Distributed Object Tracking based on Square Root Cubature H-infinity Information Filter

Venkata Pathuri Bhuvana*†, Mario Huemer‡ and Carlo S. Regazzoni†
*Alpen Adria University, Institute of Networked and Embedded Systems, Klagenfurt, Austria.
†University of Genova, Department of Electrical, Electronics, and Telecommunications Engineering and Naval Architecture.
‡Johannes Kepler University, Institute of Signal Processing, Linz, Austria
venkata.pathuri@aau.at

***Abstract*—Several non-linear state estimation methods such as extended Kalman filter, cubature Kalman filter, and unscented Kalman filter are used to track objects in visual sensor networks. These conventional non-linear state estimation methods require the accurate knowledge of the object's initial conditions, process and measurement models, and corresponding noise characteristics. Often, the object trackers used in a visual sensor networks may not be provided with this knowledge. In this work, we propose a square root cubature H_∞ information Kalman filter (SCHIF) based distributed object tracking algorithm. The H_∞ method requires neither the exact knowledge of noise characteristic nor accurate process model. The information filters can be used without the knowledge of accurate initial conditions and it also makes the measurement update step computationally less complex in the distributed process. Finally, the square root version makes the filter numerically stable. Furthermore, the cameras in the network exchange their local estimates with other cameras. In the last step, the cameras fuse the received local estimates to obtain a global estimate of the object. Hence, the proposed method constitutes a more robust and efficient solution for the targeted application compared to the traditional methods.**

I. Introduction and Motivation

Object state estimation is an extensively studied topic in the field of object tracking in visual sensor networks (VSN). A VSN is a camera network of spatially distributed smart cameras with or without a central processing unit [1]. The individual cameras autonomously retrieve, process, and analyze the data locally. In addition, the cameras can communicate with each other and work cooperatively to achieve a decision within the network, e.g. on the object's state. The distributed tracking approaches make use of the diversity of the observations from different cameras in the network to improve the tracking efficiency.

Several distributed tracking methods are presented in the literature. In [2], the tracking is done locally at each camera and the locally estimated states are then aggregated into a joint state through a consensus algorithm. In [3], authors integrated the consensus algorithm to a distributed Kalman filtering approach, forming the Kalman consensus filter for limited range camera networks. The extended Kalman filter (EKF) based object tracking is presented in [4]. The authors in [5] presented the information weighted consensus based distributed object tracking. In [6], a distributed object tracking based on the cubature Kalman filter (CKF) is presented. There, it is shown that the CKF is an appealing choice for object tracking compared to the EKF and unscented Kalman filter (UKF). However, the state estimation methods used in the above-mentioned tracking algorithms require accurate knowledge of the object's process model, initial conditions, and noise characteristics. Any uncertainty leads to unstable and inefficient state estimation, and subsequently, bad tracking results.

Recently, researchers have developed intuitive and robust extensions to the conventional state estimation methods in order to encounter the above-mentioned problems. For example, the H_∞ [7] method minimizes the estimation error in the worst case noise conditions and initialization error. It requires neither the exact noise characteristics nor the exact process model. The information filters [7] propagate the information vector and information matrix instead of the corresponding state vector and error covariance. They can be used without having the accurate knowledge of initial conditions and also make the update step computationally less complex. Hence, the information filters are preferable choice for distributed implementations in multi-sensor networks such as VSN.

The authors in [8] presented the multi-sensor cubature H_∞ information filter (CHIF). They have shown that this approach improves the tracking efficiency compared to the extended H_∞ information filter (EIF) and unscented H_∞ information filter (UIF). However, this algorithm requires the preservation of positive definitiveness and symmetry of the information matrix in each iteration. The CHIF involves numerical operations such as matrix inverse and square rooting which can destroy the above-mentioned properties of the information matrix. In this paper, we propose a square root version of the CHIF (SCHIF) which improves the numerical stability of the algorithm by preserving the fundamental properties of the information matrix. Hence, the SCHIF provides not only better accuracy but also the improved robustness compared to the traditional methods such as the CKF.

The objective of this paper is to propose a distributed SCHIF based object tracking approach in a VSN. The cameras in the VSN have overlapping field of views (FOV). Each camera can detect and track objects in its FOV. Since the cameras are assumed to be calibrated, each camera is able to measure the coordinates of the object's position on the image plane. The SCHIF runs locally on each camera to estimate the ground plane position of the objects in its FOV. The local SCHIF estimate the states and error covariance in the form of information vector and information matrix, respectively. The cameras exchange the locally estimated object's information contribution vectors and square root information contribution matrices among themselves. In a final step, the participating cameras aggregate the information contribution vectors and subsequently, the object global states.

