AVERAGE CONSENSUS-BASED ASYNCHRONOUS TRACKING

Sandeep Katragadda[1], *Carlo S. Regazzoni*[2], *Andrea Cavallaro*[1]

[1]Centre for Intelligent Sensing, Queen Mary University of London, UK
[2]DITEN, University of Genoa, Italy

ABSTRACT

Target tracking in a network of wireless cameras may fail if data are captured or exchanged asynchronously. Unlike traditional sensor networks, video processing may generate significant delays that also vary from camera to camera. Moreover, the continuous and rapid change of the dynamics of the consensus variable (the target state) makes tracking even more challenging under these conditions. To address this problem, we propose a consensus approach that enables each camera to predict information of other cameras with respect to its own capturing time-stamp based on the received information. This prediction is key to compensate for asynchronous data exchanges. Simulations show the performance improvement with the proposed approach compared to the state of the art in the presence of asynchronous frame captures and random processing delays.

Index Terms— Distributed tracking, camera networks, asynchronous fusion, average consensus

1. INTRODUCTION

Distributed tracking in a wireless camera network (WCN) involves the estimation of the target state (e.g. location) via data exchange and fusion among cameras. Unlike other distributed fusion algorithms [1, 2, 3, 4], consensus-based algorithms do not require full connectivity nor prior knowledge of the routing tables [5]. In consensus-based fusion each camera node sends its local information to its neighbours. Nodes update their local information by fusing it with the received information, and send the updated information to their neighbours. Information exchange and update (consensus update) iterate until the network converges [6]. The final fusion result is then available at all nodes.

Most distributed tracking algorithms assume that the cameras in the network capture the frames synchronously [7, 8, 9, 10, 11]. However, local clock frequencies may drift between 1 and 100 parts per million (ppm) [12, 13]. While time synchronisation protocols can estimate and compensate for the timing offsets, they significantly increase the communication overhead and therefore energy consumption, thus reducing the network lifetime [14, 1]. Moreover, local frame-*processing delays* are not negligible (≈40ms) and may vary from node to node as a function of the number of observed targets [15]. Even if delays are comparable among camera nodes, the local information in a camera corresponds to the instant of capturing and not to the instant of transmission. If the local information is assumed to be synchronous and to correspond to the transmission instant, fusion will *decrease* tracking accuracy.

The asynchronous Consensus-based Distributed Target Tracking (aCDTT) is a maximum consensus-based approach that makes the network converge to the maximum certain state among the local states [16]. Because aCDTT does not fuse local information of the cameras, it does not reduce the uncertainty on the state. Moreover, the nodes may update their local estimates with the selected estimate that might correspond to a different time instant. The Information Consensus Filter (ICF) is an average consensus-based approach that makes the network converge to the Kullback-Leibler Average (KLA) of the local estimates [9, 17]. While the nodes fuse local information and thus can reduce the uncertainty on the state, ICF works only in synchronous settings. Average consensus methods exist that work with asynchronous communications [18, 19, 20, 21, 22, 23] and dynamic changes in the consensus variable [24]. However, they cannot be applied to distributed asynchronous tracking as they do not consider the continuously changing dynamics of the consensus variable (the target state in our case).

In this paper, we propose an *Average Consensus-based Asynchronous tracking Filter* (ACAF) for WCNs. The nodes perform time-alignment with respect to their capturing instant by predicting the information of the neighbours. Each node performs iteratively two phases, namely estimation and fusion. The estimation phase computes the measurement (pixel coordinates of the target) from the captured frame and estimates the target state using the measurement and the previously estimated probability density function (pdf) at the node. Then the fusion phase exchanges, aligns and fuses the local pdfs. The expected pdf of the sending node is estimated by a receiving node with a time-reversed prediction. The nodes discard the pdfs received before the frame capture and terminate their fusion phase before any other node captures the next frame. The termination criterion is decided based on its local processing time. The software of the proposed method is available at *http://www.eecs.qmul.ac.uk/~andrea/software.htm*.

2. PRELIMINARIES

Let camera C^i capture frame $\mathbf{I}_k^i$ at time k, perform target detection and compute the measurement $\mathbf{z}_k^i$. Let $f(\hat{\mathbf{x}}_{k'}^i|\mathbf{z}_{1:k'}^i)$ be the pdf of C^i corresponding to its previous capturing instant k'. In the estimation phase, C^i runs a local Bayesian filter to compute the current pdf $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)$ using the previous known pdf $f(\hat{\mathbf{x}}_{k'}^i|\mathbf{z}_{1:k'}^i)$, the state transition pdf $f(\mathbf{x}_k^i|\mathbf{x}_{k'}^i)$ and the likelihood pdf $f(\mathbf{z}_k^i|\mathbf{x}_k^i)$. If C^i does not have a measurement due to its limited Field of View (FoV), it predicts the pdf $f(\tilde{\mathbf{x}}_k^i|\mathbf{z}_{1:k'}^i)$ at k and uses it as the local pdf $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)$.

