Sensors **2015**, *15*, 10526-10546; doi:10.3390/s150510526

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

A Novel Square-Root Cubature Information Weighted Consensus Filter Algorithm for Multi-Target Tracking in Distributed Camera Networks

Yanming Chen * and Qingjie Zhao

Beijing Key Laboratory of Intelligence Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China; E-Mail: zhaoqj@bit.edu.cn

* Author to whom correspondence should be addressed; E-Mail: cym@bit.edu.cn; Tel.: +86-187-0168-7173.

Academic Editor: Y. Ahmet Sekercioglu

Received: 2 March 2015 / Accepted: 27 April 2015 / Published: 5 May 2015

Abstract: This paper deals with the problem of multi-target tracking in a distributed camera network using the square-root cubature information filter (SCIF). SCIF is an efficient and robust nonlinear filter for multi-sensor data fusion. In camera networks, multiple cameras are arranged in a dispersed manner to cover a large area, and the target may appear in the blind area due to the limited field of view (FOV). Besides, each camera might receive noisy measurements. To overcome these problems, this paper proposes a novel multi-target square-root cubature information weighted consensus filter (MTSCF), which reduces the effect of clutter or spurious measurements using joint probabilistic data association (JPDA) and proper weights on the information matrix and information vector. The simulation results show that the proposed algorithm can efficiently track multiple targets in camera networks and is obviously better in terms of accuracy and stability than conventional multi-target tracking algorithms.

Keywords: cubature Kalman filter; information filter; consensus algorithm; multi-target tracking; distributed camera networks

1. Introduction

With the rapid development of image processing, sensor and semiconductor technology, the availability of inexpensive hardware, such as CMOS cameras, that are able to ubiquitously capture video content from the environment has fostered the development of camera networks [1]. Cameras have been widely used in smart homes, wide-area surveillance, intelligent transportation, medical care, industrial control, *etc.*

Multiple cameras can cover a large area, communicate with each other through the network and then fuse all of their measurements to achieve robust scene understanding. However, factors, such as weather, illumination and shadow, make the measurements suffer noise easily. At the same time, there are multiple targets in the scene, which increases the difficulty of targets tracking. In this paper, we focus on the problem of tracking multi-targets through a camera network. In many application scenarios of camera networks, the observation is a nonlinear function of the target state. Consequently, we propose a novel algorithm for these complicated application scenarios.

A camera network is a set of resource-constrained camera-equipped sensor nodes that are spread over a large area. The limit of the centralized architecture is obvious. When there are large volumes of data that need to be transmitted, processed and interpreted by resource-constrained nodes to deliver to the fusion center, the network may easily fail because of the energy consumption and communication burden.

One way of addressing this issue is through the novel paradigm of distributed algorithms. Recently, distributed algorithms have witnessed a surge in interest that has enabled a wide range of cooperation and information fusion in bandwidth-limited sensor networks. They are advantageous for target tracking in camera networks due to their scalability and high fault tolerance [2,3].

In a distributed estimation scheme, the system must adopt certain strategies to share information. In recent years, many researchers have proposed linear consensus protocols to deal with this problem through multiple iterations of communication between the local node and its neighboring nodes. For example, Olfati-Saber *et al.* [4] provided a theoretical framework for consensus and cooperation in multi-agent systems. In their paper, they made a detailed analysis of a consensus algorithm for multi-agent networked systems with an emphasis on the role of directed information flow, robustness to changes in the network topology due to link/node failures, time delays and performance guarantees. Ren *et al.* [5] considered the problem of information consensus among multiple agent exchange with dynamically changing interaction topologies and gave conditions for asymptotic consensus under dynamically changing interaction topologies and the weighting factors using update schemes.

Combining with the above-mentioned consensus algorithm and then using a filter algorithm, such as Kalman filter, one can achieve the goal of target tracking. Olfati-Saber introduced a novel distributed Kalman consensus filtering (KCF) algorithm for sensor networks [6]. The KCF algorithm works under the assumption that every sensor has the ability to sense all targets. However, in a realistic camera network, a target could usually be seen by none or only a few of the cameras. In [7], Olfati-Saber *et al.* considered the case mentioned above. However, the solution is a hybrid P2P/hierarchical architecture, not fully distributed and not suitable for large-scale networks. Kamal *et al.* [2] proposed an information weighted consensus filter (IWCF) to deal with this problem by proper weights on the prior state and the measurement information. In camera networks, the measurement model does not evolve linearly. Hence,

tracking algorithms depending on linear filters, such as the traditional Kalman filter and the information filter, cannot be applied. Katragadda *et al.* [8] proposed two consensus-based distributed algorithms for nonlinear systems using the extended information filter (EIF). However, this filter adopts multivariate Taylor series expansions to linearize a model. The accuracy may not meet the requirements when they are used in the case of camera networks.

To solve these problems, this paper proposes a novel consensus filter based on the square-root cubature Kalman filter (SCKF) [9]. The SCKF adopts a third-degree spherical-radial cubature rule that provides a set of cubature points scaling linearly with the state-vector dimension. The SCKF can provide a robust and systematic solution for high-dimensional nonlinear filtering problems. Meanwhile, compared with the unscented Kalman filter (UKF) [10], the SCKF can preserve two properties of the error covariance matrix: symmetry and positive definiteness in each update cycle [9]. In the UKF, due to errors introduced by arithmetic operations performed on finite word-length digital computers, these two properties are often lost.

One advantage of the information filter over the Kalman filter arises from its natural fit for multi-agent problems. Multi-agent problems often involve the integration of sensor data collected decentrally. Such integration is commonly performed using Bayes' rule. When represented in logarithmic form, Bayes' rule becomes an addition. Information integration is achieved by summing up information from multiple sensors. Addition is commutative. Because of this, information filters often integrate the information in an arbitrary order, with arbitrary delays and in a completely decentralized manner [11]. In this paper, we use the information form of SCKF, which is called the square-root cubature information filter (SCIF) [12].

Multi-target tracking is the combination of data association and estimation. However, the above-mentioned methods do not consider the measurement-to-track association. Among many algorithms that are available for data association, the multiple hypothesis tracking (MHT) [13] and joint probabilistic data association (JPDA) [14] are two popular schemes. JPDA achieves reasonable results at a much lower computational cost than MHT and can be easily integrated into a distributed system.

The main contribution of this paper is proposing data association with a square-root cubature information filter, taking special care of the issues of nonlinearity and finite word-length digital computers and using the proposed algorithm to track multi-targets in a camera network. In Section 2 the state-of-the-art in distributed multi-target tracking in camera networks is described. Section 3 presents preliminaries for this paper, such as the model, average consensus and JPDA. In Section 4, the distributed square-root cubature information weighted consensus filter (DSCIWCF) is proposed. We describe the JPDA with DSCIWCF for the multi-target tracking algorithm, called the multi-target square-root cubature information weighted consensus filter (MTSCF), in Section 5. In Section 6, the proposed method is compared against others experimentally. The simulation results show that the proposed algorithm can efficiently track multiple targets in camera networks. Finally, we will give the conclusion of this paper in Section 7.

2. Related Work

This section discusses consensus-based distributed multi-target tracking in camera networks, focusing on the problems of nonlinearity, redundancy and robustness.

There are many research papers on multi-target tracking under sensor networks [15–20]. However, most of these methods do not consider the problem of naive nodes [2] and numerical difficulties resulting from the finite word-length of computers. Now that computers have become so much more capable, we do not have to worry about numerical problems as before. Nevertheless, numerical issues still arise in finite-word-length implementations of algorithms, especially in sensor networks.

In [15], a distributed data association for multi-target tracking in sensor networks was proposed by Sandell *et al.* In their paper, they considered that each sensor node can make noisy measurements of the target state. In this situation, data association techniques must be employed. Therefore, they used the JPDA algorithm to deal with the data association. Although their proposed method is distributed, their method is based on KCF, which is used in a linear system and does not consider naive nodes.

In [18], Roy-Chowdhury *et al.* extended the method proposed in [15] to deal with nonlinear problems. Although it can be used in nonlinear camera networks, their method is based on EKCF, and thus, naive nodes have not been considered.

Kamal proposed extended multi-target information consensus to deal with the problems of nonlinearity and naive nodes [20]. Their method is based on IWCF [2], which is more robust and accurate than the KCF algorithm. However, their method has the problem of numerical difficulties mentioned above in resource-constrained camera networks.

