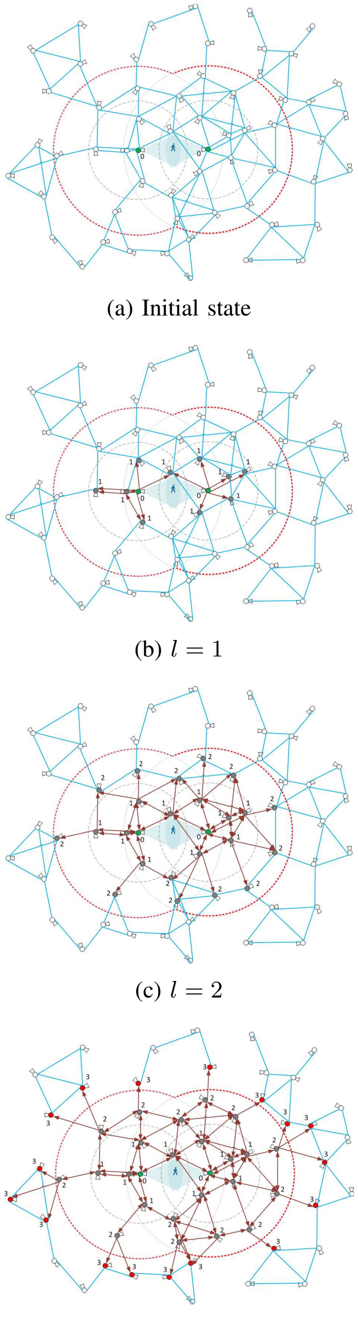
In the consensus update step, all identified N-Nodes fuse their local posteriors with the received posteriors using the IFCI algorithm [12] as follows:

$$[\mathbf{y}^l_{i,k} \quad \mathbf{Y}^l_{i,k}] = \left[\sum_{c_{i'} \in N^N_{i,k}} w^l_{i',k} \mathbf{y}^{l-1}_{i',k} \quad \sum_{c_{i'} \in N^N_{i,k}} w^l_{i',k} \mathbf{Y}^{l-1}_{i',k} \right], \quad (2)$$

where $w^l_{i',k}$ is

$$\frac{\left| \sum_{c_{j'} \in N^N_{i,k}} Y^{l-1}_{j',k} \right| + \left| Y^{l-1}_{i',k} \right| - \left| \sum_{c_{j'} \in N^N_{i,k}} Y^{l-1}_{j',k} - Y^{l-1}_{i',k} \right|}{\sum_{c_j \in N^N_{i,k}} \left[\left| \sum_{c_{j'} \in N^N_{i,k}} Y^{l-1}_{j',k} \right| + \left| Y^{l-1}_{j,k} \right| - \left| \sum_{c_{j'} \in N^N_{i,k}} Y^{l-1}_{j',k} - Y^{l-1}_{j,k} \right| \right]}.$$

Here $|\cdot|$ represents determinant and l is the iteration index. N-consensus algorithm is more complex than A-consensus and trace based ICI because of the computation of determinants.


(a) Initial state

(b) $l = 1$

(c) $l = 2$

(d) $l = 3, 4, ..., L$

Fig. 3: Illustration of N-consensus when the threshold hop distance $D = 2$. Red: consensus region, blue: connectivity among the nodes, brown: information flow, green: current viewing nodes, grey: future viewing nodes, red: sink nodes, white: inactive nodes. (a) Initially only the current viewing nodes identify themselves as N-Nodes. (b), (c) In each consensus iteration ($l \le D$) the neighbourhood of already known N-Nodes identify themselves as future viewing nodes based on their hop distance (shown next to each node). (d) During $l = 3$, the neighbourhood of known N-Nodes at 3-hop distance ($D^H_{i,k} > D$) identify themselves as sink nodes. The nodes status does not change when $l > 3$.