Let $\mathbf{C} = \left\{C^1, C^2, ..., C^N\right\}$ be a WCN with N cameras and $\mathcal{N}^i$ be the set of cameras in the communication range of C^i. Let us assume that the communication is ideal (no communication delays). The WCN tracks a target moving on a ground plane. Each camera C^i will estimate the target state $\hat{\boldsymbol{\mu}}_k^i$ corresponding to its capturing instant k by fusing the local pdfs (i.e. $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i), \forall i$) in a distributed way under bounded random estimation delays.

S. Katragadda was 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.

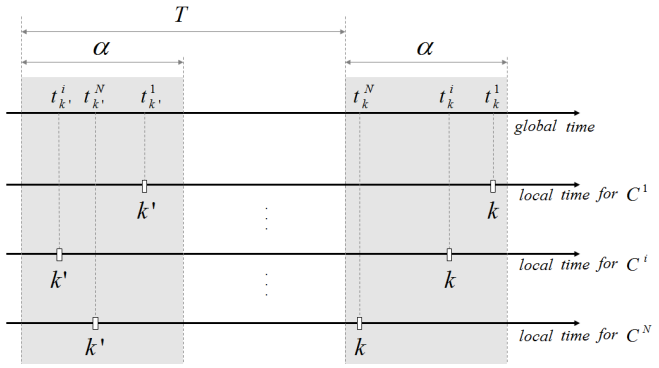

Fig. 1: Local and global clocks in partial asynchronism. The vertical grey stripes associate the local time instants. Key – T: inter-frame capturing period, α: amount of asynchronism.

The target state at time k is $\mathbf{x}_k = [x_k\ y_k\ \dot{x}_k\ \dot{y}_k]$ where $[x_k\ y_k]$ and $[\dot{x}_k\ \dot{y}_k]$ are the position and velocity of the target at k on the ground plane, respectively. Let the target dynamics for a temporal interval Δk be

$$\mathbf{x}_{k+\Delta k} = \mathbf{F}(k, k+\Delta k)\mathbf{x}_k + \mathbf{w}(k, k+\Delta k), \tag{1}$$

where $\mathbf{x}_{k+\Delta k}$ is the target state at $k+\Delta k$, $\mathbf{F}(k, k+\Delta k)$ is the state transition function from k to $k+\Delta k$ and $\mathbf{w}(k, k+\Delta k)$ is the cumulative effect of the process noise from k to $k+\Delta k$. The process noise is assumed to be Gaussian with zero mean and covariance matrix $\mathbf{Q}(k, k+\Delta k)$.

Let t_k^i be the global-clock time when the local-clock time of C^i is k and T be the desired inter-frame capturing period for all cameras. In the synchronous case, $t_k^i = k, \forall i$ and the cameras capture frames at $\{0, T, 2T, ...\}$. In the asynchronous case, the cameras capture frames at different instants. We assume the asynchronism to be partial (as in [16]), where α is the upper bound of the relative capturing offset (see Fig. 1). Each camera knows T and α.

3. PREDICTION AND FUSION

At the beginning of the fusion phase, each node C^i performs three predictions. The first predicts the target pdf for $k-\alpha$ based on the computed local pdf $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)$ (backward prediction) as

$$f(\tilde{\mathbf{x}}_{k-\alpha}^i|\mathbf{z}_{1:k}^i) = \int f(\tilde{\mathbf{x}}_{k-\alpha}^i|\mathbf{x}_k^i) f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i) d\mathbf{x}_k^i. \tag{2}$$

We use $k-\alpha$ because other cameras in the network must have captured at most α time steps earlier or at most α time steps later (partial asynchronism assumption). The second prediction is based on the previous known pdf $f(\hat{\mathbf{x}}_{k'}^i|\mathbf{z}_{1:k'}^i)$ (forward prediction) as

$$f(\tilde{\mathbf{x}}_{k-\alpha}^i|\mathbf{z}_{1:k'}^i) = \int f(\tilde{\mathbf{x}}_{k-\alpha}^i|\hat{\mathbf{x}}_{k'}^i) f(\hat{\mathbf{x}}_{k'}^i|\mathbf{z}_{1:k'}^i) d\hat{\mathbf{x}}_{k'}^i. \tag{3}$$

The predicted pdfs $f(\tilde{\mathbf{x}}_{k-\alpha}^i|\mathbf{z}_{1:k}^i)$ and $f(\tilde{\mathbf{x}}_{k-\alpha}^i|\mathbf{z}_{1:k'}^i)$ are then compared and the one with the lowest uncertainty is considered. This step helps to avoid over-prediction when a camera does not have measurements due to an occlusion or the limited (directional) FoV. Let $f(\tilde{\mathbf{x}}_{k-\alpha}^{i,*}|\mathbf{z}_{1:k}^i)$ be the pdf with lower uncertainty. The third predicts the pdf for the capturing instant k based on the certain predicted pdf corresponding to $k-\alpha$ (forward prediction) as

$$f(\tilde{\mathbf{x}}_{k(\tau_k^i)}^i|\mathbf{z}_{1:k}^i) = \int f(\tilde{\mathbf{x}}_{k(\tau_k^i)}^i|\tilde{\mathbf{x}}_{k-\alpha}^{i,*}) f(\tilde{\mathbf{x}}_{k-\alpha}^{i,*}|\mathbf{z}_{1:k}^i) d\tilde{\mathbf{x}}_{k-\alpha}^{i,*}. \tag{4}$$