As described above, this paper uses JPDA with the SCIF-based tracking algorithm at each camera node to track multi-targets in a camera network. Our algorithm can not only overcome the numerical difficulties mentioned above, but also gets much more accurate results at the same time.

3. Basic Theories

3.1. System Model

The general nonlinear system model for camera networks is the form:

$$x_{k,i}^{j} = f(x_{k-1,i}^{j}) + v_{k-1,i}^{j} \tag{1}$$

$$z_{k,i}^{j} = h(x_{k,i}^{j}) + w_{k,i}^{j} \tag{2}$$

where the system equation $f(\cdot)$ and the measurement equation $h(\cdot)$ are time-varying nonlinear functions. At time k, $x_{k,i}^{j} \in \mathbf{R}^{n_x}$ is the state vector of the j-th target. Each camera C_i gets $m_i(k)$ measurements denoted as $\{z_{k,i}^{j}\}_{j=1}^{m_i(k)}$, and $z_{k,i}^{j} \in \mathbf{R}^{n_z}$ is the nonlinear measurement from the j-th target measured by the node C_i at time k. Cameras do not know the relationship between measurements and targets. That is to say, they do not know which measurement is generated from which target. $v_{k-1,i}^{j} \in \mathbf{R}^{n_x}$ is the process noise of the node C_i on time $k-1$, $w_{k,i}^{j} \in \mathbf{R}^{n_z}$ is the measurement noise of node C_i at time k. The noise sequences v_{k-1}^{i} and w_{k}^{i} are assumed to be independent and white with $v_{k-1}^{i} \sim \mathrm{N}(0, Q_{k-1}^{i})$ and $w_{k}^{i} \sim \mathrm{N}(0, R_{k}^{i})$, respectively.

Given a camera network with N_C cameras, there are no specific assumptions about the overlap among the FOVs of these cameras. In the FOV, there are N_T moving targets. In this paper, we assume that all cameras have been calibrated, so we can get the target position corresponding to the same reference plane. The communication in the network can be represented using an undirected connected

graph $G(\tau) = (C, E(\tau), A(\tau))$ [21,22]. The set of vertices $C = \{C_1, C_2, \cdots, C_{N_C}\}$ represents the cameras. The set $E \subseteq C \times C$ contains the edges of the graph, which represents the available communication channels between different cameras. $A(\tau) = [a_{ij}]_{N_C \times N_C}$ is an adjacency matrix, which is a symmetric $01-$matrix. Because the graph has no loops, the diagonal entries of $A(\tau)$ are zero ($a_{ii} = 0, i = 1, \cdots, N_C$). $\Omega_i = \{j \in C \mid (i,j) \in E\}$ is an adjacency set of node C_i. (i,j) represents the direct communication channel between node C_i and node C_j. The degree of node C_i is the number of its neighbors $\triangle_i = \sum_j a_{ij}$. The degree matrix is an $N_C \times N_C$ matrix defined as $\Delta(\tau) = diag\{A(\tau) \cdot \mathbf{1}\}$. The Laplacian of graph $G(\tau)$ is defined by $L(\tau) = \Delta(\tau) - A(\tau)$.

In this paper, we use the "+" superscript to denote the *a posteriori* estimate and the "–" superscript to denote the *a priori* estimate. For example, $\hat{x}_{k,i}^{j-}$ (and its covariance $P_{k,i}^{j-}$) represents the prior/predicted state estimate (and covariance) of $x_{k,i}^{j}$.

3.2. Average Consensus

To compute the average, average consensus [23,24] is a popular distributed algorithm. Suppose, each node i holds an initial scalar value $a_i(0) \in \mathbf{R}$, and $\mathbf{a}(0) = \{a_i\}_{i=1}^{N_C}$ denotes the vector of the initial node values on the network. We are interested in computing the average of the initial values, $\frac{1}{N_C}\sum_{i=1}^{N_C} a_i$, via a distributed algorithm, in which the nodes only communicate with their neighbors.

In the average consensus algorithm, at the beginning of iteration τ, a node C_i sends its previous state $a_i(\tau - 1)$ to its direct network neighbors $C_j \in \Omega_i$ and also receives the neighbors' previous states $a_j(\tau - 1)$. Then, the iterative form of the average consensus algorithm can be stated as follows in discrete-time:

$$a_i(\tau) = a_i(\tau - 1) + \epsilon \sum_{j \in \Omega_i} (a_j(\tau - 1) - a_i(\tau - 1)) \tag{3}$$

By several iterations, a consensus is asymptotically reached for all initial states. The rate parameter ϵ should be chosen in $0 \sim 1/\Delta_{max}$, where Δ_{max} is the maximum degree of the network graph G. Choosing a larger value of ϵ will result in faster convergence, but choosing values equal or more than $1/\Delta_{max}$ will render the algorithm unstable. The paper [24] provided a good choice of ϵ using Metropolis weights. The Metropolis weight matrix is defined as:

$$W_{ij} = \begin{cases} \frac{1}{1+max\{\Delta_i, \Delta_j\}}, & \text{if} \quad \{i,j\} \in E \\ 1 - \sum_{\{i,k\} \in E} W_{ik}, & \text{if} \quad i = j \\ 0, & otherwise \end{cases} \tag{4}$$

Arranging the local consensus states into the vector $\mathbf{a}[\tau] = [a_1^T[\tau], \cdots, a_{N_C}^T[\tau]]$, the update Equation (3) can be written in the matrix form as:

$$\mathbf{a}[\tau + 1] = (\Psi[\tau] \otimes \mathbf{I})\mathbf{a}[\tau] \tag{5}$$

where $\Psi[\tau] = \mathbf{I} - \epsilon \mathbf{L}(\tau)$, $\mathbf{I}$ is the appropriate size identity matrix and $\otimes$ denotes the matrix Kronecker product; $\Psi[\tau]$ is a stochastic matrix.

3.3. Joint Probabilistic Data Association

In the real world, in addition to the data originating from the target, a set of measurements are clutter, which correspond to no targets. A direct measurement to target assignment may lead to poor performance. Thus, a data association algorithm is needed. In this paper, we use JPDA for data association [14,15]. Here, we briefly review this algorithm.

The idea of JPDA is to compute the smoothing property of expectations. In other words, the conditional mean of the state is obtained by averaging over all of the association events. Let $\beta_{ij}^t = P[\chi_{ij}^t \mid Z_i^k]$ and $\beta_{i0}^t = 1 - \sum_{j=1}^{m_i(k)} P[\chi_{ij}^t \mid Z_i^k]$. β_{i0} denotes the probability that no measurement is associated with target t for node C_i, and χ_{ij}^t denotes the event that the measurement j on node i originated from target t. See [14] for details about computing β_{ij}'s values. The JPDA filter (JPDAF) state estimate is:

$$\begin{aligned} \hat{x}_i^{t+} &= E[x_t \mid Z_i^k] \\ &= \hat{x}_i^{t-} + K_i^t(z_i^t - (1 - \beta_{i0}^t)H_i^t\hat{x}_i^{t-}) \end{aligned} \tag{6}$$

where $\hat{x}_i^{t+}$ and $\hat{x}_i^{t-}$ denote the *a posteriori* estimate and prior estimate of the state of target t by node i at time k, respectively. z_i^t and K_i^t denote the mean measurement and the Kalman gain for target t, respectively. H_i^t is the observation matrix for node C_i for target t. $Z_i(k) = \{z_{i1}(k), z_{i2}(k), \cdots\}$ denotes the set of $m_i(k)$ measurements obtained by node i at time k, and we define $Z_i^k = \{Z_i(1), Z_i(2), \cdots Z_i(k)\}$.

From Equation (6), the mean measurement innovation $\tilde{z}_i^t$ for target t is defined as:

$$\tilde{z}_i^t = z_i^t - (1 - \beta_{i0}^t)H_i^t\hat{x}_i^{t-} \tag{7}$$

where $z_i^t = \sum_{j=1}^{m_i(k)} \beta_{ij}^t z_{ij}$.