In iterations $l = 1, 2, ..., D$, future viewing nodes that are 1-hop, 2-hop, ..., D-hop neighbours are identified. When $l = D + 1$, sink nodes are identified. When $l > D + 1$, the sets $C_k^V, C_{k+}^V, C_k^I, C_k^S$ do not change. Fig. 2 shows the flow of operations of N-consensus and how different types of nodes are identified.

Between two time steps, L consensus iterations are run. Fig. 3 illustrates the N-consensus iterations with an example. When using less iterations than the threshold (i.e. $L < D$), the posterior might not be available at all future viewing nodes. For example, in Fig. 3b, $l = 1$ and $D = 2$, all 1-hop neighbours are identified but 2-hop neighbours are not yet identified so the target posterior is not available at the 2-hop neighbours. In the next time step, if the target enters the FOV of any 2-hop neighbour, the EIF running at the node fails to compute the posterior because the posterior from the previous time step is not available. Hence, N-consensus can not perform tracking unless a minimum of $L = D$ iterations are used. Running the consensus algorithm for D iterations ensures the identification of all N-Nodes and for $D+1$ iterations ensures the information exchange by all the N-Nodes. For example, in Fig. 3, though all N-Nodes are identified during iteration $l = 2$ (Fig. 3c), some neighbouring N-Nodes started exchanging information during iteration 3 (Fig. 3d). N-consensus ensures that the node having measurement at k but not at $k - 1$ holds the posterior from $k - 1$ because the node must have participated in consensus at $k - 1$ as a future viewing node.

III. Results and Discussion

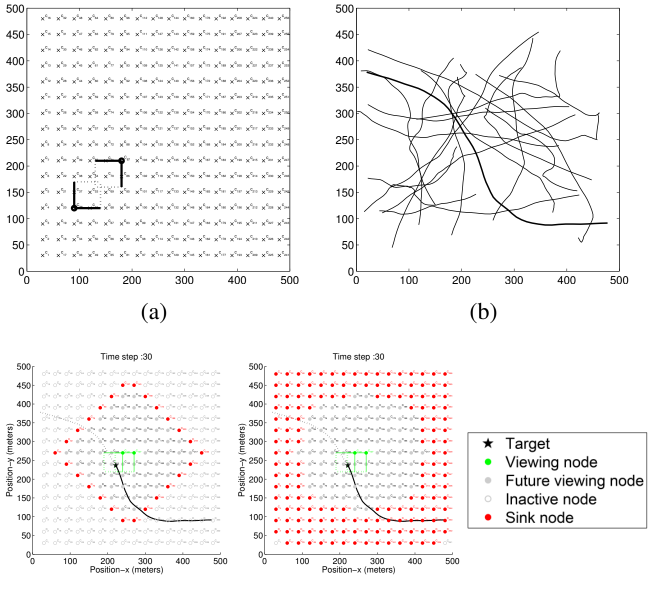
We compare the performance of four fusion algorithms, namely centralised fusion using Improved Fast Covariance Intersection (CCI) [12], distributed fusion using A-consensus (AC) [8], Iterative Covariance Intersection (ICI) [10] and the proposed N-consensus (NC) using numerical simulations.

A. Performance measures

We use as performance measures accuracy and communication cost. At each time step, the average of the position estimates of all the N-Nodes is considered as the estimated target position. *Accuracy* is the Euclidean distance between the estimated target positions and the corresponding ground-truth positions on the ground plane. *Communication cost* can be evaluated either as the total number of scalars transmitted in the network or as energy consumption for their transmission and reception. The energy spent not only depends on the number of scalars (or number of bits) transmitted but also on the communication range of each transmitting node. The energy is calculated by summing transmission energy $E_T = E_e p + \epsilon_a p r_c^2$ and receiving energy $E_R = E_e p$, where E_e is the electrical energy ($Jouls/bit$) used for running transmitter or receiver components, ϵ_a is the power amplification ($Jouls/bit/m^2$) required to guarantee acceptable received signal strength within the communication range r_c and, p is the number of bits transmitted or received [18].