In other words, we predict the target pdf for the same capturing instant k via backward and forward predictions. The subscript $k(l)$ is used to indicate that the information corresponds to k before starting the consensus iteration at $k+l$ ($l \geq \tau_k^i$). τ_k^i is the estimation delay of C^i at k.

Camera nodes fuse these predicted pdfs $f(\tilde{\mathbf{x}}_{k(\tau_k^i)}^i|\mathbf{z}_{1:k}^i), \forall C^i$ via distributed average consensus. Let γ be the periodicity of the consensus iterations. Each node can compute the elapsed time after an estimation phase and after each consensus iteration.

Each consensus iteration involves two predictions, one before the transmission and one after the reception. Before transmission, each node C^i predicts the pdf for the transmission instant $k+l$ based on the predicted pdf $f(\tilde{\mathbf{x}}_{k(l)}^i|\mathbf{z}_{1:k}^i)$ corresponding to the capturing instant k (forward prediction) as

$$f(\tilde{\mathbf{x}}_{k+l}^i|\mathbf{z}_{1:k}^i) = \int f(\tilde{\mathbf{x}}_{k+l}^i|\tilde{\mathbf{x}}_{k(l)}^i) f(\tilde{\mathbf{x}}_{k(l)}^i|\mathbf{z}_{1:k}^i) d\tilde{\mathbf{x}}_{k(l)}^i. \tag{5}$$

The predicted pdf $f(\tilde{\mathbf{x}}_{k+l}^i|\mathbf{z}_{1:k}^i)$ represents the opinion of the sender C^i at the transmission instant. The node C^i sends the predicted pdf $f(\tilde{\mathbf{x}}_{k+l}^i|\mathbf{z}_{1:k}^i)$ to its neighbours $\mathcal{N}^i$.

If C^i receives a similar predicted pdf $f(\tilde{\mathbf{x}}_{k''}^j|\mathbf{z}_{1:k''}^j)$ from C^j at any local time k'' after its capturing instant (i.e. $k'' \in [k, k+l]$), it stores the received pdf in a buffer. During the consensus update at $k+l$, C^i predicts the pdf of C^j for C^i's capturing instant k based on the received pdf (reverse prediction) as

$$f(\tilde{\mathbf{x}}_{k(l)}^j|\mathbf{z}_{1:k''}^j) = \int f(\tilde{\mathbf{x}}_k^j|\tilde{\mathbf{x}}_{k''}^j) f(\tilde{\mathbf{x}}_{k''}^j|\mathbf{z}_{1:k''}^j) d\tilde{\mathbf{x}}_{k''}^j. \tag{6}$$

This predicted pdf represents the opinion of the sender C^j for the capturing instant k of the receiver C^i. Now, C^i fuses the time-aligned pdfs $f(\tilde{\mathbf{x}}_{k(l)}^i|\mathbf{z}_{1:k}^i)$ and $f(\tilde{\mathbf{x}}_{k(l)}^j|\mathbf{z}_{1:k''}^j), \forall C^j \in \mathcal{N}^i$ as

$$f(\tilde{\mathbf{x}}_{k(l+\gamma)}^i|\mathbf{z}_{1:k}^i) = \frac{f(\tilde{\mathbf{x}}_{k(l)}^i|\mathbf{z}_{1:k}^i)^\epsilon f(\tilde{\mathbf{x}}_{k(l)}^j|\mathbf{z}_{1:k''}^j)^{1-\epsilon}}{\int f(\tilde{\mathbf{x}}_{k(l)}^i|\mathbf{z}_{1:k}^i)^\epsilon f(\tilde{\mathbf{x}}_{k(l)}^j|\mathbf{z}_{1:k''}^j)^{1-\epsilon} d\tilde{\mathbf{x}}_{k(l)}^i}, \tag{7}$$

where ϵ is the weight given to the instantaneous pdf of the node. The fusion happens for all the received neighbours' pdfs. The fusion result is used in the next consensus iteration that repeats (5), (6) and (7) at $k+l+\gamma$.