The covariance estimate for JPDAF is given by:

$$P_i^{t+} = P_i^{t-} - (1 - \beta_{i0}^t)K_i^t W_i^t (K_i^t)^T + K_i^t \tilde{P}_i^t (K_i^t)^T \tag{8}$$

where:

$$\tilde{P}_i^t = \sum_{j=1}^{m_i(k)} \beta_{ij}^t (z_{ij} - H_i^t\hat{x}_i^{t-})(z_{ij} - H_i^t\hat{x}_i^{t-})^T - (\tilde{z}_i^t)(\tilde{z}_i^t)^T \tag{9}$$

The JPDAF is based on the Kalman filter, which is the best linear estimator. However, in this paper, the camera model is a nonlinear system. The JPDAF needs to be modified to fit the nonlinear system. Details will be discussed in Section 5.

4. Square-Root Cubature Information Weighted Consensus Filter

The square-root cubature Kalman filter (SCKF) algorithm [9] was proposed by Arasaratnam *et al.* It is a more accurate nonlinear filter that could be applied to solve high-dimensional nonlinear filtering problems with minimal computational effort. In multi-sensor data fusion applications, because of the advantages of the information filter mentioned above, this paper uses the information square-root cubature information filter (SCIF) [12]. Firstly, we will give a brief review of SCIF of node i as follows; thus, in order to facilitate the description, the sensor index i will be dropped in this review.

4.1. Square-Root Cubature Information Filter: A Brief Review

The information filter propagates the inverse of P, rather than propagating P. The state estimate and its corresponding covariance in the Kalman filter are replaced by the information vector and information matrix, respectively, in the information filter. The updated information vector and information matrix can be written as:

$$Y_{k|k-1} = P^{-1}_{k|k-1} = S_{y,k|k-1} S^T_{y,k|k-1} \tag{10}$$

$$\hat{y}_{k|k-1} = P^{-1}_{k|k-1} \hat{x}_{k|k-1} = Y_{k|k-1} \hat{x}_{k|k-1} \tag{11}$$

where $S_{y,k|k-1}$ is the square-root information matrix. The information update at time k is given by:

$$Y_{k|k} = Y_{k|k-1} + I_k \tag{12}$$

$$\hat{y}_{k|k} = \hat{y}_{k|k-1} + i_k \tag{13}$$

Here, I_k and i_k are defined as follows, respectively [12]:

$$I_k = (Y_{k|k-1} P_{xz,k|k-1}) R^{-1}_k (Y_{k|k-1} P_{xz,k|k-1})^T \tag{14}$$

$$i_k = (Y_{k|k-1} P_{xz,k|k-1}) R^{-1}_k (z_k - \hat{z}_{k|k-1} + P^T_{xz,k|k-1} \hat{y}_{k|k-1}) \tag{15}$$

In matrix theory, an covariance matrix P can be written as:

$$P = AA^T \tag{16}$$

where $P \in \mathbf{R}^{n \times n}, A \in \mathbf{R}^{n \times m}, m \geq n$. Equation (16) can be considered as the square-root of P. For a simple calculation, in this paper, A is transformed into a $n \times m$ triangular matrix S using a triangularization decomposition algorithm, as follows:

$$S = \mathbf{Tria}(A), S \in \mathbf{R}^{n \times m} \tag{17}$$

Tria denotes a triangularization decomposition algorithm. If we use QR decomposition, A^T will be decomposed into an orthogonal matrix $Q \in \mathbf{R}^{m \times m}$ and an upper triangular matrix $R \in \mathbf{R}^{m \times n}$, $A^T = QR$; then, Equation (16) can be written as:

$$P = AA^T = R^T Q^T QR = R^T R = SS^T \tag{18}$$

then $S = R^T$. S is a lower triangular matrix, which is a sparse matrix. The sparsity of S will benefit calculations and reduce storage space. The steps involved in the square-root cubature information filter algorithm are summarized in the following:

In the time update, at time k, assume that $(\hat{y}_{k-1|k-1}, S_{y,k-1|k-1})$ is known; we compute the square-root of the predicted information matrix $S_{y,k|k-1}$ and the predicted information vector $\hat{y}_{k|k-1}$.

In the measurement update, we will compute the updated information vector $\hat{y}_{k|k}$ and the square-root of the updated information matrix $S_{y,k|k}$ according to the results of the time update step. See [15] for details about these two steps.

4.2. Centralized Square-Root Cubature Information Filter

Multi-sensor fusion is the process by which information from many sensors is combined to yield an improved description of the observed system. In this section, we will give a brief introduction of the centralized square-root cubature information filter (CSCIF), which is the base of the proposed algorithm. A centralized camera network system comprises a fusion center with connections to all other cameras. In order to distinguish between the information state contribution i and node index, we use the s to denote the node index in the rest of paper. Each camera C_s obtains data about the environment, which is forwarded to the fusion center, where $s = (1, 2, \cdots, N_c)$. The global estimate in the fusion center can be computed from N_C sensor measurements at time k by simple summing of the local information vectors and matrices (the "c" superscript denotes "centralized"):

$$Y^c_{k|k} = Y^c_{k|k-1} + \sum_{s=1}^{N_c} I^s_k \tag{19}$$

$$\hat{y}^c_{k|k} = \hat{y}^c_{k|k-1} + \sum_{s=1}^{N_c} i^s_k \tag{20}$$

In Equation (20), the i^s_k can be computed using the equation $i^s_k = S^s_{i,k}\bar{S}^s_{R,k}(z^s_k - \hat{z}^s_{k|k-1}) + S^s_{i,k}(S^s_{i,k})^T\hat{x}^s_{k|k-1}$ [15], where $I^s_k = S^s_{i,k}(S^s_{i,k})^T$, $\bar{S}^s_{R,k} = (S^s_{R,k})^{-T}$ and $(R^s_k)^{-1} = \bar{S}^s_{R,k}(\bar{S}^s_{R,k})^T$. The square-root of $Y^c_{k|k}$ can be computed using Equation (17) as follows:

$$S^c_{y,k|k} = \mathbf{Tria}([S^c_{y,k|k-1} \quad S^1_{i,k} \quad S^2_{i,k} \quad \cdots \quad S^{N_c}_{i,k}]) \tag{21}$$

Then, we compute the square-root of the predicted information matrix $S^c_{y,k+1|k}$ and the predicted information vector $\hat{y}^c_{k+1|k}$ using the standard time update step of SCIF.

Although the centralized camera network system is an improvement over a single camera system, it has a number of disadvantages. These include severe computational loads imposed on the fusion center, the possibility of catastrophic failure and high communication overheads.

4.3. Distributed Square-Root Cubature Information Weighted Consensus Filter

Generally, there are no fusion centers in large-scale camera networks, and the capabilities of all cameras are equal in the network. In this scenario, a distributed approach is required. The average consensus algorithm, which has been introduced in Section 3 of this paper, meets the needs of this scenario. In this section, we propose a novel DSCIWCF algorithm for target tracking in camera networks.

In the average consensus algorithm, node C_s only communicates with its direct neighbors $C_j \in \Omega_s$, then the values of the states at all of the nodes converge to the average of the initial values.

In [2], a distributed state estimation framework was proposed by Kamal *et al.* They used a value $1/N_c$ as weights on the information matrix and information vector. This algorithm can overcome the naivety issue and information redundancy in camera networks. In this paper, we use a similar strategy to deal with the square-root cubature information matrix and information vector. Here, we summarize the DSCIWCF algorithm as follows.

