IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 9, 2018, accepted January 6, 2019, date of publication January 23, 2019, date of current version February 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894672

Square-Root Cubature Information Hybrid Consensus Filter With Correlated Noise and Its Applications in Camera Networks

YANMING CHEN AND JIAHUI WANG
School of Computer Science and Technology, Anhui University, Hefei 230601, China
Corresponding author: Yanming Chen (cym@ahu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61802001.

ABSTRACT This paper proposes a hybrid consensus-based square-root cubature information filter for target tracking in camera networks in the case that the process and measurement noises are correlated with each other at the same time. To the best of our knowledge, this is the first work that shows how to utilize the square-root cubature information filter with correlated noise at the same time to track the target in camera networks. This paper first decouples the correlation and rearranges the state transition equation to a new one. Then, apply square-root cubature information filter based on the new state transition equation and the original measurement equation to proposed a hybrid consensus-based square-root cubature information filter with correlated noise (SCHF-CN). What's more, it is proved that the proposed algorithm is stable via the consistency of estimates. The simulation results demonstrate the superior performance of the proposed SCHF-CN as compared to other algorithms via applications about the target tracking in camera networks.

INDEX TERMS Correlated noise, camera networks, distributed tracking, information filter.

I. INTRODUCTION

In the past few decades, more and more attention has been paid to nonlinear recursive state estimation of discrete-time systems because of its widespread applications such as target tracking [1], signal processing [2], etc. There are lots of recursive algorithms for state estimation of nonlinear systems, such as the classical extended Kalman filter(EKF) [3], unscented Kalman filter(UKF) [4], and recently proposed cubature Kalman filters(CKF) [5], which are based on the assumption that the process and measurement noises are uncorrelated with each other. But in actual application the process and measurement noises may be correlated with each other [6]–[8], e.g. state estimation for camera networks with non-synchronized communication between cameras. In this paper, we will consider the target tracking problem with correlated noise in camera networks. Usually, according to the correlation time between the process and measurement noises, the correlated noise is divided into cross-correlation at the same time and cross-correlation one time step apart [9]. Because the cross-correlation at the same time is more common, most studies consider this kind of correlation [10]–[13]. Of course, cross-correlation one time step apart also has been studied [14]. Our work is deferent from [14] in the type of the cross-correlation and the form of filter. It has been confirmed in our previous article [15] that the square root cubature information filter has a significant effect compared to the general cubature information filter in handling the errors introduced by arithmetic operations on finite word-length digital computers. So we will discuss the problem of correlated noise in the square-root cubature information filter. To the best of our knowledge, this is the first work that shows how to utilize the square-root cubature information filter with correlated noise **at the same time** to track the target in camera networks.

The camera network considered in this paper is a set of resource-constrained camera-equipped sensor nodes that are spread over a large area. For target tracking in this network, distributed state estimation (DSE) have obvious advantages over corresponding centralized algorithms. When implementing DSE in a distributed camera network, a consensus-based algorithm is required to integrate the states from each camera. The consensus-based methods for DSE can be classified into several categories [16]. The first category is called consensus on estimates (CE), which achieves information exchange with cameras via spreading the available information on the network, and then performs an

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

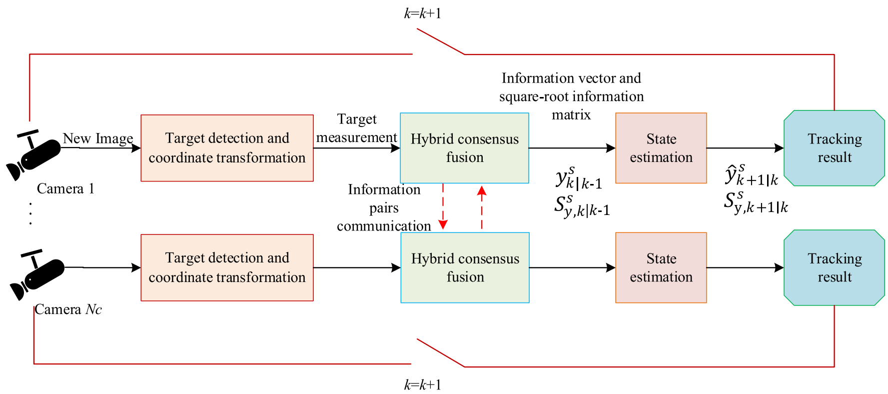


FIGURE 1. Overall system diagram depicting a framework for target-tracking in a camera network.

average consensus on state estimates from camera itself and neighbors [17]–[19]. In this average consensus algorithm, only using state estimations can reduce communication cost of each camera and then the communication overhead of the entire system. But covariance matrices also contain lots of useful information to improve the system performance. Consequently, the second category: consensus on measurements (CM) is proposed, which performs the average consensus on local measurements and innovation covariances. It is shown that CM can approximate, in a distributed way, the correction step of the centralized Kalman-like filter. The covariance intersection fusion rule [20] suggests an another consensus approach, namely consensus on the information (CI) [21], which performs a consensus among the inverse covariance (information) matrix and the information vector. CI algorithms can ensure boundedness of the estimation error for any number of consensus steps (even a single one) [21], but this kind of approach relies on the assumption that the correlation between the estimates receiving from different nodes to be completely unknown, which may lead to a bad effect on its mean-square estimation error performance. Although CM strategy does not make any assumptions about the correlation between the estimates by fusing only the novel information, it does not guarantee stability unless the number of consensus steps is sufficiently high. Hence, in order to combine the advantages and neutralize the disadvantages of both CI and CM, a hybrid consensus approach called HCMCI (Hybrid Consensus on Measurement and Consensus on Information) is proposed [22], [23]. Different choices for the combination weights lead to different properties. When the combination weights choice $1/N_C$ (N_C is the total number of cameras in the network), the HCMCI approach is equivalent to the Information Consensus Filter (ICF) [24].

What's more, one advantage of the information filter over the cubature Kalman filters arises from its natural fit for multi-agent problems. Because information filters often integrate the information in an arbitrary order, with arbitrary delays and in a completely decentralized manner [15].

Based on the above study, this paper proposes a distributed square-root cubature information hybrid consensus filter with correlated noise for target tracking in the camera network, its framework shown in Fig. 1. When the video is captured from a camera, the following processes must be accomplished. The first process is target detection using existing algorithms, such as the target detection method based on background subtraction [25]. The target detection module in each camera takes its raw image and returns the image plane positions of each target recognized in the image, and then project the targets' positions in different image planes to the same ground plane. The second process is distributed information fusion using a hybrid consensus algorithm. Communication among cameras is allowed to enhance the processes of information fusion and association for target recognition. The third process is target state estimation using square-root cubature information filter with correlated noise. Using the prior information (information vector $\hat{y}^s_{k|k-1}$ information matrix $S^s_{y,k|k-1}$) at time k and the assumed target model, the prior information ($\hat{y}^s_{k+1|k}$ and $S^s_{y,k+1|k}$)) is computed as the tracking result and an input for time $k+1$.

The **main contributions** of this paper include: (1) a distributed nonlinear information filter with correlated noise at the same time and hybrid consensus strategy has been developed with the framework of square-root cubature information filter; (2) the consistency of estimates for the proposed algorithm has been proven with the help of the pseudo system and a pseudo measurement matrix. The structure of this paper is as follows. We introduce the system model for this paper in Section II. In Section III, a square-root cubature information hybrid consensus filter with correlated noise (SCHF-CN) is proposed. The stability analysis of the proposed algorithm is given in Section IV. The algorithm applies in experiments to compare against others in section V. The simulation results show that the proposed algorithm can efficiently track the target in camera networks. Finally, we give conclusion of this paper in section VI.