B. Experimental setup

Let the WCN contain 256 homogeneous cameras that monitor a $500m \times 500m$ area. Each camera has a (directional) viewing range $r_v = 50m$ and 90^o FOV. The position and FOV of each camera are kept constant (Fig. 4a). A target whose state vector at time step k is given by $\mathbf{x}_k = [\begin{array}{ccccc} x_k & y_k & v_x & v_y & \delta_k \end{array}]^T$ and motion model



(a) (b)

(c) (d)

Fig. 4: Simulation setup. (a) Wireless camera network surveilling a $500m \times 500m$ area using cameras $c_1, ..., c_{256}$. Each camera has a Field of View (FOV) of $50m \times 50m$ on the ground plane (black). (b) Sample trajectories used in the experiments. (c) and (d) N-Nodes at time step 30 of the bold track shown in (b) when the communication range of nodes are $r_c = 30m$ ($D = 5$) and $r_c = 150m$ ($D = 1$), respectively.

$$\mathbf{x}_{k+1} = \left[\begin{array}{c} x_k + v_{x,k}\delta_k + a_x \dfrac{\delta_k^2}{2} \\ y_k + v_{y,k}\delta_k + a_y \dfrac{\delta_k^2}{2} \\ v_{x,k} + a_x \delta_k \\ v_{y,k} + a_y \delta_k \\ \delta_k + e \end{array} \right] \quad (3)$$

is considered, where (x_k, y_k) and $(v_{x,k}, v_{y,k})$ are the target position and velocity on the ground plane, respectively; and δ_k is the time step between two consecutive measurements [8]. The vector $\mathbf{w} = (a_x, a_y, e)$ is the process noise, (a_x, a_y) is the target acceleration and e is the synchronisation error among cameras. We use 20 trajectories (Fig. 4b) for performance analysis. The measurement model of the camera c_i is:

$$\mathbf{z}_{i,k} = \left[\begin{array}{c} u_{i,k} \\ v_{i,k} \end{array} \right] = \left[\begin{array}{c} \dfrac{H_{11}^i x_k + H_{12}^i y_k + H_{13}^i}{H_{31}^i x_k + H_{32}^i y_k + H_{33}^i} \\ \dfrac{H_{21}^i x_k + H_{22}^i y_k + H_{23}^i}{H_{31}^i x_k + H_{32}^i y_k + H_{33}^i} \end{array} \right] + \mathbf{v}_k, \quad (4)$$

where $(u_{i,k}, v_{i,k})$ are the pixel coordinates of the target in the image plane of camera c_i at time step k. The values $H_{11}^i, ..., H_{33}^i$ are the elements of the homography matrix H^i and are taken from one of the cameras of APIDIS dataset[1] as:

[1] http://sites.uclouvain.be/ispgroup/index.php/Softwares/APIDIS, last accessed February 2015.

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

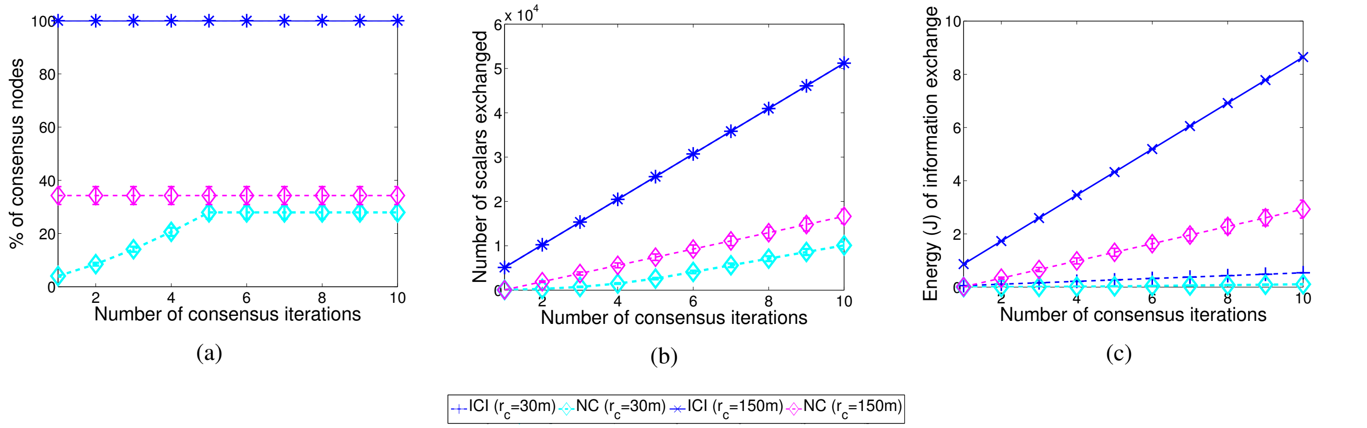
$\mathbf{v}_k$ is the measurement noise. The non-additive process noise and the additive measurement noise are assumed to be zero mean Gaussians with covariances $\mathbf{Q}_k = diag[1\ 1\ 0.0001]$ and $\mathbf{R}_{i,k} = diag[15\ 15], \forall c_i \in C$ respectively at all time steps.