(1) Compute the square-root form of the local information vector $\hat{y}^s_{k|k}$ and the information matrix $Y^s_{k|k}$:

$$S^s_{y,k|k} = \mathbf{Tria}([\frac{1}{\sqrt{N_C}} S^s_{y,k|k-1} \quad S^s_{i,k}]) \tag{22}$$

$$\hat{y}^s_{k|k} = \frac{1}{N_c}\hat{y}^s_{k|k-1} + i^s_k \tag{23}$$

$$S^s_{i,k} = S^s_{y,k|k-1}(S^s_{y,k|k-1})^T P_{s,xz,k|k-1}\bar{S}^s_{R,k} \tag{24}$$

where $Y^s_{k|k-1} = S^s_{y,k|k-1}(S^s_{y,k|k-1})^T$ and $P_{s,xz,k|k-1} = T_{21}T_{11}^T$ can be computed using Equation (A5) (see Appendix A). Equation (22) is equivalent to the equation below.

$$Y^s_{k|k} = \frac{1}{N_c} Y^s_{k|k-1} + I^s_k \tag{25}$$

(2) Let $\nu^s_0 = \hat{y}^s_{k|k}$ and $V^s_0 = S^s_{y,k|k}$, then perform average consensus on ν^s_0 and V^s_0 independently for K iterations:

For $k = 1$ to K:
Broadcast (ν^s_{k-1}, V^s_{k-1}) to neighbors $C_j, C_j \in \Omega_s$ and receive (ν^j_{k-1}, V^j_{k-1}) from neighbors.
Run average consensus on ν^s_{k-1}, V^s_{k-1}:

$$V^s_k = \mathbf{Tria}([\sqrt{1-\epsilon N_{s,E}} V^s_{k-1} \quad \sqrt{\epsilon} V^s_{1,k-1} \quad \cdots \quad \sqrt{\epsilon} V^s_{N_{s,E},k-1}]) \tag{26}$$

$$\nu^s_k = \nu^s_{k-1} + \epsilon \sum_{j\in\Omega_s} (\nu^j_{k-1} - \nu^s_{k-1}) \tag{27}$$

END for:
where $N_{s,E}$ is the number of the direct neighbors of node s, $N_{s,E} = \sum_j a_{sj}$.
If $V^{'s}_0 = \frac{1}{N_c} Y^s_{k|k-1} + I^s_k$, the equivalent square form of the Equation (26) is as follows:

$$V^{'s}_k = V^{'s}_{k-1} + \epsilon \sum_{j\in\Omega_s} (V^{'j}_{k-1} - V^{'s}_{k-1}) \tag{28}$$

(3) Compute the *a posteriori* information vector and information matrix for time k:

$$\begin{aligned} \hat{y}^s_{k|k} &= N_c \nu^s_K \\ S^s_{y,k|k} &= \sqrt{N_c} V^s_K \end{aligned} \tag{29}$$

(4) Compute the predicted information matrix $S^s_{y,k+1|k}$ and the predicted information vector $\hat{y}^s_{k+1|k}$ using the standard time update step of SCIF.

In practice, due to errors introduced by arithmetic operations (such as matrix square-root, matrix inversion, *etc*.) performed on finite word-length digital computers, the symmetry and positive definiteness of the error covariance matrix are often lost [9]. A square root filter is the best choice to deal with these problems. In this paper, we use the square-root cubature information filter for target tracking in camera networks. At the same time, we use Equations (26) and (27) for average consensus iterations. Therefore, the whole algorithm runs under the condition of the square-root filter.

5. Multi-Target Data Association

The JPDA algorithm has been introduced in Section 3. However, the traditional JPDA algorithm is often used for linear sensing models. In this section, we will extend the JPDA algorithm to handle nonlinear sensing models. The main contribution of this section is that we propose the algorithm derived from the combination of JPDA and the information filter mentioned in the last section. The JPDAF is a single sensor algorithm; thus, we firstly introduce the algorithm of the single node s.

5.1. Joint Probabilistic Data Association With Square-Root Cubature Information Filter

In the SCIF, in order to make the information contribution equations compatible with those of the Kalman filter, a pseudo-measurement matrix H_s^t [25] is defined as (at target t, similarly hereinafter):

$$H_s^t = P_{s,xz,k|k-1}^{t^T} Y_{s,k|k-1}^t \tag{30}$$

where the subscript s denotes the terms from the s-th node.

In the cubature Kalman filter (CKF), the Kalman gain $K_{s,k}^t$ gives:

$$K_{s,k}^t = P_{s,xz,k|k-1}^t (P_{s,zz,k|k-1}^t)^{-1} \tag{31}$$

where $K_{s,k}^t = T_{21}^t T_{11}^{t^{-1}}$ can be computed using Equation (A5) (see Appendix A).

The innovation covariance matrix is given by $W_s^t = H_s^t P_s^{t-} (H_s^t)^T + R_s^t$ [15], and substituting Equation (30) into the innovation covariance matrix, we can get:

$$\begin{aligned} W_s^t &= P_{s,xz,k|k-1}^{t^T} Y_{s,k|k-1}^t P_{s,xz,k|k-1}^t + R_{s,k}^t \\ &= \Phi_{k|k-1}^t \Phi_{k|k-1}^{t^T} + R_{s,k}^t = P_{s,zz,k|k-1}^t \end{aligned} \tag{32}$$

where $Y_{s,k|k-1}^t = (P_{s,k|k-1}^t)^{-1}$ is a symmetric positive definite matrix, and $P_{s,xz,k|k-1}^t$, $P_{s,zz,k|k-1}^t$, $P_{s,k|k-1}^t$ and $\Phi_{k|k-1}^t$ come from Equation (A1). $P_{s,zz,k|k-1}^t$ can be computed using $T_{11}^t T_{11}^{t^T}$ (see Appendix A). Now, we can rewrite Equation (31) as follows:

$$K_{s,k}^t = P_s^{t-} (H_s^t)^T (W_s^t)^{-1} \tag{33}$$

where $H_{s,k}^t$ and W_s^t are defined by Equations (30) and (32); P_s^{t-} is the short form of $P_{s,k|k-1}^t$.

In the CKF, the measurement innovation term of Equation (7) becomes,

$$\tilde{z}_s^t = z_s^t - (1 - \beta_{i0}^t) \hat{z}_{s,k|k-1} \tag{34}$$

where $\hat{z}_{s,k|k-1} = \frac{1}{m} \sum_{j=1}^m Z_{sj,k|k-1}$ and $Z_{sj,k|k-1}$ denotes one of the propagated cubature points; see [15] for details about computing $Z_{sj,k|k-1}$. Substituting Equation (34) into Equations (6) gives:

$$\hat{x}_s^{t+} = \hat{x}_s^{t-} + K_{s,k}^t [z_s^t - (1 - \beta_{s0}^t) \hat{z}_{s,k|k-1}] \tag{35}$$

The JPDA state update Equation (35) has a similar form as the standard Kalman filter update, and it can be converted to the information form using $u_s^t = H_s^{t^T} R_s^{t^{-1}} z_s^t$ and $U_s^t = H_s^{t^T} R_s^{t^{-1}} H_s^t$. We get (see Appendix B):

$$\hat{x}_s^{t+} = (Y_s^{t-} + U_s^t)^{-1} \{ Y_s^{t-} \hat{x}_s^{t-} + u_s^t + U_s^t \hat{x}_s^{t-} - (1 - \beta_{s0}^t) H_s^{t^T} R_s^{t^{-1}} \hat{z}_{s,k|k-1} \} \tag{36}$$

where R_s^t is the measurement noise covariance of node s for target t and $Y_s^{t-} = (P_s^{t-})^{-1} = (P_{s,k|k-1}^t)^{-1}$.

The information matrix is denoted as follows (see Appendix B):

$$Y_s^{t+} = Y_s^{t-} + G_s^t \tag{37}$$

where $G_s^t = Y_s^{t-}K_s^t[(M_s^t)^{-1} - (K_s^t)^T Y_s^{t-} K_s^t]^{-1}(K_s^t)^T Y_s^{t-}$, $M_s^t = (1 - \beta_{s0}^t)W_s^t - \tilde{P}_s^t$ and $\tilde{P}_s^t$ (see Equation (B4) in Appendix B). Equations (36) and (37) form the JPDA-SCIF algorithm.