Asymptotically, the fusion result converges to the KLA of the predicted local pdfs of all the cameras, i.e.

$$f(\tilde{\mathbf{x}}_{k(\infty)}^+|\mathbf{z}_{1:k}^+) = \frac{\prod_{j=1}^N f(\tilde{\mathbf{x}}_{k(\tau_k^j)}^j|\mathbf{z}_{1:k}^j)^{\frac{1}{N}}}{\int \prod_{j=1}^N f(\tilde{\mathbf{x}}_{k(\tau_k^j)}^j|\mathbf{z}_{1:k}^j)^{\frac{1}{N}} d\tilde{\mathbf{x}}_{k(\tau_k^j)}^j}. \tag{8}$$

We use the superscript $+$ instead of a camera index to represent that the result is available at all cameras. As each node C^i is aware of its own capturing instant, it replaces its contribution in the KLA, i.e. the predicted local pdf $f(\tilde{\mathbf{x}}_{k(\tau_k^i)}^i|\mathbf{z}_{1:k}^i)$, with the actual local pdf $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)$. We refer to this step as the correction step and compute it as follows:

$$f(\hat{\mathbf{x}}_k^i|\mathbf{z}_{1:k}^i) = \frac{f(\tilde{\mathbf{x}}_{k(\infty)}^i|\mathbf{z}_{1:k}^i) f(\tilde{\mathbf{x}}_{k(\tau_k^i)}^i|\mathbf{z}_{1:k}^i)^{-\frac{1}{N}} f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)^{\frac{1}{N}}}{\int f(\tilde{\mathbf{x}}_{k(\infty)}^i|\mathbf{z}_{1:k}^i) f(\tilde{\mathbf{x}}_{k(\tau_k^i)}^i|\mathbf{z}_{1:k}^i)^{-\frac{1}{N}} f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)^{\frac{1}{N}} d\tilde{\mathbf{x}}_{k(\infty)}^i}. \tag{9}$$

To avoid the fusion of information corresponding to subsequent frame captures, a node terminates its consensus phase if the time elapsed since the frame capture is $T-\alpha$.

In the next section we derive an approximation of the above Bayesian fusion method under Gaussian assumptions.

4. ASYNCHRONOUS TRACKER

In the *estimation phase*, each node C^i computes the local pdf $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)$ represented by $(\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ using the Information Filter [25]. Here, $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\Sigma}_k^i$ represent the minimum mean square error estimate and the corresponding error covariance of the estimated target pdf $f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i)$, and are defined as

$$\begin{aligned}\boldsymbol{\mu}_k^i &= \int \mathbf{x}_k^i f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i) d\mathbf{x}_k^i,\\ \boldsymbol{\Sigma}_k^i &= \int (\mathbf{x}_k^i - \boldsymbol{\mu}_k^i)(\mathbf{x}_k^i - \boldsymbol{\mu}_k^i)^T f(\mathbf{x}_k^i|\mathbf{z}_{1:k}^i) d\mathbf{x}_k^i.\end{aligned} \tag{10}$$

The information pair corresponding to the estimate $(\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ is $(\mathbf{y}_k^i, \mathbf{Y}_k^i) = ({\boldsymbol{\Sigma}_k^i}^{-1}\boldsymbol{\mu}_k^i, {\boldsymbol{\Sigma}_k^i}^{-1})$.

In the *fusion phase*, each C^i performs backward prediction of the pair from k to $k-\alpha$ as

$$\begin{aligned}\tilde{\mathbf{Y}}_{k-\alpha|k}^i &= \left(\mathbf{F}(k,k-\alpha){\mathbf{Y}_k^i}^{-1}\mathbf{F}(k,k-\alpha)^T + \mathbf{Q}(k,k-\alpha)\right)^{-1},\\ \tilde{\mathbf{y}}_{k-\alpha|k}^i &= \tilde{\mathbf{Y}}_{k-\alpha|k}^i \mathbf{F}(k,k-\alpha)\left({\mathbf{Y}_k^i}^{-1}\mathbf{y}_k^i\right);\end{aligned} \tag{11}$$

and forward prediction of the pair from k' to $k-\alpha$ as

$$\begin{aligned}\tilde{\mathbf{Y}}_{k-\alpha|k'}^i &= \left(\mathbf{F}(k',k-\alpha){\hat{\mathbf{Y}}_{k'}^i}^{-1}\mathbf{F}(k',k-\alpha)^T + \mathbf{Q}(k',k-\alpha)\right)^{-1},\\ \tilde{\mathbf{y}}_{k-\alpha|k'}^i &= \tilde{\mathbf{Y}}_{k-\alpha|k'}^i \mathbf{F}(k',k-\alpha)\left({\hat{\mathbf{Y}}_{k'}^i}^{-1}\hat{\mathbf{y}}_{k'}^i\right).\end{aligned} \tag{12}$$

Here, $(\hat{\mathbf{y}}_{k'}^i, \hat{\mathbf{Y}}_{k'}^i)$ is the information pair of the known pdf corresponding to $k' < k$. As the certainty of a distribution is proportional to the trace of its information matrix, the information pair between $(\tilde{\mathbf{y}}_{k-\alpha|k}^i, \tilde{\mathbf{Y}}_{k-\alpha|k}^i)$ and $(\tilde{\mathbf{y}}_{k-\alpha|k'}^i, \tilde{\mathbf{Y}}_{k-\alpha|k'}^i)$ with the higher trace is considered. We represent the winning pair as $(\tilde{\mathbf{y}}_{k-\alpha}^{i,*}, \tilde{\mathbf{Y}}_{k-\alpha}^{i,*})$.