At each time step the local posteriors (to be fused) are estimated by EIF running at each node. Both in A-consensus and ICI, non-viewing nodes use predicted posteriors as their local posteriors. We use $\frac{0.65}{\Delta_{max}}$ as the weight of each neighbour's information for A-consensus update assuming that each camera knows the maximum degree (Δ_{max}) of the underlying communication graph [3]. The tracking experiment is conducted by considering the communication range r_c of each node as $30m$ ($< r_v$) and $150m$ ($> 2r_v$). We analyse the accuracy and communication cost of tracking the 20 trajectories for different number of consensus iterations, L in both cases.

C. Discussion

The hop distance thresholds $D = \lceil \frac{2r_v}{r_c} \rceil$ are 5 and 1 when the r_c values are $30m$ and $150m$, respectively. The N-Nodes in each case are shown in Fig. 4c and 4d. More nodes are identified as N-Nodes for $r_c = 150m$ compared to $r_c = 30m$ because the higher communication range turns more nodes to be 1-hop (future viewing nodes) and 2-hop neighbours (sink nodes). N-Nodes are not completely identified until D iterations, so for $r_c = 30m$ the percentage of N-Nodes increases for iterations 1 to 5 and from the 5^{th} iteration the value does not change. Note that the increment is not linear and depends on the number of newly identified future viewing nodes in each iteration. Fig. 5a shows the percentage of nodes participating in consensus in ICI and N-consensus. By using N-consensus, the number of participating nodes is reduced to approximately 35%.

Each transmission of N-consensus involves a message containing information vector $\mathbf{y}_k^i$, information matrix $\mathbf{Y}_k^i$ and hop distance. The target state vector size is 5 and the information matrix is symmetric. To reduce the number of scalar transmissions we send only the upper triangular values, i.e. 15 scalars.

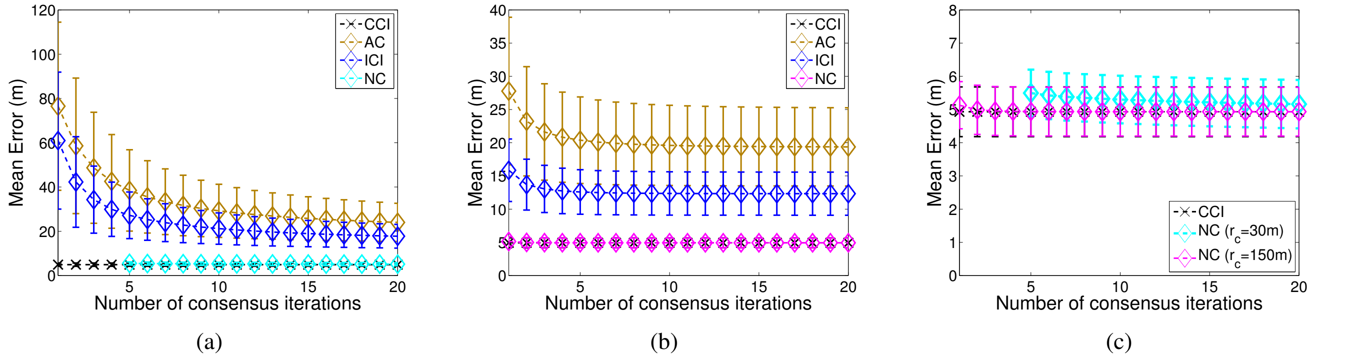


(a)