5.2. Joint Probabilistic Data Association With Centralized Square-Root Cubature Information Filter

In Equations (12) and (13), the information filter form has the advantage that the update equations for the estimator are computationally simpler than the equations for the Kalman filter. Here, we rewrite I_s^t and i_s^t using H_s^t from Equation (30) as follows:

$$I_s^t = H_s^{t^T} R_s^{t^{-1}} H_s^t \tag{38}$$

$$i_s^t = H_s^{t^T} R_s^{t^{-1}} (z_s^t - \hat{z}_s^{t-} + H_s^t \hat{x}_{s,k|k-1}^t) \tag{39}$$

where the measurement innovation term $\tilde{z}_s^t = z_s^t - \hat{z}_s^{t-}$ and $\hat{x}_{s,k|k-1}^t$ represents the prior/predicted state estimate. In the JPDAF, $\tilde{z}_s^t$ can be written as Equation (34). Therefore, Equation (39) can be rewritten as follows:

$$i_s^t = H_s^{t^T} R_s^{t^{-1}} (\tilde{z}_s^t + H_s^t \hat{x}_{s,k|k-1}^t) \tag{40}$$

From Equations (38) and (40), we rewrite Equations (12) and (13) as follows:

$$\begin{aligned} Y_c^{t+} &= Y_c^{t-} + \sum_{s=1}^{N_C} I_s^t \\ &= Y_c^{t-} + \sum_{s=1}^{N_C} H_s^{t^T} R_s^{t^{-1}} H_s^t \end{aligned} \tag{41}$$

$$\begin{aligned} Y_c^{t+}\hat{x}_c^{t+} &= Y_c^{t-}\hat{x}_c^{t-} + \sum_{s=1}^{N_C} i_s^t \\ &= Y_c^{t-}\hat{x}_c^{t-} + \sum_{s=1}^{N_C} H_s^{t^T} R_s^{t^{-1}} (\tilde{z}_s^t + H_s \hat{x}_s^{t-}) \end{aligned} \tag{42}$$

Substituting Equation (34) into Equation (42), we can extend Equations (36) and (37) into the multi-sensor centralized estimate in the information form as follows:

$$\hat{x}_c^{t+} = (Y_c^{t-} + \sum_{s=1}^{N_C} U_s^t)^{-1} \{Y_c^{t-}\hat{x}_c^{t-} + \sum_{s=1}^{N_C} [u_s^t + U_s^t \hat{x}_s^{t-} - (1 - \beta_{s0}^t) H_s^{t^T} R_s^{t^{-1}} \hat{z}_{s,k|k-1}]\} \tag{43}$$

$$Y_c^{t+} = Y_c^{t-} + \sum_{s=1}^{N_C} G_s^t \tag{44}$$

where $G_s^t = Y_s^{t-}K_s^t[(M_s^t)^{-1} - (K_s^t)^T Y_s^{t-} K_s^t]^{-1}(K_s^t)^T Y_s^{t-}$, $u_s^t = H_s^{t^T} R_s^{t^{-1}} z_s^t$, $U_s^t = I_s^t$, $M_s^t = (1 - \beta_{s0}^t)W_s^t - \tilde{P}_s^t$, and $\tilde{P}_s^t$ is defined as follows:

$$\tilde{P}_s^t = \sum_{j=1}^{m_s(k)} \beta_{sj}^t (z_{sj} - \hat{z}_{s,k|k-1})(z_{sj} - \hat{z}_{s,k|k-1})^T - (\tilde{z}_s^t)(\tilde{z}_s^t)^T \tag{45}$$

5.3. Multi-Target Square-Root Cubature Information Weighted Consensus Filter

In the MTSCF, if all nodes have reached consensus on the previous time step, we will have $\hat{x}_s^{t-} = \hat{x}_c^{t-}$ and $Y_s^{t-} = Y_c^{t-}$ for all s. That is to say, $Y_c^{t-} = \sum_{s=1}^{N_C} \frac{Y_s^{t-}}{N_C}$, and $Y_c^{t-}\hat{x}_c^{t-} = \sum_{s=1}^{N_C} \frac{Y_s^{t-}}{N_C}\hat{x}_s^{t-}$. Thus, we can write,

$$\hat{x}_s^{t+} = \sum_{s=1}^{N_C}(\frac{Y_s^{t-}}{N_C} + U_s^t)^{-1} \sum_{s=1}^{N_C}\{\frac{Y_s^{t-}}{N_C}\hat{x}_s^{t-} + u_s^t + U_s^t\hat{x}_s^{t-} - H_s^{t^T} R_s^{t^{-1}}(1 - \beta_{s0}^t)\hat{z}_{s,k|k-1}\} \tag{46}$$

$$Y_s^{t+} = \sum_{s=1}^{N_C}(\frac{Y_s^{t-}}{N_C} + G_s^t) \tag{47}$$

Thus, for the MTSCF algorithm, the consensus variables are initialized as,

$$\nu_s^t[0] = \frac{Y_s^{t-}}{N_C}\hat{x}_s^{t-} + u_s^t + U_s^t\hat{x}_s^{t-} - (1 - \beta_{s0}^t)H_s^{t^T} R_s^{t^{-1}}\hat{z}_{s,k|k-1} \tag{48}$$

$$V_s^t[0] = \frac{Y_s^{t-}}{N_C} + U_s^t \tag{49}$$

$$M_s^t[0] = \frac{Y_s^{t-}}{N_C} + G_s^t \tag{50}$$

The MTSCF algorithm is summarized in Algorithm 1.

5.4. Computing the Square-Root of the Information Matrix

The algorithms of this section that were mentioned above are based on information matrix. However, in order to be consistent with the algorithms proposed in this section, we need to transform the information matrix into the square-root form.

Illustrated by the case of the MTSCF algorithm, we will describe how to compute the square-root of the information matrix and other algorithms using similar methods.

Let $J_s^t J_s^{t^T} = M_s^t[0]$; because of the positive definite matrix of $M_s^t[0]$, we can use the Cholesky decomposition to compute J_s^t. The square-root form of Equation (50) is as follows:

$$S_{M_s^t[0]} = \mathbf{Tria}(J_s^t) \tag{51}$$

It is easy to get $S_{U_s^t}$ and $S_{Y_s^{t-}}$; then, the square-root form of Equation (49) is as follows:

$$S_{V_s^t[0]} = \mathbf{Tria}([\frac{1}{\sqrt{N_C}} S_{Y_s^{t-}} \quad S_{U_s^t}]) \tag{52}$$

where $S_{Y_s^{t-}}$ and $S_{U_s^t}$ are the square-root form of Y_s^{t-} and U_s^t, respectively. They can be computed using $U_s^t = S_{U_s^t} S_{U_s^t}^T$ and $Y_s^{t-} = S_{Y_s^{t-}} S_{Y_s^{t-}}^T$.

Algorithm 1 MTSCF Algorithm for target t at node C_s at time k.

Input: $x_s^{t-}(k), S_{Y_s^{t-}}(k)$ and R_s^t

(1) Compute H_s^t using Equation (30)

(2) Get measurements: $\{z_s^n\}_{n=1}^{m_s(k)}$

(3) Compute $W_s^t, K_s^t, \beta_{s0}^t, M_s^t, z_s^t, \hat{z}_{s,k|k-1}, Y_s^{t-}(k)$

(4) Compute information vectors and matrices (using Equation (24)):

$$u_s^t = H_s^{t^T} R_s^{t^{-1}} z_s^t \tag{53}$$

$$S_{U_s^t} = S_{Y_s^{t-}}(k) S_{Y_s^{t-}}(k)^T P_{s,xz,k|k-1} \bar{S}_{s,R,k} \tag{54}$$

(5) Broadcast message to neighbors and receive neighbors' messages

(6) Compute inter-camera track-to-track matchings

(7) Initialized consensus data

$$\nu_s^t[0] = \frac{Y_s^{t-}}{N_C} \hat{x}_s^{t-} + u_s^t + U_s^t \hat{x}_s^{t-} - (1 - \beta_{s0}^t) H_s^{t^T} R_s^{t^{-1}} \hat{z}_{s,k|k-1} \tag{55}$$

$$S_{V_s^t[0]} = \mathbf{Tria}([\frac{1}{\sqrt{N_C}} S_{Y_s^{t-}} \quad S_{U_s^t}]) \tag{56}$$

$$S_{M_s^t[0]} = \mathbf{Tria}(J_s^t) \tag{57}$$

(8) Perform average consensus on $\nu_s^t[0]$, $S_{V_s^t[0]}$ and $S_{M_s^t[0]}$ independently for K iterations

(9) Estimate:

$$\begin{aligned} \hat{x}_s^{t+}(k) &= \frac{1}{(S_{V_s^t[K]} S_{V_s^t[K]}^T)} \nu_s^t[K] \\ S_{Y_s^{t+}}(k) &= \sqrt{N_c} S_{M_s^t[K]} \end{aligned} \tag{58}$$

(10) Compute the predicted state $S_{s,k+1|k}^t$ and the predicted error covariance $\hat{x}_s^{t-}(k+1)$ [12], respectively, then compute the square-root of the predicted information matrix and information vector

$$S_{Y_s^{t-}}(k+1) = S_{s,k+1|k}^{t-T} \tag{59}$$

Output: $\hat{x}_s^{t+}(k), S_{Y_s^{t+}}(k), \hat{x}_s^{t-}(k+1), S_{Y_s^{t-}}(k+1)$

5.5. Inter-Camera Association

In distributed tracking of multiple targets, every node has a set of information from each of its neighbors about the targets and its own set of estimated tracks. Therefore, it is necessary to use an assignment algorithm to form a set of optimal matchings g_{sj}, where g_{sj} matches the tracks of node s with the tracks of node j. We can use the Hungarian algorithm [26] to find the maximum matching. The

matching cost between two track estimates from different cameras can be defined as the Mahalanobis distance as follows [15]:

$$D(\bar{x}_s^{t1}, \bar{x}_j^{t2}) = (\bar{x}_s^{t1} - \bar{x}_j^{t2})^T (P_s^{t1} + P_j^{t2})^{-1} (\bar{x}_s^{t1} - \bar{x}_j^{t2})$$

6. Experimental Evaluation

In this section, we evaluate the performance of the proposed MTSCF algorithm in a nonlinear simulated environment and compare it with other methods: JPDA-EKCF [18] and EMTIC [20]. Our experiments are performed on an Intel 3.4 GHz PC with 4 G memory and implemented in MATLAB.