II. SYSTEM MODEL

In this paper, we consider a camera network with N_C cameras to monitor moving targets in overlapping field of views (FOVs). The communication in the network can be represented using an undirected connected graph $G = (C, E, A)$ [26]. The sets of vertices $C = \{C_1, C_2, \cdots, C_{N_C}\}$ and edges $E \subseteq C \times C$ represent the cameras and available communication channels between different cameras, respectively. $A = [a_{sj}]_{N_C \times N_C}$ is an adjacency matrix which is a symmetric 01−matrix. Because the graph has no loops, the diagonal entries of A are zero ($a_{ss} = 0, s = 1, \cdots, N_C$). (C_s, C_j) represents the direct communication channel between node C_s and node C_j. $\Omega_s = \{C_j \in C \mid (C_s, C_j) \in E\}$ is the adjacency set of node C_s. $N_s = C_s \cup \Omega_s$ is the set including the node C_s and its direct neighbors. The degree of node C_s is the number of its neighbors $\triangle_s = \sum_j a_{sj}$. The degree matrix is a $N_C \times N_C$ matrix defined as $\Delta = diag\{A \cdot 1\}$.

The general nonlinear system model for camera networks is the form

$$x^s_{k+1} = f(x^s_k) + v^s_k \quad (1)$$
$$z^s_k = h(x^s_k) + w^s_k \quad (2)$$

where the system equation $f(\cdot)$ and the measurement equation $h(\cdot)$ are time-varying nonlinear functions. At time k, $x^s_k \in R^{n_x}$ is the state vector of the target in the camera C_s, n_x is the dimension of the state vector. For a target, the state vectors obtained by different cameras should be broadly consistent in theory. $z^s_k \in R^{n_z}$ is the nonlinear measurement from the target measured by the node C_s on time k, n_z is the dimension of the observation vector. v^s_k and w^s_k are process noise and measurement noise in the camera C_s at time k, respectively. They are both Gaussian white noise sequences with zero mean and variances are Q^s_k and R^s_k, which satisfy

$$E\left\{\begin{pmatrix} v^s_k \\ w^s_k \end{pmatrix} \begin{pmatrix} v^{sT}_k & w^{sT}_k \end{pmatrix}\right\} = \begin{bmatrix} Q^s_k & D^s_k \\ D^{sT}_k & R^s_k \end{bmatrix} \delta_{k,l} \quad (3)$$

where D^s_k is covariance of process noise v^s_k and measurement noise w^s_k. $\delta_{k,l}$ is the Kronecker delta, which is 1 if the time l and time k are equal, and 0 otherwise. Many researches [24], [27] assume that $D^s_k = 0$, but in fact this assumption is not reasonable. Therefore, this paper considers the case where the covariance of these two noises is not zero, i.e., $D^s_k \neq 0$. That is to say, the system used in this paper is a system with correlated noise.

III. SQUARE-ROOT CUBATURE INFORMATION HYBIRD CONSENSUS FILTER WITH CORRELATED NOISE

Many previous researches on system modeling to assume that the measurement noise of the system equation is uncorrelated with the process noise, but in fact, the two noises may be related. In this section we will discuss the concrete expressions of the hybrid consensus filter in the system with correlated noise.

A. THE DECOUPLE METHOD FOR SQUARE-ROOT CUBATURE INFORMATION FILTER WITH CORRELATED NOISE

The cubature Kalman filter (CKF) [5] and the square-root cubature information filter (SCIF) [28] were proposed by Arasaratnam *et al.* at 2009 and 2013, respectively. However, the classical cubature Kalman filter and square-root cubature information filter do not take correlated noise into account (i.e., $D^s_k = 0$), which are not suitable for systems with correlated noise. In addition, the information filter is more suitable for the information fusion in multisensor systems [3]. Therefore, this subsection will give the form of square-root cubature information filter with correlated noise.

Since two noises in (1) and (2) are correlated, the state estimation can not be calculated directly via the classical square-root cubature information filter [28]. In this section, we deal with the square-root cubature information filter with correlated noise using a similar approach, which is proposed for Kalman filter with correlated noise [9].

Using a matrix T_k, to be determined later, one can rewrite (1) as follow (for the sake of brevity, the index s is omitted in (1) and (2)).

$$\begin{aligned} x_{k+1} &= f(x_k) + v_k + T_k[z_k - h(x_k) - w_k] \\ &= f(x_k) + T_k[z_k - h(x_k)] + v_k - T_k w_k \\ &= f^*(x_k) + v^*_k \end{aligned} \quad (4)$$

where $f^*(x_k) = f(x_k) + T_k[z_k - h(x_k)]$ and the new process noise as $v^*_k = v_k - T_k w_k$. z_k is a known observation. Since $z_k - h(x_k) - w_k = 0$, then the (4) is equivalent to (1). In order to use the framework of the classical SCIF, the cross-correlation between the new process noise v^*_k and the measurement noise w_k must be zero. That is,

$$E[v^*_k w^T_k] = E[(v_k - T_k w_k) w^T_k] = D_k - T_k R_k = 0 \quad (5)$$

where D_k and R_k are defined in (3). From (5), $T_k = D_k R^{-1}_k$. With the above, the covariance of the new process noise is

$$\begin{aligned} Q^*_k &= E[v^*_k (v^*_k)^T] \\ &= E[(v_k - D_k R^{-1}_k w_k)(v_k - D_k R^{-1}_k w_k)^T] \\ &= Q_k - D_k R^{-1}_k D^T_k. \end{aligned} \quad (6)$$

By identity transformation, the modified nonlinear system model for camera networks is the form

$$x^s_{k+1} = f^*(x^s_k) + v^{*s}_k \quad (7)$$
$$z^s_k = h(x^s_k) + w^s_k. \quad (8)$$

Now, the new state estimation in (7) and (8) can be calculated directly via the classical square-root cubature information filter [28].

B. DISTRIBUTED SQUARE-ROOT CUBATURE INFORMATION HYBIRD CONSENSUS FILTER WITH CORRELATED NOISE

Each camera in the camera network is distributed in different locations in the observation area. If centralized management is adopted, it will inevitably lead to difficulty in wiring, cause a large communication cost and storage load to the central node. At present, the emergence of Internet of things (IoT) and cloud computing have made the rapid development of distributed processing technology, which technical advantages have been highlighted. Therefore, this paper also uses a distributed approach for information exchanging and processing. The application scenario of this paper is the camera network. In a camera network, each camera is regarded as a node, each node only establishes communication and exchanges the information with its neighbors. Because the cameras are arranged in different locations in a large scale area, the FoV of each camera is different. Thus, a distributed average consensus algorithm is needed to make the information consistent from each camera, so as to realize the information fusion of the whole network. Because of benefits of HCMCI approach, this paper chooses this consensus stratety as the multi-source information fusion strategy. The input of the hybird consensus algorithm in this paper is the information vector $\hat{y}^s_{k|k}$ and square-root information matrix $S^s_{y,k|k}$, where $s = 0, 1, \cdots, N_C$, $S^s_{y,k|k}$ is determined by information matrix $Y^s_{k|k} \left(= (P^s_{k|k})^{-1} = S^s_{y,k|k}(S^s_{y,k|k})^T\right)$ [28]. For simplicity, $\hat{y}^s_{k|k}$ and $S^s_{y,k|k}$ are denoted by the information pairs $(\hat{y}^s_{k|k}, S^s_{y,k|k})$, then make the following definition:

Definition 1: Given the information pairs in the form of $(\hat{y}^s_{k|k}, S^s_{y,k|k})$, $C_s \in C$, it is said their hybird average consensus is achieved if the following limit exists, that is

$$(\hat{y}^*_{k|k}, S^*_{y,k|k}) = \lim_{l\to\infty} (\hat{y}^s_{k|k,l}, S^s_{y,k|k,l}) \tag{9}$$

