Distributed Sigma Point Information Filters for Target Tracking in Camera Networks

Shiva Kumar K A
Electrical Engineering Department
Indian Institute of Science
Bangalore, India
e-mail: shivakumar.acs@gmail.com

K. R. Ramakrishnan
Electrical Engineering Department
Indian Institute of Science
Bangalore, India
e-mail: krr@ee.iisc.ernet.in

G. N. Rathna
Electrical Engineering Department
Indian Institute of Science
Bangalore, India
e-mail: rathna@ee.iisc.ernet.in

Abstract

Distributed processing is a new paradigm to analyse the huge volume of video data in camera networks. This paper addresses the problem of distributed single target tracking considering false positives and missed detections. Target tracking is modelled as a dynamic state estimation problem with nonlinear process and measurement model. We propose to use the sigma point information filters combined with a consensus algorithm. Sigma point information filters are integrated with probabilistic data association filter to deal with false positives and missed detections. We use a distributed average consensus algorithm which converges in finite time. Unlike other related state of the art technique papers, we report results on real data and show the effectiveness of the proposed algorithm.

1 Introduction

Increasing applications of camera networks pose new challenges to the computer vision community. The applications include surveillance, monitoring the elderly people in smart homes and military applications. The challenges include developing new algorithms which make use of full potential of the installed camera networks. The algorithms should be simple, real time and robust to camera failures.

Distributed video processing is gaining lot of importance. Distributed processing has the following characteristics: i) No central server. ii) Cameras carry out local processing and communicate with immediate neighbours iteratively to improve the results. iii) The results at each camera should be equal to that of centralized method. Consensus based distributed processing methods are very popular because of their simplicity, robustness and scalability. Specifically, distributed average algorithms are core algorithms for many distributed processing. Our work is also based on distributed averaging algorithm.

Cameras are directional sensors and have limited field of view (FoV). In a general camera network a target is covered by few cameras only. This causes the camera networks to face a new challenge known as naivety [1]. The camera which does not has information about the target either directly or from neighbours is referred as naive camera. These naive cameras can effect the final result in the distributed processing. Hence naivety is one of the key issue while addressing the distributed target tracking in camera networks.

In [3], authors integrated extended Kalman consensus filter (EKCF) with joint probabilistic data association filter (JPDA) and developed JPDA-EKCF. But EKCF does not work well [1] for camera networks. In [2], the author developed extended multi target information consensus filter (EMTIC) by extending information consensus filter (ICF) to account for nonlinear measurement model and by integrating with JPDA to account for measurement uncertainty. Another extension of ICF was developed in [4], to account for nonlinear process and measurement model but in this paper the authors have not considered false positives and missed detections. JPDA-EKCF and EMTIC both are based on extended Kalman filter principles. But in this paper, we use a consensus based method for distributed target tracking using the principles of sigma point information filters and PDA filter. In existing consensus based tracking methods, the authors have used a distributed average algorithm which converges asymptotically. But we use a distributed average algorithm which converges in finite time, so that we can obtain the exact convergence to the results of centralized method. It also reduces the communication and computation cost. Above mentioned state of the art consensus based papers reported results on simulation data only. In this paper we report results on real world and simulation data.

2 Problem Statement

Consider a set of networked cameras $\mathcal{C} = \{C_1, C_2, \dots, C_{N_c}\}$, ($N_c$ is number of cameras) with overlapping FoV. We assume the communication between cameras is noise free and the communication graph is time invariant. In this paper we address the single target tracking problem considering false positives and missed detections. The target movement is modelled by

$$\mathbf{x}(k+1) = f(\mathbf{x}(k)) + \mathbf{G}\boldsymbol{\eta}(k) \quad (1)$$

where k denotes time index, $\mathbf{x}(\cdot) \in \mathbb{R}^n$ a state vector, $f(\cdot)$ is a nonlinear function, $\mathbf{G}$ is a coefficient matrix, $\boldsymbol{\eta}(\cdot)$ is white Gaussian noise with covariance $\mathbf{Q}$. The measurement model for the camera i is given by

$$\tilde{\mathbf{z}}_i(k+1) = h_i(\mathbf{x}(k+1)) + \mathbf{v}_i(k+1), i = 1, 2, .., N_c \quad (2)$$