Four simulated targets ($N_T = 4$) moving in a 500 m $\times$ 500 m area under the 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 200 m $\times$ 200 m around the camera. The target's state vector is a 5D vector, which includes the target's position (x_k, y_k) at discrete time instant k, its velocity (v_x, v_y) and the time interval δ_k between the two consecutive measurements. Accordingly, the state vector is given by $\mathbf{x}_k = [x_k \quad y_k \quad v_x \quad v_y \quad \delta_k]^T$.

The motion model of the targets is described by the nonlinear equation [8]:

$$\mathbf{x}_{k+1} = \begin{pmatrix} x_k + v_{x,k}\delta_k + a_x \frac{\delta_k^2}{2} \\ y_k + v_{y,k}\delta_k + a_y \frac{\delta_k^2}{2} \\ v_{x,k} + a_x \delta_k \\ v_{y,k} + a_y \delta_k \\ \delta_k + e \end{pmatrix} \tag{60}$$

where the target acceleration (a_x, a_y) is modeled as Gaussian noise. To account for synchronization errors among cameras, we consider a time uncertainty e, which is also assumed to be a Gaussian variable. We consider the vector $v = (a_x, a_y, e)$ as the Gaussian noise vector with zero mean and covariance $Q = diag([5 \quad 5 \quad 0.01])$. The initial speed is randomly obtained from the range $10 \sim 30$ units per time step and with a random direction uniformly chosen from 0–2π.

The measurement model can be defined as:

$$\mathbf{z}_k^s = \begin{pmatrix} \gamma_k^s \\ \phi_k^s \end{pmatrix} = \begin{pmatrix} \frac{H_{11}^s x_k + H_{12}^s y_k + H_{13}^s}{H_{31}^s x_k + H_{32}^s y_k + H_{33}^s} \\ \frac{H_{21}^s x_k + H_{22}^s y_k + H_{23}^s}{H_{31}^s x_k + H_{32}^s y_k + H_{33}^s} \end{pmatrix} + w_k \tag{61}$$

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 Gaussian with zero mean and variance $R = diag([5 \quad 5])$. The homography matrix values of each camera are taken from the camera C_6 of the APIDISdataset [27] whose values are:

$$H_s = \begin{pmatrix} 1,930.8939 & -89.8033 & -2,393,800 \\ 117.2530 & 91.8121 & 1,022,700 \\ 0.3485 & -0.8720 & 1,971.8862 \end{pmatrix} \tag{62}$$

The initial prior covariance $P_s^{t-}(1) = diag([100, 100, 10, 10, 0.01])$ is used at each node for each target. The initial prior state $\hat{x}_s^{t-}(1)$ is generated by adding zero-mean Gaussian noise of covariance

$P_s^{t-}(1)$ to theinitial ground truth state. The observations are generated using Equation (61). The total number of consensus iterations per measurement step, K, is set to 20. The parameters for computing the association probabilities, β_{sj}^t, are set as follows (see [14] for details about computing β_{sj}^t). False measurements (clutter) are generated at each node at each measurement step using a Poisson process with λ, where λ is the average number of false measurements per sensor per measurement step. Gate probability P_G is set to 0.99. The probability of detecting a target P_D in each camera is set to 0.8.

In this paper, we perform the experiments for a sparse connectivity network with a low average network degree equal to two (see Figure 1). Therefore, the $\Delta_{max} = 2$; then, the consensus rate parameter ϵ is set to $0.65/\Delta_{max}$. In the experiment, four targets' trajectories are generated (see Figure 2). The simulation results are averaged over 20 Monte Carlo simulation runs.

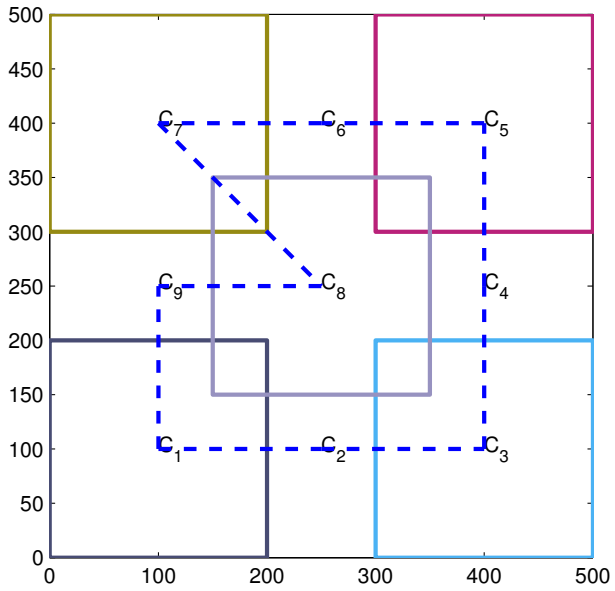

Figure 1. Sparse connectivity of the network.

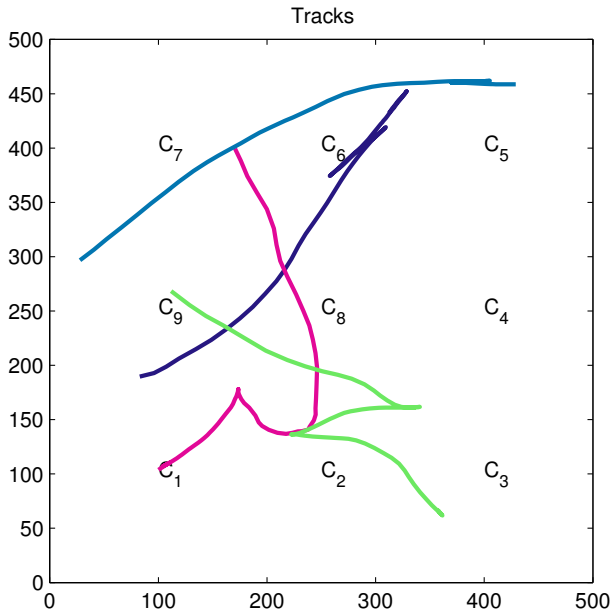

Figure 2. Four true trajectories of targets and their geographical positions in the cameras.

Figure 3 shows the performance comparison by varying the amount of clutter. The average amount of clutter per sensor per measurement step λ is varied from $1/64$–8 (consensus is run for a fixed number of iterations (eight)). From Figure 3, it can be seen that both EMTIC and MTSCF are very robust, even to a very high amount of clutter. The amount of clutter is kept at $\lambda = 1$ for the other experiments in the rest of the paper.

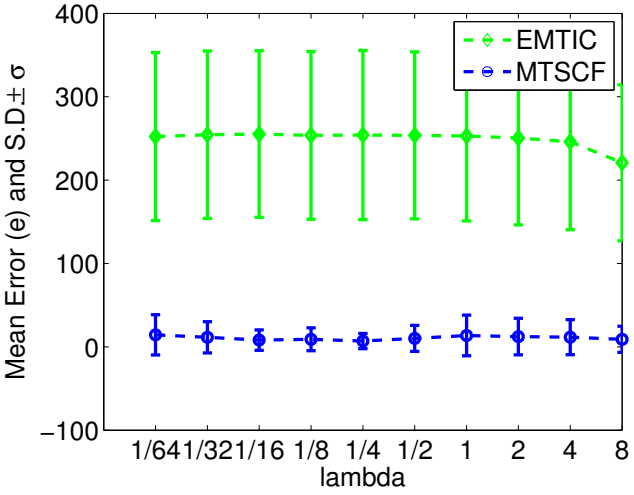

Figure 3. Performance comparison by varying the amount of clutter.