The remainder of this paper is organized as follows: Section II describes the object's motion model and measurement model in a VSN. Section III presents abstract theoretical concepts of the sequential Bayesian estimation including the CKF, information filtering, and H_∞ filter. Section IV presents the SCHIF algorithm with necessary mathematical descriptions. Section V and Section VI present the simulation results and the conclusions, respectively.

II. System Model

In this work, we consider a VSN consisting of a fixed set of calibrated smart cameras c_i, where $i = 1, 2, \cdots, C$ with overlapping FOVs as illustrated in Fig. 1.

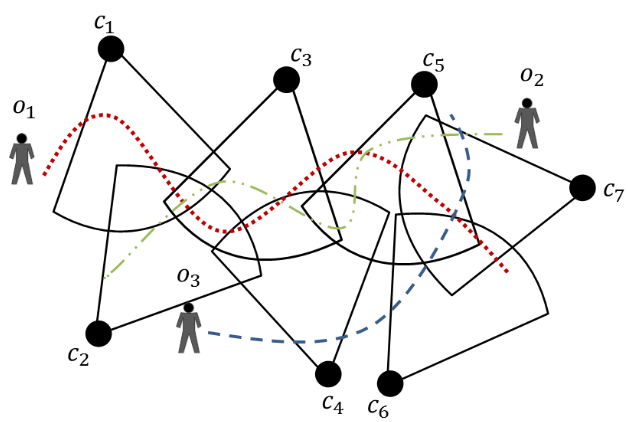


Fig. 1. Visual sensor network.

The task of the object tracker in a VSN is to monitor the given environment and to identify and track a specific object o_j, where $j = 1, 2 \cdots, O$. This is achieved by a distributed tracking algorithm performed by each of the cameras c_i in the network. As these cameras are calibrated, there exists a homography to calculate the object's position on the ground plane. Re-identification and data association of the objects at any camera are typical features of the distributed tracking. They can be achieved with the help of the position on the ground plane in case of overlapping FOVs, or with appearance features calculated in the tracking process. Since the main goal of the work is to present and evaluate the object tracking algorithm in the state estimation perspective, we assume perfect object re-identification and data association.

The state of an object o_j comprises of its position and the velocity on the ground plane. Thus, the state at time k is described as $\mathbf{x}_k^{ij} = \left[x_k^{ij}\ y_k^{ij}\ \dot{x}_k^{ij}\ \dot{y}_k^{ij}\right]^T$, where j and i represent the identity of the object j and the camera i, respectively. The state transition or motion model of the object o_j at camera i and time $k+1$ is given as

$$\mathbf{x}_{k+1}^{ij} = f_{k+1}^{ij}\left(\mathbf{x}_k^{ij}, \mathbf{w}_k^{ij}\right) = \begin{bmatrix} x_k^{ij} + \delta\dot{x}_k^{ij} + \frac{\delta^2}{2}\ddot{x}_k^{ij} \\ y_k^{ij} + \delta\dot{y}_k^{ij} + \frac{\delta^2}{2}\ddot{y}_k^{ij} \\ \dot{x}_k^{ij} + \delta\ddot{x}_k^{ij} \\ x^{ij}{}_k + \delta\ddot{y}_k^{ij} \end{bmatrix}, \quad (1)$$

where $\ddot{x}$ and $\ddot{y}$ are the acceleration of the object in x and y directions that are modeled by the independent and identically distributed (IID) white Gaussian noise vector $\mathbf{w}_k^{ij}$ with covariance $\mathbf{Q}_k^{ij}$. δ is the time interval between the two measurements.

The state of the object is estimated from a set of measurements taken at each time step k. The measurement equation of the object o_j at camera i and time k is given as

$$\mathbf{z}_k^{ij} = \mathbf{h}_k^{ij}\left(\mathbf{x}_k^{ij}\right) + \mathbf{v}_k^{ij}, \quad (2)$$

where $\mathbf{v}_k^{ij}$ is an IID measurement noise vector with covariance $\mathbf{R}_k^{ij}$. The measurement function $\mathbf{h}_k^{ij}$ is the non-linear homography function which converts the object's coordinates from the ground to the image plane. The motion model (1) and measurement model (2) considered in this paper are taken from [9].