where $\tilde{\mathbf{z}}_i \in \mathbb{R}^p$ is a measurement vector, $h_i(\cdot)$ is a nonlinear function, $\mathbf{v}_i(\cdot)$ is the measurement noise (white Gaussian with covariance $\mathbf{R}_i \in \mathbb{R}^{p \times p}$). $\mathbf{v}_i$ and $\boldsymbol{\eta}$, as well as measurement noise across cameras are assumed to be uncorrelated. Our proposed solution estimates the true states $\mathbf{x}(\cdot)$ of the target in a distributed manner given the observations $\tilde{\mathbf{z}}_i(\cdot)$ from cameras. We hope that context makes it clear some abuse of notations in our work.

3 Sigma Point Information Filters

In this section, we review sigma point information filters [5] and develop the proposed distributed sigma point information filter. Sigma point filter (SPF) is an approximate estimation technique used in nonlinear estimation. SPF uses standard Kalman filter structure which has two stages, predict and update. During these stages, Kalman filter requires only mean and covariance of the posterior distribution. In general, for nonlinear systems these moments may be difficult to obtain. But SPF uses sigma point transform (SPT) to obtain the required moments.

3.1 Sigma Point Transform

Let $\mathbf{y} = g(\mathbf{x})$, $g(\cdot)$ is a nonlinear function, $\mathbf{x}$ is a random vector with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ and we seek the mean and covariance of $\mathbf{y}$. The working principle of SPT is as follows: A set of samples known as sigma points are generated deterministically from the given prior random vector $\mathbf{x}$. One choice to generate the sigma points $\boldsymbol{\chi}_m$ is according to following equations

$$\begin{aligned}\boldsymbol{\chi}_0 &= \hat{\mathbf{x}} \\ \boldsymbol{\chi}_m &= \hat{\mathbf{x}} + (\sqrt{(n+\lambda)\mathbf{P}_{xx}})_m, m = 1, 2, \ldots, n \\ \boldsymbol{\chi}_{m+n} &= \hat{\mathbf{x}} - (\sqrt{(n+\lambda)\mathbf{P}_{xx}})_m, m = 1, 2, \ldots, n\end{aligned} \quad (3)$$

where n is dimension of $\mathbf{x}$, $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter with α a constant parameter (usually set to a small value) and κ a scaling parameter. The term $(\sqrt{(n+\lambda)\mathbf{P}_{xx}})_m$ is the m^{th} row or column of the matrix square root of $(n+\lambda)\mathbf{P}_{xx}$. The corresponding weights for the mean w_m^{mean} and covariance w_m^{cov} are given by

$$\begin{aligned} w_0^{mean} &= \frac{\lambda}{n+\lambda}; w_0^{cov} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \xi) \\ w_m^{cov} &= w_m^{mean} = \frac{1}{2(n+\lambda)}, m = 1, 2, \ldots, 2n \end{aligned} \quad (4)$$

where ξ is a parameter used to incorporate the prior knowledge of the distribution.
The generated sample points are propagated through the nonlinear equation

$$\mathcal{Y}_m = g(\boldsymbol{\chi}_m), m = 0, 1, \ldots, 2n \quad (5)$$

The mean and covariance matrix of the posterior random vector $\mathbf{y}$ are given by

$$\hat{\mathbf{y}} = \sum_{m=0}^{2n} w_m^{mean} \mathcal{Y}_m \quad (6)$$

$$\mathbf{P}_{yy} = \sum_{m=0}^{2n} w_m^{cov} [\mathcal{Y}_m - \hat{\mathbf{y}}][\mathcal{Y}_m - \hat{\mathbf{y}}]^T \quad (7)$$

Sigma point filters can also be interpreted as statistical linearization of $g(\mathbf{x})$. Here we seek $\mathbf{A}$ and $\mathbf{c}$ such that $\mathbf{y} = g(\mathbf{x}) \approx \mathbf{A}\mathbf{x} + \mathbf{c}$. We obtain $\mathbf{A}$ and $\mathbf{c}$ by minimizing the sum of squared errors $\mathbf{e}_m = \mathcal{Y}_m - (\mathbf{A}\boldsymbol{\chi}_m + \mathbf{c})$ i.e.,

$$(\mathbf{A}, \mathbf{c}) = \underset{\mathbf{A},\mathbf{c}}{\operatorname{argmin}} \sum_{m=0}^{2n} \mathbf{e}_m^T \mathbf{e}_m \quad (8)$$