C^i performs forward prediction of the pair from $k-\alpha$ to k as

$$\begin{aligned}\tilde{\mathbf{Y}}_{k(\tau_k^i)}^i &= \left(\mathbf{F}(k-\alpha,k){\tilde{\mathbf{Y}}_{k-\alpha}^{i,*}}^{-1}\mathbf{F}(k-\alpha,k)^T + \mathbf{Q}(k-\alpha,k)\right)^{-1},\\ \tilde{\mathbf{y}}_{k(\tau_k^i)}^i &= \tilde{\mathbf{Y}}_{k(\tau_k^i)}^i \mathbf{F}(k-\alpha,k)\left({\tilde{\mathbf{Y}}_{k-\alpha}^{i,*}}^{-1}\tilde{\mathbf{y}}_{k-\alpha}^{i,*}\right).\end{aligned} \tag{13}$$

Each consensus iteration at $k+l$ ($l \geq \tau_k^i$) consists of four steps, namely forward prediction, time alignment, fusion and correction. The *forward prediction* of the pair from k to $k+l$ is performed as

$$\begin{aligned}\tilde{\mathbf{Y}}_{k+l}^i &= \left(\mathbf{F}(k,k+l){\tilde{\mathbf{Y}}_{k(l)}^i}^{-1}\mathbf{F}(k,k+l)^T + \mathbf{Q}(k,k+l)\right)^{-1},\\ \tilde{\mathbf{y}}_{k+l}^i &= \tilde{\mathbf{Y}}_{k+l}^i \mathbf{F}(k,k+l)\left({\tilde{\mathbf{Y}}_{k(l)}^i}^{-1}\tilde{\mathbf{y}}_{k(l)}^i\right).\end{aligned} \tag{14}$$

C^i transmits the pair $(\tilde{\mathbf{y}}_{k+l}^i, \tilde{\mathbf{Y}}_{k+l}^i)$ to its neighbours $\mathcal{N}^i$.

The second step is *time alignment*. Let $k'' \in [k, k+l]$ be the local time instant when C^i receives the pair $(\tilde{\mathbf{y}}_{k''}^j, \tilde{\mathbf{Y}}_{k''}^j)$ from C^j. C^i predicts the information pair of C^j for k via reverse prediction as

$$\begin{aligned}\tilde{\mathbf{Y}}_{k(l)}^j &= \mathbf{F}(k,k'')^T\left({\tilde{\mathbf{Y}}_{k''}^j}^{-1} - \mathbf{Q}(k,k'')\right)^{-1}\mathbf{F}(k,k''),\\ \tilde{\mathbf{y}}_{k(l)}^j &= \tilde{\mathbf{Y}}_{k(l)}^j \mathbf{F}(k'',k)\left({\tilde{\mathbf{Y}}_{k''}^j}^{-1}\tilde{\mathbf{y}}_{k''}^j\right).\end{aligned} \tag{15}$$

In the information *fusion* step the predicted pair $\left(\tilde{\mathbf{y}}_{k(l)}^i, \tilde{\mathbf{Y}}_{k(l)}^i\right)$ and the predicted pairs $\left(\tilde{\mathbf{y}}_{k(l)}^j, \tilde{\mathbf{Y}}_{k(l)}^j\right), \forall C^j \in \mathcal{N}^i$ are fused via the average consensus update as

$$\begin{aligned}\tilde{\mathbf{Y}}_{k(l+\gamma)}^i &= \tilde{\mathbf{Y}}_{k(l)}^i + \epsilon \sum_{\forall C^j \in \mathcal{N}^i}\left(\tilde{\mathbf{Y}}_{k(l)}^j - \tilde{\mathbf{Y}}_{k(l)}^i\right),\\ \tilde{\mathbf{y}}_{k(l+\gamma)}^i &= \tilde{\mathbf{y}}_{k(l)}^i + \epsilon \sum_{\forall C^j \in \mathcal{N}^i}\left(\tilde{\mathbf{y}}_{k(l)}^j - \tilde{\mathbf{y}}_{k(l)}^i\right).\end{aligned} \tag{16}$$

Here, $\epsilon \in \left(0, \frac{1}{\Delta_{max}}\right)$, where $\Delta_{max} = \max_{\forall C^i \in \mathbf{C}}\{|\mathcal{N}^i|\}$.