(b)

(c)

Fig. 5: Communication cost analysis. (a) Average percentage of nodes participating in consensus for the two r_c values ($30m$ and $150m$). Average communication cost in terms of (b) number of scalars transmitted and (c) energy (Jouls) spent. The algorithms are Iterative Covariance Intersection (ICI) [10] and the proposed Neighbourhood-consensus (NC). Note that the results of ICI for the two r_c values overlap. As the results of Average Consensus (AC) [8] are the same as ICI, they are not reported here.



(a)

(b)

(c)

Fig. 6: Accuracy analysis when the communication range, r_c, of the nodes is (a) $30m$ and (b) $150m$. The algorithms are: Centralised fusion using IFCI (CCI) [12], distributed fusion using Average Consensus (AC) [8], Iterative Covariance Intersection (ICI) [10] and the proposed Neighbourhood-consensus (NC). (c) is the zoom of (a) and (b) with focus on NC.

Hence, the posterior ($[\mathbf{y}_{i,k}\ \mathbf{Y}_{i,k}]$) contains 20 scalars each of which is a 32-bit floating point number. The hop distance is a 16-bit integer. Therefore, each N-consensus message contains $(20 \times 32) + (1 \times 16) = 656$ bits. The values of E_e and ϵ_a are considered as $50nJ$ and $0.1nJ/b/m^2$, respectively. As only the N-Nodes participate in consensus, the number of transmissions is smaller than that of ICI and A-consensus in which all nodes participate. The total communication cost is therefore smaller in N-consensus than that of ICI and A-consensus. As the N-Nodes are more for $r_c = 150m$ than for $r_c = 30m$, the number of scalars transmitted are also more for $r_c = 150m$ than for $r_c = 30m$ (Fig. 5b) of N-consensus. The energy consumption is also smaller in N-consensus than that of ICI and A-consensus. While the number of scalars transmitted by ICI (and A-consensus) is the same for the two communication ranges, the energy spent is different because of the different transmission ranges. As one would expect, N-consensus with $r_c = 30m$ consumes less energy than N-consensus with $r_c = 150m$ and ICI (and A-consensus) with both the communication ranges (Fig. 5c) because of the smaller number of N-Nodes and the lower transmission range. The communication cost of A-consensus and ICI are the same so the cost of A-consensus is not shown in Fig. 5.

As mentioned in Section II-C, N-consensus cannot perform tracking until all N-Nodes are identified so tracking is not feasible when using consensus iterations less than D. For $r_c = 30m$, until the 5^{th} iteration the error is not available (Fig. 6). For $r_c = 150m$, $D = 1$ so tracking is performed for all the iterations used. The N-consensus estimate achieves faster convergence to the centralised estimate compared to the other algorithms. The error computed using CCI is $4.9m$. N-consensus with $r_c = 150m$ spends $0.34J$ energy for 2 iterations and has a mean error $5.0m$, whereas N-consensus with $r_c = 30m$ spends $0.26J$ for 20 iterations and has a mean error $5.1m$.