The result of target tracking can be seen in Figure 4 in one experiment from one run (the result is based on the consensus algorithm, and the number of consensus iterations is eight). As you can see from Figure 4, the MTSCF algorithm is closer to the ground-truth curves than EMTIC.

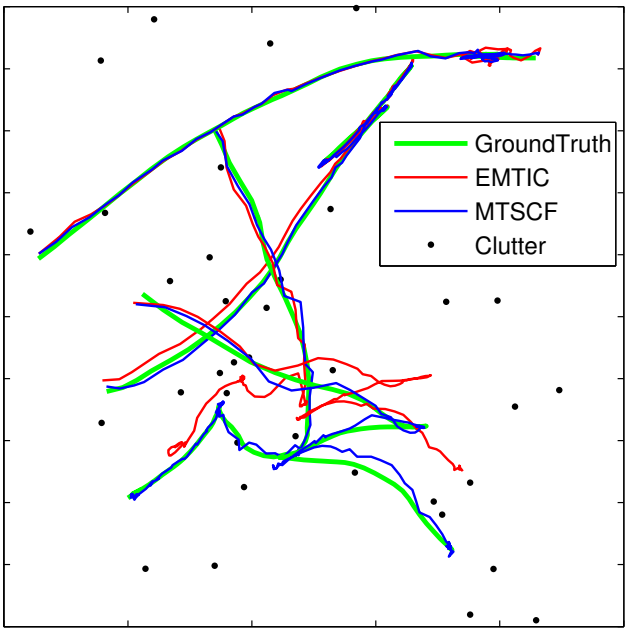

Figure 4. The result of target tracking in one experiment from one run.

To show the convergence of different methods, the total number of iterations per measurement step, K, is varied. It can be seen from Figures 5 and 6 (Figure 6 shows an enlarged part of Figure 5 focusing on

MTSCF and EMTIC) that with an increased number of iteration, MTSCF approaches the ground-truth tracks. It can also be seen that MTSCF outperforms EMTIC for any given K. Meanwhile, It can be seen that JPDA-EKCF has large mean error, which does not suit nonlinear multi-target tracking in distributed camera networks.

Our simulation is based on MATLAB. In MATLAB, it handles floating-point numbers in double precision (default setting) format; while double precision numbers use 64 bits, based on IEEE Standard 754. In the experiment, we convert all double precision numbers to single precision (32 bits) numbers using the command single (number). Unfortunately, it may be impossible for us to use the single precision in JPDA-EKCF and EMTIC. The reason is that, when the single precision number is used to calculate the updated inverse matrix, the resulting matrix may possibly be non-positive definite. In the simulation, we get the warning "Matrix is close to singular or badly scaled. Results may be inaccurate." Hence, errors may occur during the execution of the JPDA-EKCF and EMTIC algorithms in a limited word-length system. This is not a problem for the MTSCF algorithm.

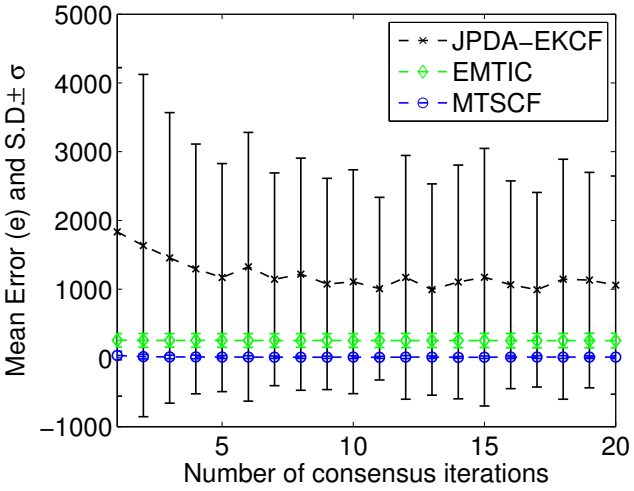

Figure 5. The mean errors and the variation of the estimation errors about three algorithms (joint probabilistic data association (JPDA)-EKCF, EMTICand multi-target square-root cubature information weighted consensus filter (MTSCF)).

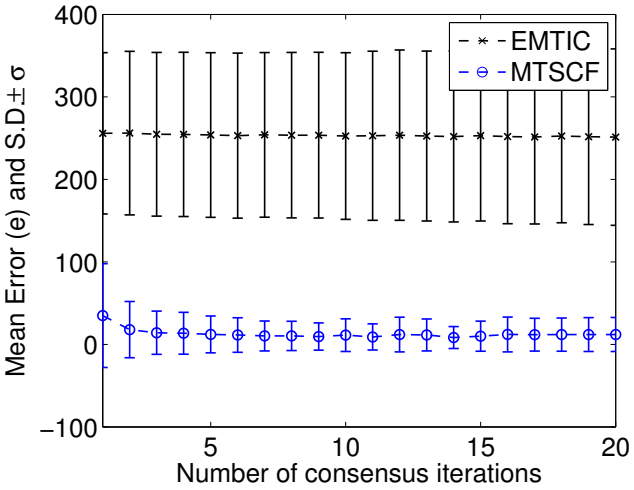

Figure 6. Zoom of Figure 5 with focus on MTSCF and EMTIC.

7. Conclusions

In this paper, we propose a novel multi-target square-root cubature information weighted consensus filter (MTSCF) algorithm, which is a generalized consensus-based distributed multi-target tracking scheme applicable to a wide variety of sensor networks. MTSCF handles the issue of naivety, which makes it applicable to sensor networks where sensors may have limited FOV (which is the case for a camera network). The algorithm is efficient for considering the estimation errors in tracking and data association, the influence of naivety and the numerical difficulties from the finite word-length of computers, which makes it resistive to false measurements/clutter. Experimental analysis shows the strength of the proposed method over existing ones.

In our future work, we will explore applying the MTSCF to a real camera network, which may be a limited word-length embedded system. Handling out-of-sequence measurements, the unknown number of targets and asynchronous networks are some other possible future works.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61175096 and the Specialized Fund for the Joint Building Program of the Beijing Municipal Education Commission.

Author Contributions

The work presented here was carried out in collaboration between all authors. Qingjie Zhao proposed the research theme with regular feedbacks of suggestions/ideas. Yanming Chen designed the methods, implementation and carried out the experiments. All authors have contributed, reviewed and improved the manuscript.

Appendixes

A. Squared-Matrix Forms of Three Covariance Matrices

$P_{k|k-1}, P_{zz,k|k-1}, P_{xz,k|k-1}$ may be expressed in squared-matrix forms [28]:

$$\begin{aligned}&\begin{pmatrix} P_{zz,k|k-1} & P_{zx,k|k-1} \\ P_{xz,k|k-1} & P_{k|k-1} \end{pmatrix} \\ &= \begin{pmatrix} \Phi_{k|k-1} & S_{R,k} \\ \Psi_{k|k-1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Phi_{k|k-1} & S_{R,k} \\ \Psi_{k|k-1} & \mathbf{0} \end{pmatrix}^T \end{aligned} \tag{A1}$$

where $S_{R,k}$ is the square-root of R_k, $\mathbf{0} \in \mathbf{R}^{n_x \times n_z}$, and:

$$\begin{aligned}\Psi_{k|k-1} = &\frac{1}{\sqrt{m}}[X_{1,k|k-1} - \hat{x}_{k|k-1} \quad X_{2,k|k-1} - \hat{x}_{k|k-1} \\ &\cdots \quad X_{m,k|k-1} - \hat{x}_{k|k-1}]\end{aligned} \tag{A2}$$

$$\Phi_{k|k-1} = \frac{1}{\sqrt{m}}[Z_{1,k|k-1} - \hat{z}_{k|k-1} \quad Z_{2,k|k-1} - \hat{z}_{k|k-1} \quad \cdots \quad Z_{m,k|k-1} - \hat{z}_{k|k-1}] \tag{A3}$$