The *correction* step replaces the initial predicted local information pair $\left(\tilde{\mathbf{y}}_{k(\tau_k^i)}^i, \tilde{\mathbf{Y}}_{k(\tau_k^i)}^i\right)$ with the actual local information pair $\left(\mathbf{y}_k^i, \mathbf{Y}_k^i\right)$ as

$$\begin{aligned}\hat{\mathbf{Y}}_k^i &= \tilde{\mathbf{Y}}_{k(l+\gamma)}^i - \frac{\tilde{\mathbf{Y}}_{k(\tau_k^i)}^i}{N} + \frac{\mathbf{Y}_k^i}{N},\\ \hat{\mathbf{y}}_k^i &= \tilde{\mathbf{y}}_{k(l+\gamma)}^i - \frac{\tilde{\mathbf{y}}_{k(\tau_k^i)}^i}{N} + \frac{\mathbf{y}_k^i}{N}.\end{aligned} \tag{17}$$

The state estimate $\hat{\boldsymbol{\mu}}_k^i$ and the corresponding error covariance $\hat{\boldsymbol{\Sigma}}_k^i$ are

$$\hat{\boldsymbol{\mu}}_k^i = {\hat{\mathbf{Y}}_k^i}^{-1}\hat{\mathbf{y}}_k^i, \text{ and } \hat{\boldsymbol{\Sigma}}_k^i = {\hat{\mathbf{Y}}_k^i}^{-1}. \tag{18}$$

If the time elapsed since the capturing instant k is larger than $T-\alpha$, then C^i terminates its fusion phase. Otherwise, the same process, (14)-(17), is repeated using $\left(\tilde{\mathbf{y}}_{k(l+\gamma)}^i, \tilde{\mathbf{Y}}_{k(l+\gamma)}^i\right)$ as input, i.e. $l \leftarrow l+\gamma$.

Note that when $\alpha = 0$ (synchronous case), ICF and ACAF yield the same result but differs in the type of information exchanged: ACAF exchanges predicted information corresponding to $k+l$, whereas ICF exchanges the actual information corresponding to k.

5. RESULTS

We compare the performance of ACAF, the proposed filter, with (i) ICF [17], which uses average consensus assuming synchronous setting; (ii) aCDTT [16], which uses maximum consensus in asynchronous settings; (iii) the distributed filter that computes the local state estimates but never performs fusion (No fusion); and (iv) a centralised filter (CEN) that assumes the fusion centre is aware of the capturing instants, i.e. the delays are known.

ACAF requires fewer scalar transmissions than ICF and aCDTT. In particular, ACAF has fewer transmissions than ICF because of the early termination of the consensus phase. In aCDTT, each consensus iteration exchanges the instantaneous local estimate, the index of the camera that generated the estimate and the label to distinguish information from subsequent estimation phases. In contrast, only the local estimates are exchanged in ACAF and ICF.

We use simulated and real trajectories for a WCN that monitors a 30m $\times$ 20m area using $N = 7$ static cameras whose positions and FoVs are taken from the APIDIS dataset[1]. The validation is conducted for full (Fig. 2a) and sparse connectivity (Fig. 2d), for full observability of the cameras (Fig. 2b) and limited observability (Fig. 2e), without estimation delays ($\tau_k^i = 0,\ \forall C^i \in \mathbf{C}$) and with random estimation delays ($\tau_k^i \in \{0,1,2,3\},\ \forall C^i \in \mathbf{C}$), and with known and unknown motion models. Here, $\gamma = 1$ time step, $T = 25$ time steps and one time step ≈40ms. To consider trajectories with known motion model, we generate $N_p = 20$ simulated trajectories with a known motion model (Fig. 2c) each 300 time-step long. The considered motion model is the nearly constant velocity

[1] http://sites.uclouvain.be/ispgroup/index.php/Softwares/APIDIS

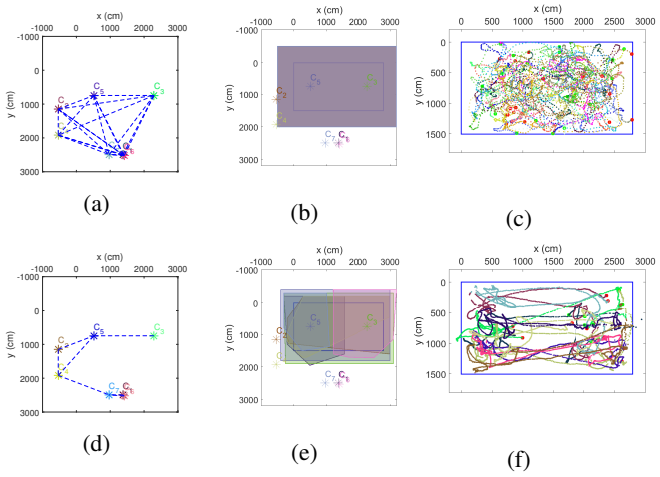

Fig. 2: Experimental setup. (a) full connectivity, (b) full observability, (c) simulated tracks, (d) sparse connectivity, (e) limited observability (APIDIS), (f) real tracks (APIDIS)