Above equation has the following solution:

$$\mathbf{A} = \mathbf{P}_{xy}^T \mathbf{P}_{xx}^{-1}; \qquad \mathbf{c} = \hat{\mathbf{y}} - \mathbf{A}\hat{\mathbf{x}} \quad (9)$$

$\mathbf{P}_{xy}$ can be calculated by

$$\mathbf{P}_{xy} = \sum_{m=0}^{2n} w_m^{cov} [\boldsymbol{\chi}_m - \hat{\mathbf{x}}][\mathcal{Y}_m - \hat{\mathbf{y}}]^T \quad (10)$$

3.2 Information Form Filters

Information filter is algebraically equivalent to standard Kalman filter but is more suitable for distributed implementation. Because the equations in information filter are easy to decouple. The details of the following "information form" of sigma point filters can be found in [5]. Define information matrix $\mathbf{Y}_{xx}(k/k)$ and information vector $\hat{\mathbf{y}}(k/k)$ as

$$\mathbf{Y}_{xx}(k/k) = \mathbf{P}_{xx}^{-1}(k/k) \quad (11)$$

$$\hat{\mathbf{y}}(k/k) = \mathbf{Y}_{xx}(k/k)\hat{\mathbf{x}}(k/k) \quad (12)$$

For linearized equations (1) and (2), the standard Kalman filter equations can be expressed in information form as below. The prediction step equations are

$$\begin{aligned}\hat{\mathbf{y}}(k/k-1) &= \mathbf{Y}_{xx}(k/k-1) \\ &[\bar{\mathbf{F}}(k)\mathbf{Y}_{xx}^{-1}(k-1/k-1)\hat{\mathbf{y}}(k-1/k-1) + \mathbf{c}^x(k)]\end{aligned} \quad (13)$$

$$\mathbf{Y}_{xx}(k/k-1) = [\bar{\mathbf{F}}(k)\mathbf{Y}_{xx}^{-1}(k-1/k-1)\bar{\mathbf{F}}^T(k) + \bar{\mathbf{Q}}(k)]^{-1} \quad (14)$$

where $\bar{\mathbf{F}}(k)$ and $\mathbf{c}^x(k)$ are obtained as discussed in section 3.1.

$$\bar{\mathbf{Q}}(k) = \mathbf{P}_{xx}(k/k-1) - \bar{\mathbf{F}}(k)\mathbf{P}_{xx}(k-1/k-1)\bar{\mathbf{F}}^T(k) \quad (15)$$

Under the assumption of uncorrelated noise across cameras, the update step equations are given by

$$\hat{\mathbf{y}}(k/k) = \hat{\mathbf{y}}(k/k-1) + \sum_{i=1}^{N_c} \mathbf{i}_i(k) \quad (16)$$

$$\mathbf{Y}_{xx}(k/k) = \mathbf{Y}_{xx}(k/k-1) + \sum_{i=1}^{N_c} \mathbf{I}_i(k) \quad (17)$$

$$\mathbf{i}_i(k) = \bar{\mathbf{H}}_i^T(k)\bar{\mathbf{R}}_i^{-1}(k)[\tilde{\mathbf{z}}_i(k) - \mathbf{c}_i^{\tilde{z}}(k)] \quad (18)$$

$$\mathbf{I}_i(k) = \bar{\mathbf{H}}_i^T(k)\bar{\mathbf{R}}_i^{-1}(k)\bar{\mathbf{H}}_i(k) \quad (19)$$

$$\bar{\mathbf{R}}_i(k) = \mathbf{P}_{\tilde{z}_i\tilde{z}_i}(k/k) - \bar{\mathbf{H}}_i(k)\mathbf{P}_{xx}(k/k-1)\bar{\mathbf{H}}_i^T(k) \quad (20)$$

where $\bar{\mathbf{H}}_i(k)$, $\mathbf{c}_i^{\tilde{z}}(k)$ are obtained as discussed in section 3.1. To deal with missed detections and false positives, sigma point filters are integrated with probabilistic data association (PDA) filter.