III. Non-Linear State Estimation

The Bayesian method estimates the state vector $\mathbf{x}_k$ (The superscript ij is avoided for the sake of simplicity) as the expectation of the posterior probability density function (PDF) $p\left(\mathbf{x}_k \mid \mathbf{z}_{1:k}\right)$ [10]. The sequential Bayesian estimation has two primary steps at every time instance k:

- State prediction:
  Predict the state at k from the posterior PDF at $k-1$.

$$p\left(\mathbf{x}_k \mid \mathbf{z}_{k-1}\right) = \int p\left(\mathbf{x}_{k-1} \mid \mathbf{z}_{k-1}\right) p\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}\right) d\mathbf{x}_{k-1}. \quad (3)$$

- Measurement update:
  Upon receiving the measurement $\mathbf{z}_k$, correct the predicted state.

$$p\left(\mathbf{x}_k \mid \mathbf{z}_k\right) = \frac{p\left(\mathbf{x}_k \mid \mathbf{z}_{k-1}\right) p\left(\mathbf{z}_k \mid \mathbf{x}_k\right)}{\int p\left(\mathbf{x}_k \mid \mathbf{z}_{k-1}\right) p\left(\mathbf{z}_k \mid \mathbf{x}_k\right) d\mathbf{x}_k}. \quad (4)$$

Under the assumptions of linearity and additive Gaussian noise, the integrals in (3) and (4) can be solved by using the KF. Otherwise, these integrals are intractable. Often, the non-linear state estimation methods such as the EKF, UKF, CKF and particle filters are used to approximate them. The efficiency and complexity of these methods depend on the corresponding approximation methodology used. In general, the Monte Carlo approaches such as the PF are more accurate and computationally complex than the EKF, CKF and UKF. However, the CKF is an appealing option for object tracking in a VSN in terms of tracking efficiency and complexity [6]. On the other hand, the CKF requires accurate knowledge of the object such as initial conditions, noise characteristics, and so on. The accurate knowledge of moving objects in a VSN is not readily available in the real time application.

The cubature H_∞ information filter (CHIF) has the ability to deal with the above-mentioned problem. In general, cubature filters are the derivative free non-linear filters. The information version of CKF is not only easy to initialize but also preferred in multi-sensor networks such as a VSN due to relatively easy measurement update step compared to the conventional filters. The H_∞ requires neither the knowledge of the noise characteristics nor the accurate process model. It can deal with uncertain noise conditions and modeling errors of an object's motion in a VSN. The remainder of this section briefly explains the CKF, information filters, and H-infinity filters.

A. Cubature Kalman Filter (CKF)

The CKF is an approximation of the Bayesian filter. Under the assumption that the process and the measurement noises are additive Gaussian, the integrals in (3) and (4) become multi-dimensional Gaussian-weighted integrals of the form

$$I\left(f\right)=\int_{R^{n}}f\left(\mathbf{x}\right)exp\left(-\mathbf{x}^{T}\mathbf{x}\right)d\mathbf{x}. \tag{5}$$

The third degree spherical radial cubature rule is used to approximate this type of integrals. This rule uses spherical-radial transformation to change the variables from the Cartesian to the Radial as: $\mathbf{x}=r\mathbf{z}$ with $\mathbf{z}^{T}\mathbf{z}=1$, such that $\mathbf{x}^{T}\mathbf{x}=r^{2}$ for $r\in[0,\infty)$. Then, the integral (5) can be numerically approximated by

$$I\left(f\right)\approx\frac{\sqrt{\pi^{n}}}{2n}\sum_{l=1}^{2n}f\left(\sqrt{\frac{n}{2}}\xi_{l}\right), \tag{6}$$

where n is the dimension of the vector $\mathbf{x}$ and ξ_{l} is the l-th cubature point located at the intersection of the surface of n-dimensional unit sphere and its axes. This rule can be extended to solve the prediction and posterior PDFs that are in the form of standard Gaussian with mean $\widehat{\mathbf{x}}$ and variance P. Refer [11] for the complete derivation of the CKF.

$$\int_{R^{n}}f\left(\mathbf{x}\right)\mathcal{N}\left(\mathbf{x};\widehat{\mathbf{x}},P\right)\approx\frac{1}{2n}\sum_{i=1}^{2n}f\left(\sqrt{P}\xi_{i}+\widehat{\mathbf{x}}\right). \tag{7}$$

B. Information Filters

The information filter is an alternative version of the KF. In information filtering, the information vector and information matrix are computed and propagated instead of the state vector and error covariance. The information matrix and information vector at each iteration k are given

$$\mathbf{Y}_{k}=\mathbf{P}_{k}^{-1}, \tag{8}$$

$$\widehat{\mathbf{y}}_{k}=\mathbf{Y}_{k}\widehat{\mathbf{x}}_{k}, \tag{9}$$

where $\widehat{\mathbf{x}}_{k}$ and $\mathbf{P}_{k}$ are state vector estimate and error covariance matrix, respectively. The information filters are easy to initialize compared to conventional filters when the initial conditions are not known. Moreover, the relatively low complex measurement update makes them preferable for distributed applications. The cubature information filter (CIF) is presented in [12].