model defined by the state transition function $\mathbf{F}(k, k+\Delta k)$ and the process noise covariance matrix $\mathbf{Q}(k, k+\Delta k)$ as in [1, 2, 4]. To consider trajectories with unknown motion model, we use trajectories of $N_p = 10$ players given in the APIDIS dataset each 1500 time-step long (Fig. 2f). The measurement model of each camera C^i is

$$\mathbf{z}_k^i = [\mathbf{I}_2 \ \mathbf{0}_2]\mathbf{x}_{t_k^i} + \mathbf{v}_k^i, \tag{19}$$

where $\mathbf{I}_2$ and $\mathbf{0}_2$ are the 2×2 unit and zero matrices respectively. $\mathbf{v}_k^i$ is the measurement noise vector of C^i at k, which is assumed to be Gaussian with mean $\mathbf{0}_2$ and covariance matrix $\mathbf{R}^i = 60\mathbf{I}_2$. We analyse the mean tracking error with increasing asynchronism α. To let each camera complete its estimation phase, α should be $\leq T - \tau^{max}$, with $\tau^{max} = \max_{\forall C^i \in \mathbf{C}, \forall k} \{\tau_k^i\} = 3$. If D is the network diameter, aCDTT requires at least D consensus iterations so α should be $\leq T - \tau^{max} - D$. For the sparse connectivity (Fig. 2d), $D = 4$ so we choose $\alpha \in [0, 18]$. We track each player $p \in [1, N_p]$ separately using $M = 10$ Monte-Carlo simulations. Each simulation uses a different set of estimation delays and measurements. The mean tracking error, defined as the mean of the N_p root mean square errors, is considered as the performance measure.

The tracking error increases as the asynchronism increases irrespective of the delays, observability and connectivity (Fig. 3). In the synchronous case ($\alpha = 0$), the accuracy of ACAF is equivalent to ICF irrespective of the delays, observability and connectivity. In the asynchronous case ($\alpha > 0$), ACAF achieves better tracking accuracy than aCDTT and ICF irrespective of the delays, observability and connectivity. The tracking error of ACAF is upper bounded by the tracking error of the distributed filtering that does not perform fusion. This is because ICF fuses the information without time alignment. Moreover, there is a risk of fusing the information corresponding to subsequent frames. aCDTT does not perform fusion at all. In addition, aCDTT assigns a local estimate corresponding to a time instant to different other time instants. In the case of full observability, it is better to avoid fusion instead of using ICF and aCDTT irrespective of the delays and connectivity (Fig. 3a- 3d, 3i- 3l). This is because ICF fuses asynchronous information without time alignment and aCDTT assigns highly certain information all the times. Both worsen the accuracy. In the case of limited observability, nodes that cannot view the target predict the target information. If there is no fusion, the nodes cannot correct their predicted estimates and result in maximum tracking error irrespective of delays and connectivity (Fig. 3e- 3h, 3m- 3p). When asynchronism is high, the tracking error of ACAF increases significantly. This is because the higher the asynchronism, the lower the duration of the fusion phase, thus leading to an insufficient number of consensus iterations for convergence.

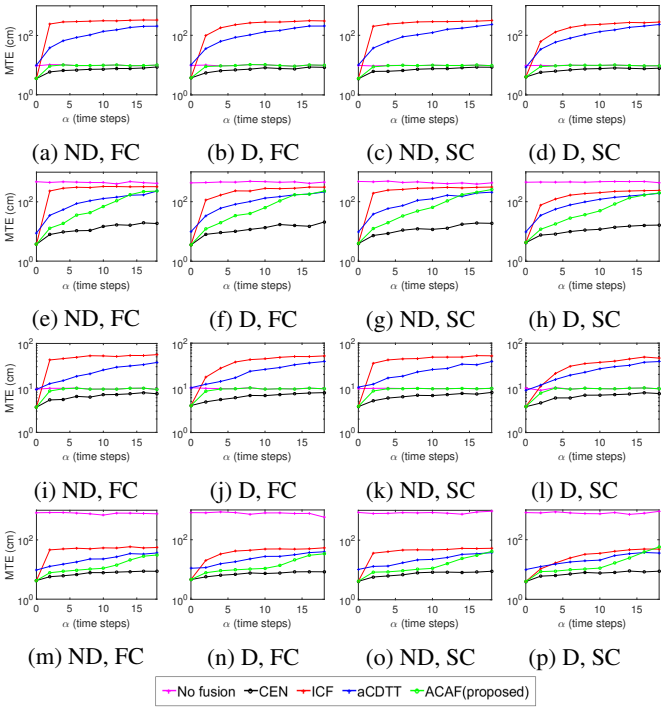

Fig. 3: Mean tracking error (MTE) with increasing asynchronism α. (a)-(h) Results with simulated tracks. (i)-(p) Results with APIDIS tracks. (a)-(d) and (i)-(l) Results with full observability. (e)-(h) and (m)-(p) Results with limited observability. KEY – D: delay, ND: no delay, FC: full connectivity, SC: sparse connectivity. The compared algorithms are the distributed filter that does not fuse (No fusion), the centralised filter (CEN), the Information Consensus Filter (ICF) [17], the asynchronous Consensus-based Distributed Target Tracking method (aCDTT) [16] and the proposed Average Consensus-based Asynchronous Filter (ACAF).