To derive information form equations of sigma point filters with PDA, first we note that the prediction step equations do not change. We use measurements during update step for obtaining the information contribution vector $\mathbf{i}_i(k)$ and matrix $\mathbf{I}_i(k)$. Hence we need to define these vector and matrix. This can be done by first writing equations of sigma point filters with PDA in standard Kalman filter form and then deriving equivalent information form equations. See [5] for details.

The information contribution vector and matrix for sigma point filters with PDA are given by

$$\mathbf{i}_i(k) = \bar{\mathbf{H}}_i^T(k)\tilde{\mathbf{R}}_i^{-1}(k)\zeta_i(k) \quad (21)$$

$$\mathbf{I}_i(k) = \bar{\mathbf{H}}_i^T(k)\tilde{\mathbf{R}}_i^{-1}(k)\bar{\mathbf{H}}_i(k) \quad (22)$$

$$\tilde{\mathbf{R}}_i(k) = \bar{\mathbf{R}}_i(k) + [\mathbf{B}_i^{-1}(k) - \bar{\mathbf{S}}_i^{-1}(k)]^{-1} \quad (23)$$

$$\zeta_i(k) = [\mathbf{Iden} + (\mathbf{B}_i^{-1}(k) - \bar{\mathbf{S}}_i^{-1}(k))^{-1}\bar{\mathbf{S}}_i^{-1}(k)]\boldsymbol{\nu}_i(k) + \bar{\mathbf{H}}_i(k)\hat{\mathbf{x}}(k/k-1) \quad (24)$$

$$\bar{\mathbf{S}}_i(k) = \bar{\mathbf{R}}_i(k) + \bar{\mathbf{H}}_i(k)\mathbf{Y}_{xx}^{-1}(k/k-1)\bar{\mathbf{H}}_i^T(k) \quad (25)$$

$$\boldsymbol{\nu}_i(k) = \sum_{t=1}^{l_i(k)} \beta_i^t \boldsymbol{\nu}_i^t(k) \quad (26)$$

$$\boldsymbol{\nu}_i^t(k) = \tilde{\mathbf{z}}_i^t(k) - h_i(\mathbf{x}(k/k-1)) \quad (27)$$

$$\mathbf{B}_i(k) = \beta_i^0\bar{\mathbf{S}}_i(k) + \sum_{t=1}^{l_i(k)} \beta_i^t\boldsymbol{\nu}_i^t(k)\boldsymbol{\nu}_i^{tT}(k) - \boldsymbol{\nu}_i(k)\boldsymbol{\nu}_i^T(k) \quad (28)$$

β_i^t, $t = 1, 2, .., l_i(k)$, ($l_i(k)$ is number of measurements in camera i at time k) are association probabilities used in PDA. **Iden** denotes identity matrix.

3.3 Proposed Method

From the equations (16) and (17), we note that in both equations second term is the only term which requires information from all cameras. We also note that the second term is expressed as sum of the information contribution vector $\mathbf{i}_i(k)$ and matrix $\mathbf{I}_i(k)$ from individual cameras. By denoting the $\mathbf{i}_{avg}(k)$ as average of $\mathbf{i}_1(k), \mathbf{i}_2(k), ..., \mathbf{i}_{N_c}(k)$ and similarly for $\mathbf{I}_{avg}(k)$, the update equations are given by

$$\hat{\mathbf{y}}^{cent}(k/k) = \hat{\mathbf{y}}^{cent}(k/k-1) + N_c\mathbf{i}_{avg}(k) \quad (29)$$

$$\mathbf{Y}_{xx}^{cent}(k/k) = \mathbf{Y}_{xx}^{cent}(k/k-1) + N_c\mathbf{I}_{avg}(k) \quad (30)$$

The superscript *cent* is introduced to emphasize that the above equations are "centralized equations". From (29) and (30), we note that the problem of centralized estimation boils down to an averaging problem. To obtain average of vectors and matrices in distributed manner, we apply a distributed average algorithm [7] on each component of the vector and the matrix independently. This algorithm converges to true average in finite time. Thus we implement centralized target tracking problem in distributed manner.

Algorithm 1 gives the complete procedure for proposed single target distributed sigma point information filter (STDSPIF). We run this algorithm at each camera and at every time instant. We initialize all cameras with the same prior values. Since the distributed average algorithm converges to exact average in finite time, the convergence of the proposed algorithm to centralized algorithm is ensured.