In (9), $\left(\hat{y}^s_{k|k,l}, S^s_{y,k|k,l}\right)_{C_s\in C}$ denotes the information pair of camera C_S available at time step k after the l-th internal iteration satisfying

$$\hat{y}^s_{k|k,l+1} = \sum_{C_j\in N_s} \pi^{s,j}\, \hat{y}^j_{k|k-1,l} + \omega^s_k \sum_{C_j\in N_s} \pi^{s,j} i^j_{k,l} \tag{10}$$

$$\begin{aligned} S^s_{y,k|k,l+1} = Tria\Big(\Big[&\sqrt{\pi^{s,s}}S^s_{y,k|k-1,l} \quad \sqrt{\pi^{s,1}}S^1_{y,k|k-1,l} \cdots \\ &\sqrt{\pi^{s,N_{s,E}}}S^{N_{s,E}}_{y,k|k-1,l} \quad \sqrt{\omega^s_k \pi^{s,s}}S^s_{i,k,l} \\ &\sqrt{\omega^s_k \pi^{s,1}}S^1_{i,k,l} \cdots \sqrt{\omega^s_k \pi^{s,N_{s,E}}}S^{N_{s,E}}_{i,k,l}\Big]\Big) \end{aligned} \tag{11}$$

with weighting coefficients $\pi^{sj} \geq 0$, N_s is the set including the node C_s and its neighbors $C_j \in \Omega_s$, $i^j_{k,l}(C_j \in N_s)$ and $S^m_{i,k,l}(m = 1, \cdots, N_{s,E})$ are the information contribution vector and the square-root information matrix, respectively [28]. *Tria* represents the QR decomposition [5]. Since $\left(C_s, C_j\right) \in E$, it is easy to obtain the total number $N_{s,E}$ of C_j adjacent to C_s by the adjacency matrix A. In addition, $\sum_{C_j\in N_s} \pi^{sj} = 1$ and the initial conditions are assumed to be $\hat{y}^s_{k|k,0} = \hat{y}^s_{k|k}$, $S^s_{y,k|k,0} = S^s_{y,k|k}$. The (11) is the square root form of following equation.

$$Y^s_{k|k,l+1} = \sum_{C_j\in N_s} \pi^{s,j} Y^j_{k|k-1,l} + \omega^s_k \sum_{C_j\in N_s} \pi^{s,j} I^j_{k,l} \tag{12}$$

where $Y^s_{k|k,l+1}$ is the information matrix at the $l_{th}+1$ internal iteration, i.e., $Y^s_{k|k,l+1} = S^s_{y,k|k,l+1}(S^s_{y,k|k,l+1})^T$, $I^j_{k,l} = S^j_{i,k,l}(S^j_{i,k,l})^T$ is the information contribution matrix [28]. Get more information about the equivalence between (11) and (12) in [15, Sec. 4.3].

With the definition above, the following Theorem 1 guarantees the condition for the existence of the hybrid average consensus. First, a lemma is needed.

Lemma 1 [29]: Let $A \in R^{n\times n}$ be nonnegative. If A is row-stochastic primitive, then $\lim_{l\to\infty} A^l = 1v^T$, where v is a $n \times 1$ nonnegative column vector satisfying $1^T v = 1$.

Theorem 1: Consider a camera network with topology $G = (C, E, A)$. Suppose that the consensus weighted matrix $\Pi = \left(\pi^{sj}\right)_{N_C\times N_C}$ is primitive, then, each information pair $\left(\hat{y}^s_{k|k}, S^s_{y,k|k}\right)$ can reach a hybrid average consensus.

Proof: The proof of this theorem is provided in Appendix A. □

By Theorem 1, it is ready to introduce distributed square-root cubature information hybrid consensus filter with correlated noises (SCHF-CN) as Algorithm 1.

It is important to note that Algorithm 1 needs to choose the scalar weights ω^s_k. A reasonable choice consists in setting $\omega^s_k = N_C$, meanwhile the consensus weights are chosen so that $\pi^{s,j}_L \to 1/N_C$ as $L \to \infty$, where $\pi^{s,j}_L$ denotes the (s, j)-th element of Π^L, i.e., $\Pi^L = \left(\pi^{s,j}_L\right)_{N_C\times N_C}$ and Π^L denotes the L-th power of the consensus matrix Π. In this way, the distributed algorithm converges to the centralized algorithm when L tends to infinity. It is worth noting that, when such a choice is adopted, the results of Algorithm 1 are equivalent to SCWF proposed in [27].

While asymptotically optimal, the choice $\omega^s_k = N_C$ may have some drawbacks. For example, there may be some nodes that do not reach the consensus, when the choice of multiplication by N_C may lead these nodes to an overestimation of $\sum_{C_j\in C} I^j_k$ [16]. This situation needs to be avoided in order to preserve the consistency of each local filter. An alternative solution is to use consensus to calculate, in a distributed way, a normalization factor, which can improve the filter performance while preserving consistency of each local filter. For example, Not all cameras can observe all the targets in the area due to the limited of FoVs. For a target, when using the distributed consensus filter to estimate its state vector, it is necessary to consider whether the camera can observe the target or not. It requires us to treat the observations from different cameras differently. Now, the value of ω^s_k can be computed in the following way. It is first necessary to determine the value of $b^s_k(L)$, which means that the ratio of the number of cameras that can observe the target to the total number of cameras.

$$b^s_k(l+1) = \sum_{C_j\in N_s} \pi^{s,j} b^j_k(l), \quad l = 0, 1, \cdots, L-1 \tag{13}$$

with the initialization $b^s_k(0) = 1$ if the camera C_s can observe the target, and $b^s_k(0) = 0$ otherwise. Then, it can be seen that the choice

$$\omega^s_k = \begin{cases} 1/b^s_k(L) & \text{if } b^s_k(L) \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

Algorithm 1 Distributed Square-Root Cubature Information Hybrid Consensus Filter With Correlated Noises (SCHF-CN)

Input: $\hat{y}^s_{k|k-1}$ and $S^s_{y,k|k-1}$

1) Get measurements: z^s_k, $C_s \in C$;
2) Compute the information contribution vector i^s_k and square-root information matrix $S^s_{i,k}$ using (26) and (23) in [28]; $i^s_k = 0$ and $S^s_{y,k|k-1} = 0$ if the camera does not detect the target;
3) Initialized consensus data $i^s_{k,0} = i^s_k$, $S^s_{i,k,0} = S^s_{i,k}$, $\hat{y}^s_{k|k-1,0} = \hat{y}^s_{k|k-1}$, $S^s_{y,k|k-1,0} = S^s_{y,k|k-1}$;
4) For $l = 0, 1, \cdots, L-1$, implement following consensus steps.
   a) Broadcast information $i^s_{k,l}$, $S^s_{i,k,l}$, $\hat{y}^s_{k|k-1,l}$, $S^s_{y,k|k-1,l}$ to its neighbors $C_j \in \Omega_s$;
   b) Receive the messages $i^j_{k,l}$, $S^j_{i,k,l}$, $\hat{y}^j_{k|k-1,l}$, $S^j_{y,k|k-1,l}$, $C_j \in \Omega_s$;
   c) Fuse the information $i^j_{k,l}$ and $S^j_{i,k,l}$,
$$i^s_{k,l+1} = \sum\nolimits_{C_j \in N_s} \pi^{s,j} i^j_{k,l}$$
$$S^s_{i,k,l+1} = Tria\left(\left[\sqrt{\pi^{s,s}} S^s_{i,k,l} \sqrt{\pi^{s,1}} S^1_{i,k,l} \cdots \sqrt{\pi^{s,N_{s,E}}} S^{N_{s,E}}_{i,k,l}\right]\right)$$
   d) Meanwhile fuse the information $\hat{y}^j_{k|k-1,l}$ and $S^j_{y,k|k-1,l}$,
$$\hat{y}^s_{k|k-1,l+1} = \sum\nolimits_{C_j \in N_s} \pi^{s,j} \hat{y}^j_{k|k-1,l}$$
$$S^s_{y,k|k-1,l+1} = Tria\left(\left[\sqrt{\pi^{s,s}} S^s_{y,k|k-1,l} \sqrt{\pi^{s,1}} S^1_{y,k|k-1,l} \cdots \sqrt{\pi^{s,N_{s,E}}} S^{N_{s,E}}_{y,k|k-1,l}\right]\right)$$
5) Compute the updated information vector $\hat{y}^s_{k|k}$ and the square-root of the updated information matrix $S^s_{k|k}$,
$$\hat{y}^s_{k|k} = \hat{y}^s_{k|k-1,L} + \omega^s_k i^s_{k,L}$$
$$S^s_{y,k|k} = Tria\left(\left[S^s_{y,k|k-1,L} \quad \sqrt{\omega^s_k} S^s_{i,k,L}\right]\right)$$
6) Compute the predicted information vector $\hat{y}^s_{k+1|k}$ and the square-root of the predicted information matrix $S^s_{y,k+1|k}$ [28].