6. CONCLUSION

We proposed an Average Consensus-based Asynchronous tracking Filter which can deal with asynchronous capture and delayed processing that are typical in wireless camera networks. We time-align the data via predictions before fusion using the known states corresponding to the reception instants. Each camera predicts the target information of other cameras at its capturing instant. The proposed method achieves better tracking accuracy and uses less communication bandwidth than state-of-the-art methods in the asynchronous case.

In our future work we will first model false positive measurements and packet losses, and then multiple simultaneous targets.

7. REFERENCES

[1] Á. F. García-Fernández and J. Grajal, "Asynchronous particle filter for tracking using non-synchronous sensor networks," *Signal Processing*, vol. 91, no. 10, pp. 2304–2313, Oct 2011.

[2] J. Beaudeau, M. F. Bugallo, and P. M. Djurić, "Target tracking with asynchronous measurements by a network of distributed mobile agents," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Mar 2012, pp. 3857–3860.

[3] G. Zhu, F. Zhou, L. Xie, R. Jiang, and Y. Chen, "Sequential asynchronous filters for target tracking in wireless sensor networks," *IEEE Sensors Jour.*, vol. 14, no. 9, pp. 3174–3182, Sep 2014.

[4] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Distributed sequential estimation in asynchronous wireless sensor networks," *IEEE Signal Processing Let.*, vol. 22, no. 11, pp. 1965–1969, Nov 2015.

[5] S. Katragadda, J. C. SanMiguel, and A. Cavallaro, "The costs of fusion in smart camera networks," in *Proc. of the Int. Conf. on Distributed Smart Cameras*, Nov 2014.

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

[7] A. T. Kamal, J. A. Farrell, and A. K. 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, Dec 2013.

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

[9] S. Katragadda, J. C. SanMiguel, and A. Cavallaro, "Consensus protocols for distributed tracking in wireless camera networks," in *Proc. of the Int. Conf. on Information Fusion*, Jul 2014, pp. 1–8.

[10] S. Katragadda and A. Cavallaro, "Neighbour consensus for distributed visual tracking," in *Proc. of the IEEE Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing*, Apr 2015, pp. 1–6.

[11] Y. Wang and A. Cavallaro, "Prioritized target tracking with active collaborative cameras," in *Proc. of the IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Aug 2016, pp. 131–137.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. of the Symp. on Operating Systems Design and Implementation*, Dec 2002, vol. 36, pp. 147–163.

[13] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, Jul 2004.

[14] M. Vemula, J. Miguez, and A. Artes-Rodriguez, "A sequential monte carlo method for target tracking in an asynchronous wireless sensor network," in *Proc. of the Workshop on Positioning, Navigation and Communication*, Mar 2007, pp. 49–54.

[15] J. C. SanMiguel and A. Cavallaro, "Energy consumption models for smart-camera networks," *IEEE Trans. on Circuits and Systems for Video Technology*, DOI: 10.1109/TCSVT.2016.2593598.

[16] S. Giannini, A. Petitti, D. Di Paola, and A. Rizzo, "Asynchronous consensus-based distributed target tracking," in *Proc. of the IEEE Conf. on Decision and Control*, Dec 2013, pp. 2006–2011.

[17] 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, Mar 2014.

[18] L. Fang and P. J. Antsaklis, "Information consensus of asynchronous discrete-time multi-agent systems," in *Proc. of the American Control Conf.*, Jun 2005, pp. 1883–1888.

[19] M. Mehyar, D. Spanos, J. Pongsajapan, S. H. Low, and R. M. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Trans. on Networking*, vol. 15, no. 3, pp. 512–520, Jun 2007.

[20] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *Proc. of the IEEE Int. Symp. on Information Theory*, Jun 2010, pp. 1753–1757.

[21] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Asynchronous newton-raphson consensus for distributed convex optimization," in *Proc. of the IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Sep 2012, vol. 45, pp. 133 – 138.

[22] A. Carron, M. Todescato, R. Carli, and L. Schenato, "An asynchronous consensus-based algorithm for estimation from noisy relative measurements," *IEEE Trans. on Control of Network Systems*, vol. 1, no. 3, pp. 283–295, Sep 2014.

[23] R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Distributed quadratic programming under asynchronous and lossy communications via newton-raphson consensus," in *Proc. of the European Control Conf.*, Jul 2015, pp. 2514–2520.

[24] M. Kriegleder, R. Oung, and R. D'Andrea, "Asynchronous implementation of a distributed average consensus algorithm," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nov 2013, pp. 1836–1841.

[25] D. W. Casbeer and R. Beard, "Distributed information filtering using consensus filters," in *Proc. of the American Control Conf.*, Jun 2009, pp. 1882–1887.