Table 1: Dataset information: T is number of time instants.

Dataset	N_c	T	Scene Type
EPFL Terrace1 [10]	4	275	Outdoor
EPFL Lab6 [10]	4	121	Indoor
TeV MVPDT [11]	4	301	Indoor
UCSB Hallway [12]	5	201	Indoor
Simulation [4]	9	40	–

Algorithm 1 *STDSPIF: At camera i at time k (Note: we are dropping i for clarity)*

Input: $\hat{\mathbf{x}}(k/k-1), \mathbf{P}_{xx}(k/k-1), \{\tilde{\mathbf{z}}^t\}_{t=1}^{l(k)}$

Data Association:

1) Generate samples using (3) by replacing $\hat{\mathbf{x}}$ and $\mathbf{P}_{xx}$ with $\hat{\mathbf{x}}(k/k-1)$ and $\mathbf{P}_{xx}(k/k-1)$ respectively.

2) Propagate the samples through the measurement equation.

3) Obtain measurement prediction, innovation covariance and cross covariance, using (6), (7) and (10) respectively.

4) Obtain $\bar{\mathbf{H}}(k)$, $\bar{\mathbf{R}}(k)$ and $\bar{\mathbf{S}}(k)$ using (9), (20) and (25) respectively.

5) Calculate the probabilities β^t

6) Obtain $\tilde{\mathbf{R}}(k)$, $\zeta(k)$, information contribution vector $\mathbf{i}(k)$ and information contribution matrix $\mathbf{I}(k)$ using (23), (24), (21) and (22) respectively.

Consensus: *Run average consensus algorithm independently on each component of $\mathbf{I}(k)$ and $\mathbf{i}(k)$.*

Update:

$$\mathbf{Y}_{xx}(k/k) = \mathbf{Y}_{xx}(k/k-1) + N_c\mathbf{I}_{avg}(k) \quad (31)$$

$$\hat{\mathbf{y}}(k/k) = \hat{\mathbf{y}}(k/k-1) + N_c\mathbf{i}_{avg}(k) \quad (32)$$

Prediction: *1) Generate samples using (3) by replacing $\hat{\mathbf{x}}$ and $\mathbf{P}_{xx}$ with $\hat{\mathbf{x}}(k/k)$ and $\mathbf{P}_{xx}(k/k)$ respectively.*

2) Propagate the samples through the process equation.

3) Obtain state prediction, prediction covariance and cross covariance, using (6), (7) and (10) respectively.

4) Obtain $\bar{\mathbf{F}}(k+1)$, $\mathbf{c}^x(k+1)$ and $\bar{\mathbf{Q}}(k+1)$ using (9) and (15) respectively.

5) Obtain predictions using (11) ,(12) ,(13) and (14).

4 Experimental Results

We evaluated the proposed method with various publicly available multi-camera pedestrian datasets. The details are given in Table 1. For each dataset we have created ground truth for one of the camera using Video Annotation Tool VATIC [8]. We obtained target positions by running discriminatively trained part based model algorithm [9]. To realize sparse network connection we used line communication graph. To demonstrate the naivety issue, we considered only those frames in which target is detected in only one of the cameras (except for EPFL Terrace1).

State vector of the target is given by

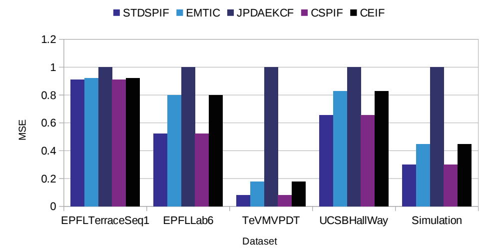

Figure 1: MSE values for various datasets

Table 2: MSE values

Dataset	STSPIF	EMTIC	JPDA EKCF	CSPIF	CEIF
EPFL Terrace1 [10]	0.9107	0.9216	1	0.9107	0.9216
EPFL Lab6 [10]	0.5217	0.8005	1	0.5217	0.8005
TeV MVPDT [11]	0.0797	0.1789	1	0.0797	0.1789
UCSB Hallway [12]	0.6540	0.8261	1	0.6540	0.8261
Simulation [4]	0.2995	0.4462	1	0.2995	0.4462