Output: $\hat{y}^s_{k+1|k}$ and $S^s_{y,k+1|k}$

has the desirable property of preserving the consistency of each local filter. In fact, the terms i^j_k and I^j_k, for $C_j \in C$, are multiplied by a weight $\omega^s_k \pi^{s,j}_L = \pi^{s,j}_L / b^s_k(L)$ which is guaranteed not to exceed 1 since, by construction, $b^s_k(L) = \sum_{C_j \in C} \pi^{s,j}_L$.

IV. STABILITY ANALYSIS

In this section, the stability of the proposed SCHF-CN algorithm is analyzed from the estimation consistency. In addition, the square-root filter and the standard filter are mathematically equivalent, and the square-root filter is used to solve the numerical difficulties due to the finite-word-length of the processor, especially in embedded systems. It is thus proved that the stability of the square-root filter algorithm can be exactly equivalent to the stability of the corresponding standard filter [30]. In order to facilitate the stability analysis of SCHF-CN, we adopt the statistical linear error propagation methodology [31]–[33] to derive a pseudo system matrix $\mathcal{F}^s_k$ and pseudo measurement matrix $\mathcal{H}^s_k$ in (7) and (8),for each camera.

A. LINEARIZATION APPROXIMATION

By the error propagation notion, the cross-covariance matrix can be obtained as follows:

$$P^s_{xz,k|k-1} = E\left[\left(x^s_k - \hat{x}^s_{k|k-1}\right)\left(z^s_k - \hat{z}^s_{k|k-1}\right)^T |Z^{k-1}\right] \triangleq (Y^s_{k|k-1})^{-1}\mathcal{H}^s_k \tag{15}$$

where Z^{k-1} is all measurements obtained before time k, $Y^s_{k|k-1}$ is the predicted information matrix in camera C_s, and $\hat{x}^s_{k|k-1}$ and $\hat{z}^s_{k|k-1}$ are predicted state and predicted measurement, respectively. Then, the pseudo measurement matrix $\mathcal{H}^s_k$ can be calculated by

$$\mathcal{H}^s_k = Y^s_{k|k-1} P^s_{xz,k|k-1} = Y^s_{k|k-1} \times \left[\frac{1}{m}\sum_{i=1}^{m} X^s_{i,k|k-1}(Z^s_{i,k|k-1})^T - \hat{x}^s_{k|k-1}(\hat{z}^s_{k|k-1})^T\right] \tag{16}$$

where $m = 2n_x$, $X_{i,k|k-1}$ and $Z_{i,k|k-1}$, $(i = 0, \cdots, m)$, are the cubature points and propagated cubature points in measurement update step of algorithm, respectively [5]. In order to obtain the pseudo system matrix $\mathcal{F}^s_k$, it is necessary to introduce the cross-covariance between the current prediction and latest previous estimate, which is calculated as follows [34], [35]:

$$P^s_{x_{k-1},x_{k|k-1}} = E\left[\left(x^s_{k-1} - \hat{x}^s_{k-1|k-1}\right) \times \left(x^s_k - \hat{x}^s_{k|k-1}\right)^T |Z^{k-1}\right] \triangleq Y^{-1}_{s,k-1|k-1}\left(\mathcal{F}^s_{k-1}\right)^T. \tag{17}$$

Via cubature calculation rules [36],

$$P^s_{x_{k-1},x_{k|k-1}} = \frac{1}{m}\sum_{i=1}^{m}\left(X^s_{i,k-1|k-1} - \hat{x}^s_{k-1|k-1}\right) \times \left(X^{*s}_{i,k|k-1} - \hat{x}^s_{k|k-1}\right)^T \tag{18}$$

where $X^s_{i,k-1|k-1}$ and $X^{*s}_{i,k|k-1}$ are the cubature points and propagated cubature points in time update step of algorithm, respectively [5]. So $\mathcal{F}^s_{k-1}$ can be approximated by

$$\mathcal{F}^s_{k-1} = (P^s_{x_{k-1},x_{k-1|k-1}})^T Y^s_{k-1|k-1}. \tag{19}$$

According to [37], in order to compensate the approximation error caused by $\mathcal{F}^s_{k-1}$ and $\mathcal{H}^s_k$, we introduce the compensation instrumental diagonal matrix

$$\alpha^s_k = diag(\alpha^s_{k,1}, \alpha^s_{k,1}, \cdots, \alpha^s_{k,n_x})$$

and

$$\beta^s_k = diag(\beta^s_{k,1}, \beta^s_{k,1}, \cdots, \beta^s_{k,n_z}).$$

Then, we can rewrite (7) and (8) as (at time k)

$$x^s_k = \alpha^s_{k-1}\mathcal{F}^s_{k-1}x^s_{k-1} + v^{*s}_{k-1} \tag{20}$$
$$z^s_k = \beta^s_k\mathcal{H}^s_k x^s_k + w^s_k. \tag{21}$$

By the above linear approximation equations, and based on the standard information filter expression [3], the corresponding modified cubature information filter is rewritten by the following:

$$\begin{aligned} Y^s_{k|k-1} &= \Big[\alpha^s_{k-1}\mathcal{F}^s_{k-1}\left(Y^s_{k-1|k-1}\right)^{-1} \\ &\quad \times \left(\alpha^s_{k-1}\mathcal{F}^s_{k-1}\right)^T + Q^*_{s,k-1}\Big]^{-1} \\ \hat{y}^s_{k|k-1} &= Y^s_{k|k-1}\left(\alpha^s_{k-1}\mathcal{F}^s_{k-1}\right)\hat{y}^s_{k-1|k-1} \\ Y^s_{k|k} &= Y^s_{k|k-1} + \left(\beta^s_k\mathcal{H}^s_k\right)^T\left(R^s_k\right)^{-1}\beta^s_k\mathcal{H}^s_k \\ \hat{y}^s_{k|k} &= \hat{y}^s_{k|k-1} + \left(\beta^s_k\mathcal{H}^s_k\right)^T\left(R^s_k\right)^{-1}z^s_k. \end{aligned} \tag{22}$$

B. CONSISTENCY OF ESTIMATES

Consistency is one of the most basic and important properties in data fusion processes [22], [35], [37], [38]. In the following, the consistency of the proposed SCHF-CN algorithm in this paper is proved based on the linear approximation model in (20) and (21).

Definition 2 [22]: Consider a random vector x. Further, let $\hat{x}$ and P be unbiased estimate of x and an estimate of the corresponding error covariance, respectively. Then, the pair $(\hat{x}, P)$ is said to be consistent if $E\left\{(x-\hat{x})(x-\hat{x})^T\right\} \leq P$.