IV. Conclusions

We proposed N-consensus, an algorithm for achieving consensus on target posteriors among only a set of Neighbouring Nodes (N-Nodes) of the target. N-Nodes include current viewing nodes (0-hop neighbours) and future viewing nodes (1-hop, 2-hop, ..., D-hop neighbours). We select D based on the viewing and the communication ranges of the nodes. Consensus update fuses posteriors using the covariance intersection algorithm to avoid the necessity of topology information and provides better fusion estimates. Experimental results show that the proposed N-consensus approach provides better accuracy and requires less communication resources compared to average consensus and iterative covariance intersection. Unlike other consensus approaches, in N-consensus, the target state is available at N-Nodes only so non N-Nodes (inactive nodes) can not take decisions about the target.

The optimal set of future viewing nodes includes only the nodes having overlapping field of view with all the current viewing nodes. The proposed algorithm considers more future viewing nodes than the optimal case making the set suboptimal. As a future work, N-Nodes could be selected based on a distributively computed vision graph [19] to generate the optimal set that will lead to saving of more resources than our approach.

Acknowledgments

S. 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. A. Cavallaro acknowledges the support of the Artemis JU and the UK Technology Strategy Board through the COPCAMS Project under Grant 332913.

References

[1] M. Taj and A. Cavallaro, “Distributed and decentralized multicamera tracking,” *IEEE Signal Proc. Mag.*, vol. 28, no. 3, pp. 46–58, 2011.

[2] S. Katragadda, J. C. SanMiguel, and A. Cavallaro, “The costs of fusion in smart camera networks,” in *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC)*, Venezia Mestre, Italy, 2014.

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

[4] G. Battistelli and L. Chisci, “Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,” *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[5] D. Gu, J. Sun, Z. Hu, and H. Li, “Consensus based distributed particle filter in sensor networks,” in *Proc. of Int. Conf. on Information and Automation*, Changsha, China, 2008, pp. 302–307.

[6] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, “Collaborative sensing in a distributed PTZ camera network.” *IEEE Trans. on Image Processing*, vol. 21, no. 7, pp. 3282–3295, 2012.

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

[8] S. Katragadda, J. C. SanMiguel, and A. Cavallaro, “Consensus protocols for distributed tracking in wireless camera networks,” in *Proc. of Int. Conf. on Information Fusion*, Salamanca, Spain, 2014.

[9] 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*, Los Angeles, CA, USA, 2005, pp. 63–70.

[10] O. Hlinka, O. Sluciak, F. Hlawatsch, and M. Rupp, “Distributed data fusion using iterative covariance intersection,” in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 1861–1865.

[11] W. Niehsen, “Information fusion based on fast covariance intersection filtering,” in *Proc. of Int. Conf. on Information Fusion*, Annapolis, MD, USA, 2002, pp. 901–904.

[12] D. Franken and A. Hupper, “Improved fast covariance intersection for distributed data fusion,” in *Proc. of Int. Conf. on Information Fusion*, Philadelphia, PA, USA, 2005, pp. 154–160.

[13] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, 1st ed. CRC Press, Inc., 1998.

[14] W. Li, “Camera sensor activation scheme for target tracking in wireless visual sensor networks,” *Int. Jour. of Distributed Sensor Networks*, vol. 2013, no. 397537, pp. 1–11, 2013.

[15] X. Dong and M. C. Vuran, “Vision graph construction in wireless multimedia sensor networks,” in *Proc. of IEEE Global Telecommunications Conf. (GLOBECOM)*, Miami, FL, USA, Dec 2010.

[16] D. Niculescu and B. Nath, “DV based positioning in ad hoc networks,” *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[17] R. Nagpal, H. Shrobe, and J. Bachrach, “Organizing a global coordinate system from local information on an ad hoc sensor network,” in *Proc. of Int. Conf. on Information Processing in Sensor Networks (IPSN)*, Palo Alto, CA, USA, 2003, pp. 333–348.

[18] A. Wang and A. Chandrakasan, “Energy-efficient DSPs for wireless sensor networks,” *IEEE Signal Proc. Mag.*, vol. 19, no. 4, pp. 68–78, 2002.

[19] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, “Socio-economic vision graph generation and handover in distributed smart camera networks,” *ACM Trans. Sen. Netw.*, vol. 10, no. 2, pp. 20:1–20:24, 2014.