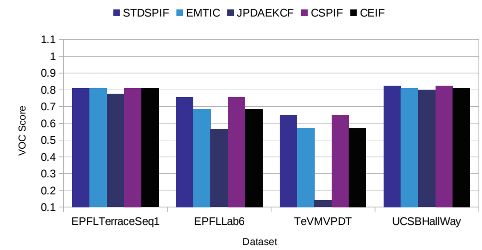

Figure 2: VOC scores for various datasets

Table 3: VOC Scores

Dataset	STSPIF	EMTIC	JPDA EKCF	CSPIF	CEIF
EPFL Terrace1 [10]	0.8093	0.8079	0.7755	0.8093	0.8079
EPFL Lab6 [10]	0.7565	0.6832	0.5661	0.7565	0.6832
TeV MVPDT [11]	0.6484	0.5709	0.1417	0.6484	0.5709
UCSB Hallway [12]	0.8247	0.8097	0.8001	0.8247	0.8097

$[x(k), y(k), \dot{x}(k), \dot{y}(k), \rho(k)]^T$. The process model is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 & \frac{sin(\rho(k)\Delta t)}{\rho(k)} & \frac{cos(\rho(k)\Delta t)-1}{\rho(k)} & 0 \\ 0 & 1 & \frac{1-cos(\rho(k)\Delta t)}{\rho(k)} & \frac{sin(\rho(k)\Delta t)}{\rho(k)} & 0 \\ 0 & 0 & cos(\rho(k)\Delta t) & -sin(\rho(k)\Delta t) & 0 \\ 0 & 0 & sin(\rho(k)\Delta t) & cos(\rho(k)\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\eta}(k) \quad (33)$$

where $(x(k), y(k))$ is the position of the target, $(\dot{x}(k), \dot{y}(k))$ is the velocity and $\rho(k)$ is the turn rate. Δt denotes the length of the time step. The noise $\boldsymbol{\eta}(k) \sim \mathcal{N}(0, \mathbf{Q}(k))$.
The measurement equation is modelled by [4]

$$\tilde{\mathbf{z}}_i(k) = \begin{bmatrix} u_i(k) \\ v_i(k) \end{bmatrix} = \begin{bmatrix} \frac{H_{11}^i x(k)+H_{12}^i y(k)+H_{13}^i}{H_{31}^i x(k)+H_{32}^i y(k)+H_{33}^i} \\ \frac{H_{21}^i x(k)+H_{22}^i y(k)+H_{23}^i}{H_{31}^i x(k)+H_{32}^i y(k)+H_{33}^i} \end{bmatrix} + \mathbf{v}_i(k) \quad (34)$$

where $(u_i(k), v_i(k))$ is the pixel coordinates of the target in the image plane of the camera i at time k. The values $H_{11}^i, \ldots H_{33}^i$ are the elements of the homography matrix and the measurement noise $\mathbf{v}_i(k) \sim \mathcal{N}(0, \mathbf{R}_i(k))$.
Following methods were implemented and compared: The proposed STDSPIF and it's centralized version (CSPIF), EMTIC [2] and it's centralized version (CEIF) and (JPDA-EKCF) [3]. EMTIC and JPDA-EKCF are modified to account the nonlinear process model. EMTIC is implemented using the finite time convergence average algorithm and JPDA-EKCF is modified to support multiple consensus iterations [2].
Since ground truth was available, most of the required parameters are estimated. The remaining parameters are choosen such that each algorithm performs reasonably well. In our experiments we found that all these algorithms diverge for inappropriate parameters.
As performance measure we report mean square error (MSE) [2] and mean VOC detection score (for real data). The error is defined as Euclidean distance between the true position and the estimated position. MSE of all algorithms are normalized with respect to JPDA-EKCF algorithm MSE. VOC detection score is ratio of intersection area of predicted bounding box and ground truth bounding box to union area of predicted bounding box and ground truth bounding box.
From Figures 1, 2 and Tables 2,3 we made following observations: i) STDSPIF and EMTIC are converged to their corresponding centralized schemes. ii) In EPFL Terrace1 sequence the target is visible in many of the cameras for most of the time and has a relatively simple motion. Hence all algorithms performed well. Even JPDA-EKCF works fine as there is no issue of naivety. iii) In EPFL Lab6 sequence, initially the target has a simple linear motion then target takes a U turn. At this point of time other algorithms deviates away from the true track but our algorithm continues to track the target. Hence the improved result. iv) TeV multi view people detection and tracking (MVPDT) sequence is a complex dataset. Although there is no naivety issue here, this dataset has high rate of false positives and more nonlinearity. For this dataset the homography matrices were not available hence they are estimated. In this case the performance of all algorithms degraded but our algorithm works well compared to other algorithms. v) In UCSB Hallway sequence camera 1 is naive about the target for most of the time. It is interesting to observe that MSE value of JPDA-EKCF is high yet the VOC score is good enough to compete with other algorithms. This is because the target is visible almost all the time in camera 5. Coincidently Camera 5 is used to calculate the VOC score. We calculated MSE over all cameras but VOC score is cal-