C. H_{∞} Filter

The H_{∞} filter is based on the game theory approach which estimates a linear combination of the state variables

$$\mathbf{l}_{k}=\mathbf{L}_{k}\mathbf{x}_{k}, \tag{10}$$

where $\mathbf{L}_{k}$ is a user defined full rank matrix. If $\mathbf{L}_{k}$ becomes identity matrix then, the state vector is estimated. The H_{∞} filter is designed to minimize the state estimation error under the worst case $\mathbf{w}_{k}$, $\mathbf{v}_{k}$, and $\mathbf{x}_{0}$ in such a way that the cost function $\mathbf{J}_{\infty}$ is bounded by a error attenuation factor $\gamma>0$. The cost function $\mathbf{J}_{\infty}$ is given as

$$\mathbf{J}_{\infty}=\frac{\sum_{k=0}^{N-1}\left\|\mathbf{l}_{k}-\widehat{\mathbf{l}}_{k}\right\|_{\mathbf{S}_{k}}^{2}}{\left\|\mathbf{x}_{0}-\widehat{\mathbf{x}}_{0}\right\|_{\mathbf{P}_{0}^{-1}}^{2}+\sum_{k=0}^{N-1}\left(\left\|\mathbf{w}_{k}\right\|_{\mathbf{Q}_{k}^{-1}}^{2}+\left\|\mathbf{v}_{k}\right\|_{\mathbf{R}_{k}^{-1}}^{2}\right)}, \tag{11}$$

where $\mathbf{S}_{k}$, $\mathbf{P}_{0}^{-1}$, $\mathbf{Q}_{k}^{-1}$, and $\mathbf{R}_{k}^{-1}$ are user defined weighting matrices based on the problem. The H_{∞} filter can deal with unknown noise parameters and process model.

The above-mentioned three methods can be intuitively combined to yield a more robust and efficient filter. The authors in [8] presented the cubature H_{∞} information filter (CHIF). This algorithm requires the preservation of positive definitiveness and symmetry of the information matrix in each iteration. The CHIF involves numerical operations such as matrix inverse and square rooting which can destroy the above-mentioned properties of the information matrix. In this paper, we propose a square root version of the CHIF (SCHIF) which improves the numerical stability of the algorithm by preserving the fundamental properties of the information matrix.

IV. Square Root Cubature H_{∞} Information Filter (SCHIF)

Before presenting SCHIF, we briefly introduce some mathematical relations and notations. Let $\mathbf{Y}$ be the information matrix

$$\mathbf{Y}=\mathbf{P}^{-1}=\mathbf{S}_{y}\mathbf{S}_{y}^{T}, \tag{12}$$

where $\mathbf{P}$ and $\mathbf{S}_{y}$ are error covariance and square root of the information matrix, respectively. Let $\mathbf{S}$ be the square root of the error covariance

$$\mathbf{P}=\mathbf{S}\mathbf{S}^{T}. \tag{13}$$

From (12) and (13), we yield

$$\mathbf{S}=\mathbf{S}_{y}^{-T}\Rightarrow\mathbf{y}=\mathbf{Y}\mathbf{x}=\mathbf{P}^{-1}\mathbf{x}=\mathbf{S}^{-T}\mathbf{S}^{-1}\mathbf{x}. \tag{14}$$

From (12), (13), and (14) the error covariance and information matrix can be replaced with their corresponding square roots. The SCHIF algorithm has two steps at each time instant $k+1$.

A. Time Update

Assume that the estimated information vector $\widehat{\mathbf{y}}_{k|k}$ and square root of the information matrix $\mathbf{S}_{y,k|k}$ from the previous iteration k are known. Compute the square root error covariance as

$$\mathbf{S}_{k|k}=\mathbf{S}_{y,k|k}^{-T}. \tag{15}$$

Calculate the state estimate at time k

$$\widehat{\mathbf{x}}_{k|k}=\mathbf{S}_{k|k}\mathbf{S}_{k|k}^{T}\widehat{\mathbf{y}}_{k|k}. \tag{16}$$

Compute the cubature points $l=(1,2,...,2n)$ using $\widehat{\mathbf{x}}_{k|k}$ and $\mathbf{S}_{k|k}$ as

$$\mathbf{c}_{l,k|k}=\mathbf{S}_{k|k}\xi_{l}+\widehat{\mathbf{x}}_{k|k}, \tag{17}$$

where n is the length of the state vector. ξ_{l} represent the l-th intersection point of the surface of the n-dimensional unit sphere and its axes. These cubature points are propagated through the process model as

$$\mathbf{x}_{l,k+1|k}^{*}=f_{k}\left(\mathbf{c}_{l,k|k}\right). \tag{18}$$

Calculate the predicted state as

$$\widehat{\mathbf{x}}_{k+1|k}=\frac{1}{2n}\sum_{l=1}^{2n}\mathbf{x}_{l,k+1|k}^{*}. \tag{19}$$