Note that the inequality relationship between two matrices in this paper is used to indicate whether the matrix is semi-definite or not, for example, $A \geq B$ if and only if $A - B$ is a semi-definite matrix.

In words, according to Definition 2, consistency amounts to requiring that the estimated error covariance P be an upper bound (in the semi-definite sense) of the true error covariance. If one considers the information pair $(\hat{y}, Y) = (P^{-1}\hat{x}, P^{-1})$, the pair $(\hat{y}, Y)$ is said to be consistent if

$$Y \leq E^{-1}\left\{\left(x - Y^{-1}\hat{y}\right)\left(x - Y^{-1}\hat{y}\right)^T\right\} \tag{23}$$

Lemma 2 [22]: The function $\psi(\cdot)$ is monotone nondecreasing, i.e., give two positive semi-definite matrices Y_1 and Y_2 with $Y_1 \leq Y_2$ one has $0 \leq \psi(Y_1) \leq \psi(Y_2)$, where the function $\psi(\cdot)$ is determined by the first equation in (22), i.e., $Y^s_{k|k-1} = \psi\left(Y^s_{k-1|k-1}\right)$ with $\psi(Y) = \left[\alpha^s_{k-1}\mathcal{F}^s_{k-1}Y^{-1}(\alpha^s_{k-1}\mathcal{F}^s_{k-1})^T + Q^*_{s,k|k-1}\right]^{-1}$.

Theorem 2: If the initial predicted estimates $\{\hat{x}^s_{1|0}\}^{N_C}_{s=1}$ satisfies the following equation:

$$Y^s_{1|0} \leq E^{-1}\left\{\left(x_1 - \hat{x}^s_{1|0}\right)\left(x_1 - \hat{x}^s_{1|0}\right)^T\right\}, \tag{24}$$

then, for each time step k and $C_s \in C$,

$$Y^s_{k|k} \leq E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k}\right)\left(x_k - \hat{x}^s_{k|k}\right)^T\right\}.$$

That is, Algorithm 1 preserves consistency.

Proof: The proof of this theorem is provided in Appendix B. □

V. EXPERIMENTAL EVALUATION

Considering the two aspects of the system with correlated noise and consensus algorithm, a distributed square-root cubature information hybrid consensus filter with correlated noises called SCHF-CN is proposed in this paper. This algorithm not only considers the correlation between measurement noise and process noise, but also takes the advantages of hybrid consensus algorithm into account.

In order to evaluate the performance of the proposed algorithm, we compare it with other methods: DUKF [34] and DCIF [35]. Our experiments are performed on an Intel i7-7700k 4.7GHz PC with 16G memory and implemented in Matlab R2017a.

A. SIMULATION ENVIRONMENT

A target moving in a $500m \times 500m$ area where is under observation of nine cameras ($N_C = 9$) with overlapping FOVs is considered. To simplify the simulation, the FOV of each camera is assumed to be a square region of $200m \times 200m$ around the camera. At discrete time instant k, the state vector consists of the target's position (x_k, y_k), its velocity (v_x, v_y) and the time interval δ_k between the two consecutive measurements. That is: $x_k = [x_k \quad y_k \quad v_x \quad v_y \quad \delta_k]^T$.

The motion model of the targets is described by the nonlinear equation [39]:

$$x_{k+1} = \begin{pmatrix} x_k + v_{x,k}\delta_k + a_x\frac{\delta_k^2}{2} \\ y_k + v_{y,k}\delta_k + a_y\frac{\delta_k^2}{2} \\ v_{x,k} + a_x\delta_k \\ v_{y,k} + a_y\delta_k \\ \delta_k + e \end{pmatrix} \tag{25}$$

where the target acceleration (a_x, a_y) is modelled as Gaussian noise. To account for synchronisation errors among cameras, we consider a time uncertainty e, which is also assumed to be a Gaussian variable. We consider the vector (a_x, a_y, e) as the Gaussian noise vector with zero mean and covariance $Q = diag([20 \quad 20 \quad 0.2])$. In order to facilitate the calculation, it needs to be independent of the process noise v_k as the (1). Here, computing the Jacobian matrix $J_{v,k}$ of (25) with respect to (a_x, a_y, e), one get the $J_{v,k}$ as (26),

$$J_{v,k} = \begin{pmatrix} \frac{1}{2}\delta_k^2 & 0 & 0 \\ 0 & \frac{1}{2}\delta_k^2 & 0 \\ \delta_k & 0 & 0 \\ 0 & \delta_k & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{26}$$

Then, $v_k = J_{v,k}(a_x, a_y, e)^T$ and its variance is calculated by $Q_k = E\left(v_k v_k^T\right) = J_{v,k} diag\left([10\ \ 10\ \ 0.1]\right) J_{v,k}^T$. The initial speed is randomly got from the range $10 \sim 50$ units per time step and with a random direction uniformly chosen from 0 to 2π, and then the trajectory required for the simulation can be obtained according to the (25). The initial prior covariance $P_{1,s}^- = diag\left([100\ \ 100\ \ 25\ \ 25\ \ 0.01]\right)$. In addition, the initial prediction state information $\hat{x}_{1,s}^-$ is composed of the initial value of the real trajectory with a the zero mean Gaussian white noise by $P_{1,s}^-$ as its covariance.

The measurement model can be defined as

$$z_k^s = \begin{pmatrix} \gamma_k^s \\ \phi_k^s \end{pmatrix} = \begin{pmatrix} \frac{H_{11}^s x_k + H_{12}^s y_k + H_{13}^s}{H_{31}^s x_k + H_{32}^s y_k + H_{33}^s} \\ \frac{H_{21}^s x_k + H_{22}^s y_k + H_{23}^s}{H_{31}^s x_k + H_{32}^s y_k + H_{33}^s} \end{pmatrix} + w_k \tag{27}$$

where (γ_k^s, ϕ_k^s) is the pixel coordinates of the target in the image plane of camera C_s at time k. The values $H_{11}^s, \cdots, H_{33}^s$ are the elements of *Homography*, w_k is the measurement noise, which is considered to be white Gaussian noise. Since the *Homography* matrix is used to convert the different observation planes of cameras to the reference plane. It can be seen from the literature [27] that the same *Homography* matrix using in different cameras does not affect the simulation results. To simplify the experimental design, the *Homography* matrix values of each camera are taken from the camera C_6 of the APIDIS dataset [40] whose values are:

$$H_s = \begin{pmatrix} 1930.8939 & -89.8033 & -2393800 \\ 117.2530 & 91.8121 & 1022700 \\ 0.3485 & -0.8720 & 1971.8862 \end{pmatrix} \tag{28}$$

According to III-A, rewrite the system equation as

$$\begin{aligned} x_{k+1} &= f\left(x_k\right) + v_k + T_k^s\left[z_k^s - h\left(x_k\right) - w_k^s\right] \\ &= \begin{bmatrix} x_k + v_{x,k}\delta_k \\ y_k + v_{y,k}\delta_k \\ v_{x,k} \\ v_{y,k} \\ \delta_k \end{bmatrix} + T_k^s\left[z_k^s - \begin{pmatrix} \frac{H_{11}^s x_k + H_{12}^s y_k + H_{13}^s}{H_{31}^s x_k + H_{32}^s y_k + H_{33}^s} \\ \frac{H_{21}^s x_k + H_{22}^s y_k + H_{23}^s}{H_{31}^s x_k + H_{32}^s y_k + H_{33}^s} \end{pmatrix}\right] \\ &\quad + v_k - T_k^s w_k^s \\ &= f^*\left(x_k\right) + v_k^* \end{aligned} \tag{29}$$

At time step $k+1$, the observed value z_k^s from the camera C_s is a known value, so all items of the function $f^*(x_k)$ are known. Note that the second term $f^*(x_k)$ will be set to zero when the camera does not detect the target, i.e. $f^*(x_k) = f(x_k)$. In addition, the process noise $v_k^* = v_k - T_k^s w_k^s$ in (29) is not related to the measured noise w_k^s.