culated for only one camera (for which ground truth is available). vi) When target motion is almost linear, all algorithms work equally well. But our algorithm works better when there is more nonlinearity in the model. In some cases no algorithms work good. We have uploaded a supplementary video showing the performance of different algorithms on these datasets.
We performed simulation experiments also to further validate our results. We modified the source code provided by [4] to incorporate our process model and to consider the false positives and missed detections. In this experiment we used ring topology. Following parameters are used in this experiment: $\mathbf{Q} = diag([2, 2, 0.01])$, $\mathbf{R} = diag([15, 15])$. False measurements were generated at each node at each measurement step using a Poisson process with $\lambda = 1/5$. For the fixed parameters the experiment was run for 100 different simulations, each time generating new track. Few times some of the algorithms were diverged, we ignored these cases. The reported MSE values are average of successfully run experiments. In this experiment also, our proposed algorithm works better than other algorithms.

5 Conclusions

In this paper we have proposed a consensus based sigma point information filter (STDSPIF) for distributed single target tracking in camera networks. We have used average consensus algorithm which converges in finite time. We performed experiments on real world and simulation data. The experimental results show that the STDSPIF method performs better than JPDA-EKCF and EMTIC. The performance of our algorithm is better when there is more nonlinearity in the model. We are working on extending the proposed algorithm to multiple targets and to improve its stability.

References

[1] Kamal A T et al.: "A generalized Kalman consensus filter for wide-area video networks," *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp.7863–7869, 2011.

[2] Kamal A T: "Information Weighted Consensus for Distributed Estimation in Vision Networks", 2013.

[3] Roy-Chowdhury,A K et al:"Camera networks: the acquisition and analysis of videos over wide areas", *Synt Lect on Computer Vision*, vol.3, no.1, pp.1–133, 2012.

[4] Katragadda, S, et al.: "Consensus protocols for distributed tracking in wireless camera networks," *17th International Conference on FUSION*, pp.1–8, 2014.

[5] Vercauteren, et al.: "Decentralized sigma-point information filters for target tracking in collaborative sensor networks," *IEEE Transactions on Signal Processing*, vol.53, no.8, pp.2997–3009, 2005.

[6] Bar-Shalom et al.:"The probabilistic data association filter," *IEEE Control Systems*, vol.29, no.6, pp.82–100, 2009.

[7] Kibangou, A Y et al.: "Finite-time average consensus based protocol for distributed estimation over awgn channels," *50th IEEE CDC-ECC*, pp.5595–5600, 2011.

[8] Vondrick, et al.:"Efficiently Scaling up Crowdsourced Video Annotation," *International Journal of Computer Vision*,vol.101, no.1 pp.184–204, 2013.

[9] Felzenszwalb et al.:"Object Detection with Discriminatively Trained Part Based Models," *IEEE Transactions on PAMI*, vol.32, no.9, pp.1627–1645, 2010.

[10] Berclaz, J., et al.:"Multiple Object Tracking Using K-Shortest Paths Optimization," *IEEE Transactions on PAMI*, vol.33, no.9, pp.1806–1819, 2011.

[11] Hu, et al.:"Multicamera People Tracking Using a Locus-based Probabilistic Occupancy Map," *ICIAP*, vol.8157, pp.693–702, 2013.

[12] Ni, Zefeng , et al.:"Distributed particle filter tracking with online multiple instance learning in a camera sensor network," *17th IEEE ICIP*, pp.37–40, 2010.