Calculate the square root of the predicted error covariance as

$$\mathbf{S}_{k+1|k}=\textbf{Tria}\left[\mathbf{M}_{k+1|k}\quad\mathbf{S}_{\mathbf{Q},k+1}\right], \tag{20}$$

where $\mathbf{S}_{\mathbf{Q},k+1}$ is the square root of the process noise covariance $\mathbf{Q}_{k+1}$. The function **Tria** is a triangularization algorithm such as **QR** decomposition (for details refer to [11]). The predicted weighted centered matrix $\mathbf{M}_{k+1|k}$ is given as

$$\mathbf{M}_{k+1|k} = \frac{1}{\sqrt{2n}}[\mathbf{x}^*_{1,k+1|k} - \widehat{\mathbf{x}}_{k+1|k} \quad \mathbf{x}^*_{2,k+1|k} - \widehat{\mathbf{x}}_{k+1|k} \\ \cdots \mathbf{x}^*_{2n,k+1|k} - \widehat{\mathbf{x}}_{k+1|k}]. \quad (21)$$

Compute the square root of the predicted information matrix and predicted information vector

$$\mathbf{S}_{y,k+1|k} = \mathbf{S}^{-T}_{k+1|k}, \quad (22)$$

$$\widehat{\mathbf{y}}_{k+1|k} = \mathbf{S}_{y,k+1|k}\mathbf{S}^T_{y,k+1|k}\widehat{\mathbf{x}}_{k+1|k}. \quad (23)$$

B. Measurement Update

Upon receiving the measurement $\mathbf{z}_{k+1}$, the cubature points are calculated using the predicted state $\widehat{\mathbf{x}}_{k+1|k}$ and the square root of the predicted error covariance $\mathbf{S}_{k+1|k}$ as

$$\mathbf{c}_{l,k+1|k} = \mathbf{S}_{k+1|k}\xi_l + \widehat{\mathbf{x}}_{k+1|k}. \quad (24)$$

Propagate the cubature points through the measurement function

$$\mathbf{z}^*_{l,k+1|k} = \mathbf{h}_{k+1}\left(\mathbf{c}_{l,k+1|k}\right). \quad (25)$$

The predicted measurement is calculated as

$$\widehat{\mathbf{z}}_{k+1|k} = \frac{1}{2n}\sum_{l=1}^{2n}\mathbf{z}^*_{l,k+1|k}. \quad (26)$$

The cross covariance is calculated as

$$\mathbf{P}_{xz,k+1|k} = \frac{1}{2n}\sum_{l=1}^{2n}\mathbf{c}_{l,k+1|k}\mathbf{z}^{*T}_{l,k+1|k} - \widehat{\mathbf{x}}_{k+1|k}\widehat{\mathbf{z}}^T_{k+1|k}. \quad (27)$$

The square root factor of the information contribution matrix is calculated as

$$\mathbf{SI}_{k+1} = \mathbf{S}_{y,k+1|k}\mathbf{S}^T_{y,k+1|k}\mathbf{P}_{xz,k+1|k}\mathbf{S}^{-T}_{R,k+1}, \quad (28)$$

where $\mathbf{S}_{R,k+1}$ is the square root of the measurement noise covariance matrix. Compute the information contribution vector

$$\mathbf{iv}_{k+1} = \mathbf{SI}_{k+1}\mathbf{S}^{-T}_{R,k+1}\left(\mathbf{z}_{k+1} - \widehat{\mathbf{z}}_{k+1|k}\right) \\ + \mathbf{SI}_{k+1}\mathbf{SI}^T_{k+1}\widehat{\mathbf{x}}_{k+1|k}. \quad (29)$$

The updated information vector is

$$\widehat{\mathbf{y}}_{k+1|k+1} = \widehat{\mathbf{y}}_{k+1|k} + \mathbf{iv}_{k+1}. \quad (30)$$

Compute the square root of the updated information vector

$$\mathbf{S}_{y,k+1|k+1} = \mathbf{Tria}\left[\mathbf{S}_{y,k+1|k} \quad \mathbf{SI}_{k+1} \quad \left(-\gamma^{-2}\right)^{\frac{1}{2}}\mathbf{I}\right], \quad (31)$$

where γ is error attenuation factor and $\mathbf{I}$ is the n-dimensional unity matrix.