Let φ_k^s $(s = 1, \cdots, 9)$ are the correlation matrix between v_k and w_k^s, i.e. $w_k^s = \varphi_k^s v_k$. Thus,

$$D_k^s = E\left\{v_k (w_k^s)^T\right\} = Q_k (\varphi_k^s)^T \tag{30}$$

The covariance of the measured noise R_k^s is

$$R_k^s = E\left\{w_k^s (w_k^s)^T\right\} = \varphi_k^s Q_k (\varphi_k^s)^T \tag{31}$$

In addition, according to (6), the covariance of the new process noise is

$$Q_k^* = E\left\{v_k^* (v_k^*)^T\right\} = Q_k - D_k^s \left(R_k^s\right)^{-1} \left(D_k^s\right)^T \tag{32}$$

In order to establish a uniform simulation environment, we set $\varphi_k^s = 0.7 \times [1, 0, 0, 0, 0; 0, 1, 0, 0, 0]$. In addition, according to (5), (31) and (32), one can know $T_k^s = \left(\varphi_k^s\right)^{-1}$. It should be noted that φ_k^s is not a square matrix, so the superscript -1 in the previous equation is the pseudo-inverse operation.

In this paper, we perform the experiments for a sparse connectivity network with a low average network degree equal to 2 (dotted lines indicate network links in Fig. 2). To facilitate the display, Fig. 2 only shows the Fovs of C_1, C_3, C_5, C_7, C_8, the other are similar. The consensus weight π^{sj} can be calculated through a well-known Metropolis weights rule

$$\pi^{sj} = \begin{cases} \frac{1}{1 + max\left\{\Delta_s, \Delta_j\right\}}, & if\ \ \left\{C_s, C_j\right\} \in E \\ 1 - \sum_{\{C_s, C_j\} \in E} \pi^{sj}, & if\ \ C_s = C_j \\ 0, & otherwise \end{cases} \tag{33}$$

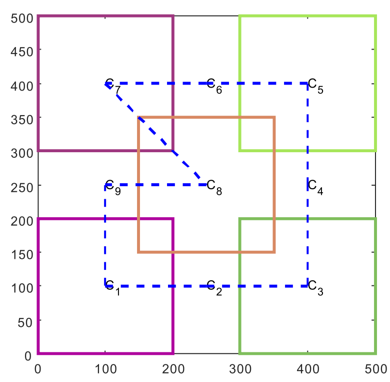


FIGURE 2. **Sparse connectivity of the network.**

B. SIMULATION RESULTS

In the following, we will compare the proposed SCHF-CN algorithm with DUKF and DCIF in two scenarios: without correlation noise ($D_k^s = 0$) and with correlation noise ($D_k^s \neq 0$). The simulation results are averaged over 50 Monte Carlo simulation runs.

1) WITHOUT CORRELATION NOISE

If the process noise is not correlated with the measurement noise, then $D_k^s = 0$. In this experiment, the measurement noise is considered to be Gaussian with zero mean and variance $R = diag([5 \quad 5])$, and $\varphi_k^s \in R^{2\times 5}$ is an all-zero matrix. The simulation results are shown in Fig 3 (Fig 3(b) zooms in Figure 3(a) with focus on SCHF-CN and DCIF.). It can be seen from the figure that the performance of the hybrid consensus algorithm proposed in this paper is better than CI consensus-based algorithms (DCIF, DUKF). From the previous theoretical analysis, we can see that the SCHF-CN algorithm combines the advantages of CM and CI algorithms, so the performance of this algorithm is improved. In addition, in the experiment, the tracking error of the DUKF algorithm is too large, which is not suitable for the target tracking in camera networks. Although the SCHF-CN algorithm can get the best tracking results, the communication cost in each iteration is more than the DCIF algorithm. Because the SCHF-CN needs to broadcast 5 vectors ($i_{k,l}^j$, $S_{i,k,l}^j$, $\hat{y}_{k|k-1,l}^j$, $S_{y,k|k-1,l}^j$, ω_k^s), while the DCIF only needs to broadcast 2 vectors ($\hat{y}_{k|k-1,l}^j$, $Y_{k|k-1,l}^j$). But the proposed algorithm has high tracking accuracy. Therefore, it needs to be considered comprehensively about the algorithm which is adopted within a specific scenario.

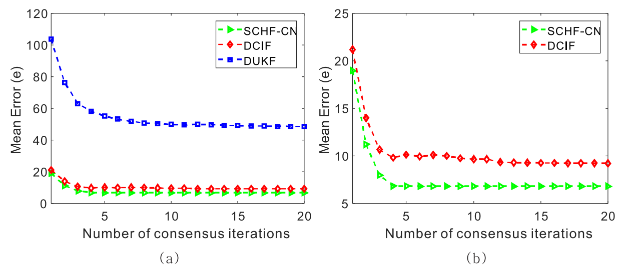


FIGURE 3. The mean errors and the variation of the estimation errors about three algorithms without correlation noise.

2) WITH CORRELATION NOISE

If the process noise is correlated with the measurement noise, then $D_k^s \neq 0$. At this time, only the SCHF-CN algorithm can handle the correlation noise. The final simulation results are shown in Fig.4. As shown in Fig.4(a), when correlated noise is present, the performance of SCHF-CN algorithm is the best. Although the SCHF-CN algorithm can deal with the correlation noise, due to the limited FoV of the camera, the target may not be observed by camera C_s, i.e., z_k^s is unknown in (29), which affects the implementation of the algorithm. And then it can destroy the performance of the algorithm.

3) DIFFERENT CORRELATION MATRICES

In above experiments, in order to make the experimental method simple, the correlation matrix φ_k^s is taken a fixed value. In order to verify the stability of each algorithm with different φ_k^s, 10 random correlation matrices are randomly generated using the $\varphi_k^s = rand(2, 5)$ function, and then 20 trajectories are randomly generated via (25) for each correlation matrix, meanwhile 10 observations are randomly generated via (28) for each trajectory. In this experiment, the number of iterations of the consensus algorithm is fixed to 5. Finally the mean error of each algorithm under each correlation matrix is calculated, the results shown in Fig. 5(a). The abscissa of Fig.5(a) is the sequence number of the correlation matrix. Since the correlation matrix is randomly generated, its sequence number is independent of the value of the correlation matrix. In addition, because the final mean error is the result of averaging the simulation results, the mean error of each algorithm is basically same at different φ_k^s. In order to verify the change rule of mean error with the increase of correlation noise, the simulation is designed as follows: set $\varphi_k^s = 5 \times [1, 0, 0, 0, 0; 0, 1, 0, 0, 0]$, then multiply it by 1 to 10. The results shown in Fig.5(b). In the figure, the the abscissa is the multiplying factor. It can be seen that the mean error of the three algorithms increases with the increase of the correlation noise, and trend of change is basically same. What's more, the SCHF-CN algorithm proposed in this paper has the best results in most cases.

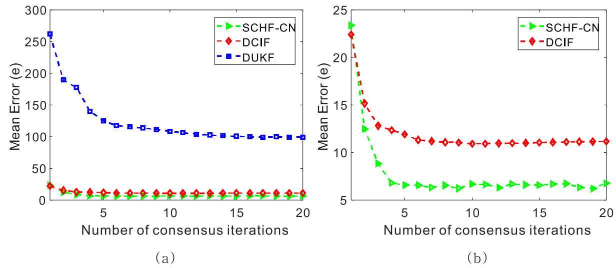


FIGURE 4. The mean errors and the variation of the estimation errors about three algorithms with correlation noise.