Applying the triangularization procedure to the square-root factor available on the RHS of Equation (A1) yields:

$$\mathbf{Tria}\begin{pmatrix} \Phi_{k|k-1} & S_{R,k} \\ \Psi_{k|k-1} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} T_{11} & \mathbf{0} \\ T_{21} & T_{22} \end{pmatrix} \tag{A4}$$

where $T_{11} \in \mathbf{R}^{n_z \times n_z}, T_{22} \in \mathbf{R}^{n_x \times n_x}$ are lower triangular matrices, and $T_{21} \in \mathbf{R}^{n_x \times n_z}$. Equation (A1) can be rewritten as follows:

$$\begin{aligned} &\begin{pmatrix} P_{zz,k|k-1} & P_{zx,k|k-1} \\ P_{xz,k|k-1} & P_{k|k-1} \end{pmatrix} \\ &= \begin{pmatrix} T_{11} & \mathbf{0} \\ T_{21} & T_{22} \end{pmatrix}\begin{pmatrix} T_{11} & \mathbf{0} \\ T_{21} & T_{22} \end{pmatrix}^T \\ &= \begin{pmatrix} T_{11}T_{11}^T & T_{11}T_{21}^T \\ T_{21}T_{11}^T & T_{21}T_{21}^T + T_{22}T_{22}^T \end{pmatrix} \end{aligned} \tag{A5}$$

B. JPDA-SCIF: Derivation

Kalman gain can be rewritten as follows [20]. The time index k has been dropped for simplicity.

$$\begin{aligned} K_s &= P_s^{t-}(H_s^t)^T(W_s^t)^{-1} \\ &= (P_s^{t-^{-1}} + H_s^{t^T} R_s^{t-^{-1}} H_s^t)^{-1} H_s^{t^T} R_s^{t-^{-1}} \end{aligned} \tag{B1}$$

The state estimate,

$$\begin{aligned} \hat{x}_s^{t+} &= \hat{x}_s^{t-} + K_s^t \tilde{z}_s^t \\ &= \hat{x}_s^{t-} + K_s^t[z_s^t - (1 - \beta_{s0}^t)\hat{z}_{s,k|k-1}] \\ &= \hat{x}_s^{t-} + (P_s^{t-^{-1}} + H_s^{t^T} R_s^{t^{-1}} H_s^t)^{-1} H_s^{t^T} R_s^{t^{-1}} \\ &\quad [z_s^t - (1 - \beta_{s0}^t)\hat{z}_{s,k|k-1}] \\ &= \hat{x}_s^{t-} + (Y_s^{t-} + U_s^t)^{-1}(u_s^t - (1 - \beta_{s0}^t) \\ &\quad H_s^{t^T} R_s^{t^{-1}} \hat{z}_{s,k|k-1}) \\ &= (Y_s^{t-} + U_s^t)^{-1}\{Y_s^{t-}\hat{x}_s^{t-} + u_s^t + U_s^t \hat{x}_s^{t-} \\ &\quad - (1 - \beta_{s0}^t) H_s^{t^T} R_s^{t^{-1}} \hat{z}_{s,k|k-1}\} \end{aligned} \tag{B2}$$

For the covariance, rewrite Equation (8) as follows:

$$\begin{aligned} P_s^{t+} &= P_s^{t-} - (1 - \beta_{s0}^t) K_s^t W_s^t (K_s^t)^T + K_s^t \tilde{P}_s^t (K_s^t)^T \\ &= P_s^{t-} - K_s^t[(1 - \beta_{s0}^t) W_s^t - \tilde{P}_s^t](K_s^t)^T \end{aligned} \tag{B3}$$

where,

$$\begin{aligned} \tilde{P}_s^t = &\sum_{j=1}^{m_s(k)} \beta_{sj}^t (z_{sj} - \hat{z}_{s,k|k-1})(z_{sj} - \hat{z}_{s,k|k-1})^T \\ &- (\tilde{z}_s^t)(\tilde{z}_s^t)^T \end{aligned} \tag{B4}$$

Let $M_s^t = (1 - \beta_{s0}^t)W_s^t - \tilde{P}_s^t$, then use the matrix inversion lemma on Equation (B3); we get:

$$Y_s^{t+} = Y_s^{t-} + G_s^t \tag{B5}$$

where $G_s^t = Y_s^{t-}K_s^t[(M_s^t)^{-1} - (K_s^t)^T Y_s^{t-} K_s^t]^{-1}(K_s^t)^T Y_s^{t-}$.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Akyildiz, I.F.; Melodia, T.; Chowdhury, K.R. A survey on wireless multimedia sensor networks. *Comput. Netw.* **2007**, *51*, 921–960.
2. Kamal, A.T.; Farrell, J.A.; Roy-Chowdhury, A.K. Information Weighted Consensus Filters and Their Application in Distributed Camera Networks. *IEEE Trans. Autom. Control* **2013**, *58*, 3112–3125.
3. Tron, R.; Vidal, R. Distributed computer vision algorithms. *IEEE Signal Process. Mag.* **2011**, *28*, 32–45.
4. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *IEEE Proc.* **2007**, *95*, 215–233.
5. Ren, W.; Beard, R.W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **2005**, *50*, 655–661.
6. Olfati-Saber, R. Kalman-consensus Filter: Optimality, Stability, and Performance. In Proceedings of the 48th IEEE Conference on Decision and Control, Held Jointly with the 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 7036–7042.
7. Olfati-Saber, R.; Sandell, N.F. Distributed Tracking in Sensor Networks with Limited Sensing Range. In Proceedings of the 2008 IEEE American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 3157–3162.
8. Katragadda, S.; SanMiguel, J.C.; Cavallaro, A. Consensus Protocols for Distributed Tracking in Wireless Camera Networks. In Proceedings of the 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.
9. Arasaratnam, I.; Haykin, S. Cubature kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269.
10. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *IEEE Proc.* **2004**, *92*, 401–422.
11. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotic (Intelligent Robotics and Autonomous Agents)*; The MIT Press: Cambridge, MA, USA, 2005.
12. Arasaratnam, I. Sensor fusion with square-root cubature information filtering. *Intell. Control Autom.* **2013**, *4*, 11, doi:10.4236/ica.2013.41002.
13. Reid, D.B. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* **1979**, *24*, 843–854.
14. Bar-Shalom, Y.; Daum, F.; Huang, J. The probabilistic data association filter. *IEEE Control Syst.* **2009**, *29*, 82–100.

15. Sandell, N.F.; Olfati-Saber, R. Distributed Data Association for Multi-target Tracking in Sensor Networks. In Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 1085–1090.
16. Soto, C.; Song, B.; Roy-Chowdhury, A.K. Distributed Multi-target Tracking in a Self-configuring Camera Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1486–1493.
17. Ni, Z.; Sunderrajan, S.; Rahimi, A.; Manjunath, B. Distributed Particle Filter Tracking with Oonline Mmultiple Iinstance Learning in a Camera Sensor Network. In Proceedings of the 2010 17th IEEE International Conference on Image Processing (ICIP), Hong Kong, China, 26–29 September 2010; pp. 37–40.
18. Roy-Chowdhury, A.K.; Song, B. Camera networks: The acquisition and analysis of videos over wide areas. *Synth. Lect. Comput. Vis.* **2012**, *3*, 1–133.
19. Kamal, A.T.; Farrell, J.A.; Roy-Chowdhury, A.K. Information Consensus for Distributed Multi-target Tracking. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2403–2410.
20. Kamal, A.T. Information Weighted Consensus for Distributed Estimation in Vision Networks. Ph.D. Thesis, University of California Riverside, Riverside, CA, USA, August 2013.
21. Godsil, C.D.; Royle, G.; Godsil, C. Graduate Texts in Mathematics *Algebraic Graph Theory*; Springer: New York, NY, USA, 2001; Volume 207.
22. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533.
23. Olfati-Saber, R. Distributed Kalman filtering for Sensor Networks. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5492–5498.
24. Boyd, S.; Diaconis, P.; Xiao, L. Fastest mixing Markov chain on a graph. *SIAM Rev.* **2004**, *46*, 667–689.
25. Lee, D.J. Nonlinear estimation and multiple sensor fusion using unscented information filtering. *IEEE Signal Process. Lett.* **2008**, *15*, 861–864.
26. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97.
27. Chen, F.; Delannay, D.; de Vleeschouwer, C. An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study. *IEEE Trans. Multimed.* **2011**, *13*, 1381–1394.
28. Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature Kalman filtering for continuous-discrete systems: theory and simulations. *IEEE Trans. Signal Process.* **2010**, *58*, 4977–4993.