C. Distributed Measurement Update

Let the VSN has C cameras. Each camera c_i has a local information contribution vector and square root information contribution matrix $(\mathbf{iv}_{i,k+1}, \mathbf{SI}_{i,k+1})$, where $i = \{1, 2, \cdots, C\}$ as shown in (29) and (28), respectively. The global information vector and square root information matrix at each camera c_i are calculated as

$$\widehat{\mathbf{y}}_{i,k+1|k+1} = \widehat{\mathbf{y}}_{i,k+1|k} + \sum_{i=1}^{C}\mathbf{iv}_{i,k+1}, \quad (32)$$

$$\mathbf{S}_{iy,k+1|k+1} = \mathbf{Tria}[\mathbf{S}_{iy,k+1|k} \quad \mathbf{SI}_{1,k+1} \quad (33)$$
$$\cdots \mathbf{SI}_{C,k+1} \quad C\left(-\gamma^{-2}\right)^{\frac{1}{2}}\mathbf{I}]. \quad (34)$$

The pseudo algorithm of the object tracking based on the distributed SCHIF is given in Algorithm 1

Algorithm 1 SCHIF Pseudo Algorithm

Initialize the filter at each camera c_i with $\left(\widehat{\mathbf{y}}_{i,0|0}, \mathbf{S}_{iy,0|0}\right)$, where $i \in c = \{1, 2, \cdots, C\}$

At each time step $k+1$ and each camera c_i the posterior information vector and information matrix $\left(\widehat{\mathbf{y}}_{i,k|k}, \mathbf{S}_{iy,k|k}\right)$ from time k are known

for $i = 1$ to C **do**

Perform time update ▷ refer to IV-A

$$\left[\widehat{\mathbf{y}}_{i,k+1|k}, \mathbf{S}_{iy,k+1|k}\right] = \mathrm{TU}\left[\widehat{\mathbf{y}}_{i,k|k}, \mathbf{S}_{iy,k|k}\right]$$

Upon receiving the measurement $\mathbf{z}_{i,k+1}$, perform measurement update ▷ refer to IV-B

$$[\mathbf{iv}_{i,k+1}, \mathbf{SI}_{i,k+1}] = \mathrm{MU}\left[\widehat{\mathbf{y}}_{i,k+1|k}, \mathbf{S}_{iy,k+1|k}, \mathbf{z}_{i,k+1}\right]$$

Send $[\mathbf{iv}_{i,k+1}, \mathbf{SI}_{i,k+1}]$ to all the other cameras ℓ in the network, where $\ell \in c$ and $\ell \neq i$

Receive $[\mathbf{iv}_{\ell,k+1}, \mathbf{SI}_{\ell,k+1}]$, from the other cameras ℓ in the network, where $\ell \in c$ and $\ell \neq i$

Fuse the received information ▷ refer to IV-C

$$\left[\widehat{\mathbf{y}}_{i,k+1|k+1}, \mathbf{S}_{iy,k+1|k+1}\right] = \mathrm{DMU}[\mathbf{iv}_{1:C,k+1}, \mathbf{SI}_{1:C,k+1}]$$

Find the global state

$$\widehat{\mathbf{x}}_{i,k+1|k+1} = \mathbf{S}_{iy,k+1|k+1}\mathbf{S}^T_{iy,k+1|k+1}\widehat{\mathbf{y}}_{i,k+1|k+1}$$

end for

V. Simulation Results

In this section, we discuss the efficiency of the SCHIF based distributed object tracking in comparison with the square root cubature information filter (SCIF), square root unscented information filter (SUIF) and square root CKF (SCKF) based distributed object tracking. The simulation considers a VSN with five cameras having overlapping FOVs as shown in Fig. 2. All of the cameras can partially observe xy-plane, where $x \in [0, 500]$ and $y \in [0, 500]$. The motion of the objects is modeled by a constant velocity with Gaussian distributed acceleration as given in (1). The ground truth of the position of the object is simulated by assuming the process noise covariance $\mathbf{Q}_k^{ij}$ and measurement noise covariance $\mathbf{R}_k^{ij}$ are diag $(10, 10)$ and diag $(1, 1)$, respectively. Each camera c_i has its own homography function h^i. Since we assume the fixed cameras, the homography of a camera does not change with time k and object j.