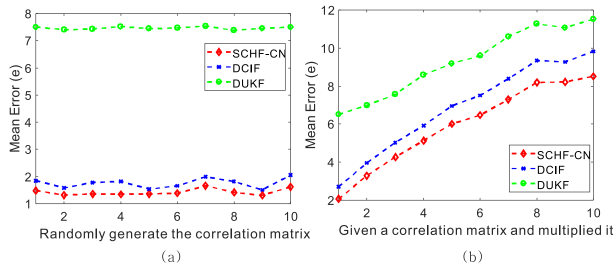


FIGURE 5. The mean errors and the variation of the estimation errors about three algorithms with correlation noise.

VI. CONCLUSION

This paper presents a square-root cubature information hybrid consensus filter with correlated noise and its application in camera networks. Firstly, a de-correlation operation for the system with correlation noise is carried out, which makes the measurement noise and process noise irrelevant in the new system equation. Then, according to the classical square-root cubature information filter, a hybrid consensus-based filter with correlated noise is proposed.

What's more, the stability of the proposed algorithm is proved by the consistency of estimates. Finally, it is verified that the proposed algorithm in this paper is superior to other algorithms via the applications about the target tracking in camera networks. The final simulation results show that the proposed algorithm can achieve best results whether the measurement noise and process noise are correlated or not.

In our future work, we will consider to apply the proposed algorithm to other applications, e.g., bearings-only tracking system, and investigate new consensus approaches to make the performance of estimation more accurate.

APPENDIX A
PROOF OF THEOREM 1

Proof: By (13), (14) and CI algorithm, it is fairly easy to get that all ω_k^s ($C_s \in N_S$) tend to a average value ω_k^* as $l \to \infty$. Then, the (10) can be rewritten as

$$\hat{y}^s_{k|k,l+1} = \sum_{C_j \in N_s} \pi^{s,j}(\hat{y}^j_{k|k-1,l} + \omega_k^* \hat{i}^j_{k,l})$$

Now, denote $\hat{y}^s_{k|k} = \hat{y}^s_{k|k-1} + \omega_k^* \hat{i}^s_k$. In addition, all the vectors can be vertically concatenated into a single column vector, i.e.,

$$\hat{y}_{k|k} = col\left(\hat{y}^s_{k|k-1} + \omega_k^* \hat{i}^s_k\right)$$

with $C_s \in N_s$. Hence, (10) can be written as

$$\begin{aligned}\hat{y}_{k|k,l+1} &= \left(\Pi \otimes I_{n_x}\right)\left(\hat{y}_{k|k,l}\right)\\ &= \left(\Pi \otimes I_{n_x}\right)\cdots\left(\Pi \otimes I_{n_x}\right)\left(\Pi \otimes I_{n_x}\right)\hat{y}_{k|k,0}\\ &= \left(\Pi^{l+1} \otimes I_{n_x}\right)\hat{y}_{k|k,0}.\end{aligned}$$

As defined before, $\pi^{sj} \geq 0$ and $\sum_{C_j \in N_s} \pi^{sj} = 1$, hence, it is naturally matrix Π is row-stochastic, and Π is assumed as primitive. According to Lemma 1, we have

$$\lim_{l \to \infty} \Pi^{l+1} = 1v^T$$

where v is a column vector with with $v = \left[v_1, v_2, \cdots, v_{N_C}\right]^T$. When $l \to \infty$, $\hat{y}_{k|k,l+1} = \left(1v^T \otimes I_{n_x}\right)\hat{y}_{k|k,0}$, that is to say,

$$\begin{aligned}\hat{y}_{k|k,l+1} &= v_1\hat{y}^1_{k|k,0} + v_2\hat{y}^2_{k|k,0} + \cdots + v_{N_C}\hat{y}^{N_C}_{k|k,0}\\ &= \hat{y}^*_{k|k}\end{aligned}$$

In addition, $S^s_{y,k|k}$ is the square-root form of $Y^s_{k|k}$, and (11) and (12) are equivalent. In the same way, when $l \to \infty$, $S^s_{y,k|k}$ has the similar result. The proof is now completed. □

APPENDIX B
PROOF OF THEOREM 2

Proof: The proof process is given by mathematical induction. To this end, assume that at time k the following equation holds,

$$Y^s_{k|k-1} \leq E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k-1}\right)\left(x_k - \hat{x}^s_{k|k-1}\right)^T\right\} \tag{34}$$

for any node $C_s \in C$.

The choice suggested in (14) for the weight ω_k^s gives following result:

There exist a positive scalar $\underline{\omega}$ such that

$$1 \leq \underline{\omega} \leq \omega_k^s, \tag{35}$$

with, $C_s \in N_C$. By recalling the identity in (12), it can be seen that (35) implies that

$$\begin{aligned}Y^s_{k|k,1} &= \sum_{C_j \in N_s} \pi^{s,j}\left[Y^j_{k|k-1,0} + \omega_k^s I^j_{k,0}\right]\\ &\geq \sum_{C_j \in N_s} \pi^{s,j}\left[Y^j_{k|k-1,0} + \underline{\omega} I^j_{k,0}\right]\end{aligned} \tag{36}$$

Now, denote

$$Y^s_{k|k,0} = Y^s_{k|k-1,0} + \omega_k^{\star} I^s_{k,0} \tag{37}$$

with $C_s \in N_C$ and $\omega_k^{\star} \geq \underline{\omega}$.

By the initialization of Algorithm 1 and (36), (34) implies,

$$\begin{aligned}&E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k,0}\right)\left(x_k - \hat{x}^s_{k|k,0}\right)^T\right\}\\ &\quad = E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k-1}\right)\left(x_k - \hat{x}^s_{k|k-1}\right)^T\right\} + \omega_k^{\star} I^s_k\\ &\quad \geq Y^s_{k|k-1} + \underline{\omega} I^s_k \geq Y^s_{k|k-1} + I^s_k = Y^s_{k|k,0}\end{aligned}$$

with $Y^s_{k|k-1,0} = Y^s_{k|k-1}$ and $I^s_{k,0} = I^s_k = S^s_{i,k} \times (S^s_{i,k})^T$.

Because the HCMCI approach is applied to Algorithm 1 and $I^s_k = Y^s_{k|k-1} P^s_{xz,k|k-1} (R^s_k)^{-1} (Y^s_{k|k-1} P^s_{xz,k|k-1})^T$ [28], we can carried out a CI approach with the noise covariance matrices R^s_k replaced by $R^s_k/\omega_k^{\star}$. Then, Algorithm 1 also has the nature of CI approach, that is satisfied Covariance Intersection fusion rule. Since the Covariance Intersection fusion rule preserves the consistency property [41], i.e.

$$E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k,l}\right)\left(x_k - \hat{x}^s_{k|k,l}\right)^T\right\} \geq Y^s_{k|k,l}$$

implies

$$E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k,l+1}\right)\left(x_k - \hat{x}^s_{k|k,l+1}\right)^T\right\} \geq Y^s_{k|k,l+1}$$

for any $l = 0, 1, \cdots, L-1$, that is to say, at time step k, $E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k}\right)\left(x_k - \hat{x}^s_{k|k}\right)^T\right\} \geq Y^s_{k|k}$ holds with $\hat{x}^s_{k|k} = \hat{x}^s_{k|k,L}$ and $Y^s_{k|k} = Y^s_{k|k,L}$. Further, according Lemma 2, it is immediate to see that

$$\begin{aligned}Y^s_{k+1|k} &= \psi\left(Y^s_{k|k}\right)\\ &\leq \psi\left\{E^{-1}\left\{\left(x_k - \hat{x}^s_{k|k}\right)\left(x_k - \hat{x}^s_{k|k}\right)^T\right\}\right\}\\ &= E^{-1}\left\{\left(x_{k+1} - \hat{x}^s_{k+1|k}\right)\left(x_{k+1} - \hat{x}^s_{k+1|k}\right)^T\right\}\end{aligned}$$

The proof is completed since the initial predicted estimates $\{\hat{x}^s_{1|0}\}_{s=1}^{N_C}$ are consistent. □

REFERENCES

[1] 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. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1397–1410, Jul. 2016.