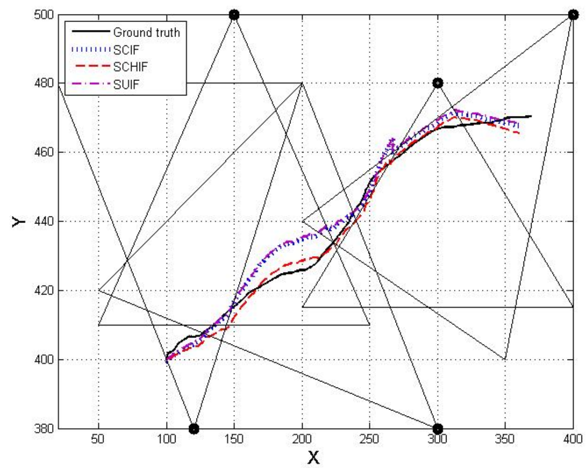
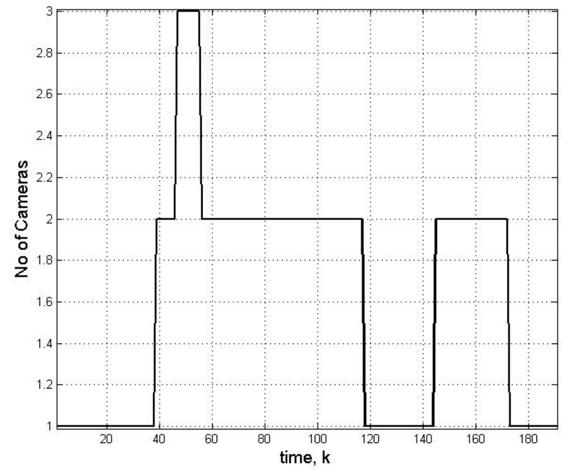
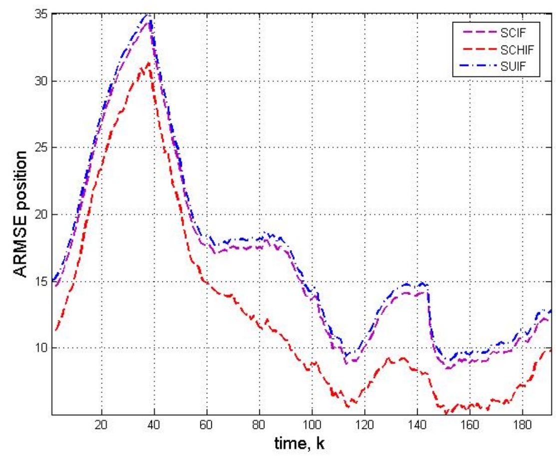


Fig. 2. Comparison of the tracking accuracy of the distributed object tracking methods based on the SCHIF SCIF, and SUIF.



(a) Number of cameras that can observe the object at each time k.



(b) Average network RMSE of the estimated position of the object using the methods based on the SCHIF, SCIF, and SUIF.

Fig. 3. Efficiency of the distributed object estimation methods based on the SCHIF, SCIF, and SUIF in the considered VSN.

In first scenario, the efficiency of the distributed object tracking methods based on the SCHIF, SCIF and SUIF is compared. Fig. 2 illustrates the estimated position of the object using all the three methods by the camera located at $[300, 380]$ compared to the ground truth. The accurate knowledge of the process noise $\mathbf{Q}_k^{ij}$ and measurement noise $\mathbf{R}_k^{ij}$ is considered to be unknown to the filters. Hence, in the estimation process, the filters at all the cameras assume the process noise covariance $\mathbf{Q}_k^{ij}$ and measurement noise covariance $\mathbf{R}_k^{ij}$ to be diag $(20, 20)$ and diag $(2, 2)$, respectively. The number of cameras that can observe the object at each time step k varies as shown in Fig. 3(a). Each camera that can observe the object at time k, performs the local estimation. The locally estimated information contribution vectors and square root information contribution matrices are exchanged among all the cameras in the network. Then, Each camera performs the data fusion as described in the Section IV to achieve the global state of the object. The VSN is also assumed to be fully connected with perfect communication links. Under these conditions, the simulation result in Fig. 2 show that the SCHIF based object tracking features a significant improvement over the SCIF and SUIF based methods. Since, the SCIF is a special case of SUIF, their tracking results are comparable.

Fig. 3(b) shows the average root mean square error (RMSE) of overall network for all three distributed object state estimation methods. The network RMSE is calculated by averaging the individual RMSE of all the cameras in the network. To achieve the statistical reliability, the average RMSE averaged over a thousand simulation runs. Each simulation run considers the same trajectory as shown in Fig. 2 with different initializations. Hence, the number of cameras that can observe the object in each simulation run does not change and it is shown in Fig. 3(a). From Fig. 3, we can infer that the SCHIF based object tracking outperforms the SCIF and SUIF based methods. Moreover, the tracking efficiency of all the three methods improves with increasing number of cameras. The average network RMSE of the three methods is given in Table I

In second scenario, the efficiency of the the distributed object tracking based on the cubature filters such as the SCHIF, SCIF and SCKF is compared. The CKF based distributed

TABLE I. AVERAGE NETWORK RMSE OF THE ESTIMATED POSITION AND VELOCITY USING THE DISTRIBUTED OBJECT TRACKING METHODS BASED ON THE SCHIF, SCIF, SUIF AND SCKF.

Method	ARMSE (position)	ARMSE (velocity)
SCHIF	12.7023	6.1814
SCIF	16.5198	7.8411
SUIF	17.1534	7.9823
SCKF	21.892	8.6678

object tracking is presented in [6]. The simulation considers the same conditions as in the first scenario. Fig. 4 illustrates average RMSE of the estimated position of the object using the distributed cubature filters for different number of cameras. The figure illustrates that the SCHIF based method clearly outperforms the SCKF and SCIF based methods. At the same time, all the cubature filters fundamentally have the same efficiency when the accurate information about the system is known.

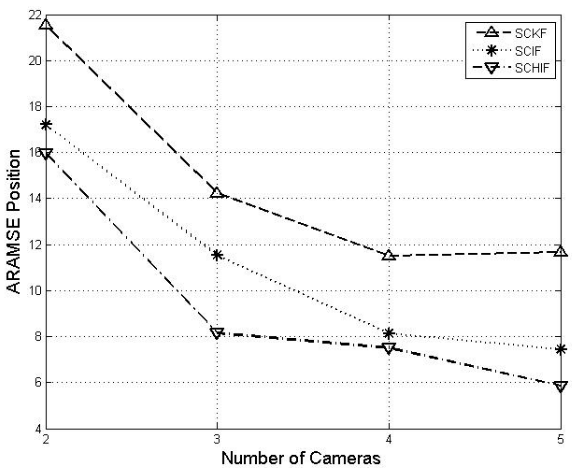


Fig. 4. Comparison of the tracking accuracy of the distributed object tracking algorithms based on the SCHIF, SCIF and SCKF.

VI. CONCLUSION

In this paper, a distributed square root cubature H_∞ information filter (SCHIF) based object tracking is proposed. In a visual sensor network with calibrated smart cameras, each camera tracks objects in its field of view using the local SCHIF. Then, all the cameras in the network exchange the locally estimated information contribution vectors and square root information contribution matrices among themselves. The global state at each camera is calculated by fusing the information contribution vectors from all the cameras. In addition, we also compared the efficiency of the this approach with cubature Kalman filter, square root cubature information filter and square root unscented information filter based tracking methods using a simulation based analysis. Our results show that the SCHIF based method outperforms the all the above-mentioned methods in terms of the efficiency and robustness.

ACKNOWLEDGMENT

This work was supported in part 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.

REFERENCES

[1] Bernhard Rinner, Markus Quaritsch, Wolfgang Schriebl, Thomas Winkler, and Wayne Wolf, *The Evolution from Single to Pervasive Smart Cameras*, ACM/IEEE International Conference on Distributed Smart Cameras, pp. 1-10, 2008.

[2] Reza Olfati-Saber and Nils F. Sandell, *Distributed Tracking in Sensor Networks with Limited Sensing Range*, IEEE American Control Conference, pp. 3157-3162, 2008.

[3] Bi Song, Chong Ding, Ahmed T. Kamal, Jay A. Farrell, and Amit K. Roy Chowdhury, *Distributed Camera Networks*, IEEE Signal Processing Magazine, Vol. 28, pp. 20-31 2011.

[4] Chong Ding, Bi Song, Akshay A. Morye, Jay A. Farrell, and Amit K. Roy Chowdhury, *Collaborative Sensing in a Distributed PTZ Camera Network*, IEEE Transactions on Image Processing, Vol. 21, pp. 3282-3295, 2012.

[5] Ahmed T. Kamal, Jay A. Farrell, and Amit K. Roy Chowdhury, *Information Weighted Consensus Filters and their Application in Distributed Camera Networks*, IEEE Transactions on Automatic Control, Vol.58, pp. 3112-3125, 2013.

[6] Venkata P. Bhuvana, Melanie Schranz, Mario Huemer, and Bernhard Rinner, *Distributed Object Tracking based on Cubature Kalman Filter*, IEEE ASILOMAR conference on Signals, Systems, and Computers, 2013.

[7] Simon Dan, *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*, John Wiley and Sons, 2006.

[8] Kumar P. Bharani Chandra, Da-Wei Gu, and Ian Postlethwaite, *Cubature H Information Filter*, European Control Conference (ECC) pp. 2639-2644, 2013.

[9] Henry Medeiros, Johnny Park, and Avinash C. Kak, *Distributed Object Tracking Using a Cluster-Based Kalman Filter in Wireless Camera Networks*, IEEE Journal of Selected Topics in Signal Processing, vol.2, pp. 448-463, 2008.

[10] Steven M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, Vol.1, 1993.

[11] Ienkaran Arasaratnam and Simon Haykin, *Cubature Kalman Filters*, IEEE Transactions on Automatic Control, vol.54, pp. 1254-1269, 2009.

[12] Kumar P. Bharani Chandra, Da-Wei Gu, and Ian Postlethwaite, *Cubature Information Filter and its Applications*, American Control Conference (ACC), pp. 3609-3614, 2011.