[2] R. Zhan and J. Wan, "Neural network-aided adaptive unscented Kalman filter for nonlinear state estimation," *IEEE Signal Process. Lett.*, vol. 13, no. 7, pp. 445–448, Jul. 2006.

[3] A. G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, 1st ed. Boca Raton, FL, USA: CRC Press, 1998.

[4] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[5] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[6] R. F. Souto, J. Y. Ishihara, and G. A. Borges, "A robust extended Kalman filter for discrete-time systems with uncertain dynamics, measurements and correlated noise," in *Proc. Amer. Control Conf.*, St. Louis, MO, USA, Jun. 2009, pp. 1888–1893.

[7] J. Chen and L. Ma, "Particle filtering with correlated measurement and process noise at the same time," *IET Radar, Sonar Navigat. Netw.*, vol. 5, no. 7, pp. 726–730, 2011.

[8] D. Zuo, C. Han, R. Wei, and Z. Lin, "Synchronized multi-sensor tracks association and fusion," in *Proc. 4th Int. Conf. Inf. Fusion*, Montréal, QC, Canada, Aug. 2001, pp. 1–6.

[9] Y. Bar Shalom, X. R. Li, and T. Kirubarajan, *Extensions of Discrete-Time Linear Estimation*. New York, NY, USA: Wiley, 2002, pp. 319–339.

[10] E. Song, Y. Zhu, J. Zhou, and Z. You, "Optimal Kalman filtering fusion with cross-correlated sensor noises," *Automatica*, vol. 43, no. 8, pp. 1450–1456, 2007.

[11] J. Xu, G. M. Dimirovski, Y. Jing, and S. Chao, "UKF design and stability for nonlinear stochastic systems with correlated noises," in *Proc. 46th IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 6226–6231.

[12] X. Wang, Y. Liang, Q. Pan, and F. Yang, "A Gaussian approximation recursive filter for nonlinear systems with correlated noises," *Automatica*, vol. 48, no. 9, pp. 2290–2297, 2012.

[13] G. Chang, "Marginal unscented Kalman filter for cross-correlated process and observation noise at the same epoch," *IET Radar, Sonar Navigat. Netw.*, vol. 8, no. 1, pp. 54–64, 2014.

[14] Q. Ge, D. Xu, and C. Wen, "Cubature information filters with correlated noises and their applications in decentralized fusion," *Signal Process.*, vol. 94, pp. 434–444, Jan. 2014.

[15] Y. Chen and Q. Zhao, "A novel square-root cubature information weighted consensus filter algorithm for multi-target tracking in distributed camera networks," *Sensors*, vol. 15, no. 5, pp. 10526–10546, 2015.

[16] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1410–1415, May 2015.

[17] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 622–633, May 2008.

[18] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. 48h IEEE Conf. Decis. Control (CDC) Jointly 28th Chin. Control Conf.*, Shanghai, China, Dec. 2009, pp. 7036–7042.

[19] F. S. Cattivelli and A. H. Sayed, "Distributed nonlinear Kalman filtering with applications to wireless localization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing.*, Dallas, TX, USA, Mar. 2010, pp. 3522–3525.

[20] S. J. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection," in *Electrical Engineering and Applied Signal Processing Series*, M. E. Liggins, D. L. Hall, and J. Llinas, Eds., 2nd ed. New York, NY, USA: CRC Press, 2009, pp. 319–345.

[21] 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.

[22] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based algorithms for distributed filtering," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Maui, HI, USA, Dec. 2012, pp. 794–799.

[23] G. Liu and G. Tian, "Square-root sigma-point information consensus filters for distributed nonlinear estimation," *Sensors*, vol. 17, no. 4, p. 800, 2017.

[24] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Maui, HI, USA, Dec. 2012, pp. 2732–2737.

[25] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli, "Real-time discriminative background subtraction," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1401–1414, May 2011.

[26] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[27] Y. Chen, Q. Zhao, and R. Liu, "A novel square-root cubature information weighted consensus filter algorithm for distributed camera networks," *Acta Electronica Sinica*, vol. 44, no. 10, pp. 2335–2343, 2016.

[28] I. Arasaratnam, "Sensor fusion with square-root cubature information filtering," *Intell. Control Automat.*, vol. 4, no. 1, pp. 11–17, 2013.

[29] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[30] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[31] T. Lefebvre, H. Bruyninckx, and J. D. Schuller, "Comment on 'A new method for the nonlinear transformation of means and covariances in filters and estimators' [with authors' reply]," *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1406–1409, Aug. 2002.

[32] G. Sibley, G. S. Sukhatme, and L. H. Matthies, "The iterated sigma point Kalman filter with applications to long range stereo," in *Proc. Robot., Sci. Syst.*, Pennsylvania, PA, USA, Aug. 2006, pp. 235–244.

[33] D.-J. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering," *IEEE Signal Process. Lett.*, vol. 15, pp. 861–864, 2008.

[34] W. Li, G. Wei, F. Han, and Y. Liu, "Weighted average consensus-based unscented Kalman filtering," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 558–567, Feb. 2016.

[35] Q. Chen, W. Wang, C. Yin, X. Jin, and J. Zhou, "Distributed cubature information filtering based on weighted average consensus," *Neurocomputing*, vol. 243, pp. 115–124, Jun. 2017.

[36] K. Pakki, B. Chandra, D. W. Gu, and I. Postlethwaite, "Cubature information filter and its applications," in *Proc. Amer. Control Conf.*, San Francisco, CA, USA, Jul. 2011, pp. 3609–3614.

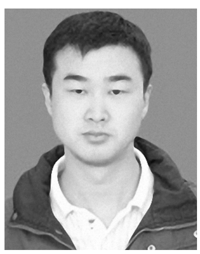
[37] K. Xiong, H. Y. Zhang, and C. W. Chan, "Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, 2006.

[38] S. Wang and W. Ren, "On the consistency and confidence of distributed dynamic state estimation in wireless sensor networks," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Osaka, Japan, Dec. 2015, pp. 3069–3074.

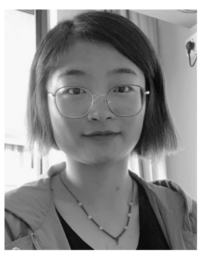
[39] S. Katragadda, J. C. SanMiguel, and A. Cavallaro, "Consensus protocols for distributed tracking in wireless camera networks," in *Proc. 17th Int. Conf. Inf. Fusion*, Salamanca, Spain, Jul. 2014, pp. 1–8.

[40] F. Chen, D. Delannay, and C. De Vleeschouwer, "An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study," *IEEE Trans. Multimedia*, vol. 13, no. 6, pp. 1381–1394, Dec. 2011.

[41] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, Albuquerque, NM, USA, Jun. 1997, pp. 2369–2373.



YANMING CHEN received the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, in 2017. He was a Research Assistant with the Institute of Computing Technology, Chinese Academy of Sciences, from 2010 to 2013. He is currently an Assistant Professor with the School of Computer Science and Technology, Anhui University, China. His current research interests include distributed algorithms, edge computing, and sensor networks.



JIAHUI WANG received the B.S. degree in computer science from the Anhui University of Chinese Medicine, in 2017. She is currently pursuing the M.S. degree in computer science with Anhui University. Her current research interests include distributed state estimation and particle filter.