IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 22, 2019, accepted September 4, 2019, date of publication September 12, 2019,
date of current version September 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940285

Distributed Multitarget Tracking Based on Diffusion Strategies Over Sensor Networks

YIHUA YU[1], (Member, IEEE), AND YUAN LIANG[2]
[1]School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]School of Humanities and Social Science, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Yuan Liang (teabit@163.com)

ABSTRACT We consider the distributed multitarget tracking over sensor networks, where each node only communicates with its neighbors. We develop a diffusion-based distributed multisensor multitarget tracking algorithm. The state update of the diffusion-based distributed algorithm is mainly composed of two phases: an adaptation phase and a combination phase. During the adaptation phase, each node updates its local estimate by using all its neighbors' measurements. It is achieved based on a multi-sensor cardinalized probability hypothesis density filter. During the combination phase, each node fuses all its neighbors' local estimates. It is achieved based on a generalized version of covariance intersection technique. Compared to the consensus-based distributed algorithm, the proposed algorithm has two advantages. First, it can provide more accurate and robust tracking results, especially when the detection probability that the sensors detect the targets is low. Second, it has lower communication load because the consensus iterations are not required. Numerical results are provided to illustrate the performance of the proposed algorithm.

INDEX TERMS Diffusion strategy, distributed estimation, multitarget tracking, sensor networks.

I. INTRODUCTION

Multitarget tracking is an important topic in many civilian and military applications [1]–[4]. The objective is to jointly estimate the number of targets and their states from a sequence of noisy and cluttered measurements as well as uncertain associations between targets and measurements. With a set theoretic approach [5], the targets and measurements can be modeled using random finite sets (RFSs), which allows multitarget tracking to be cast in a Bayesian framework. As an approximation of the first order moment, the probability hypothesis density (PHD) filter [6] propagates the multitarget state's intensity function over time. Improving on the PHD filter, the cardinalized PHD (CPHD) filter [7] jointly propagates the intensity function and the cardinality distribution of targets.

Target tracking using a single sensor may face many limitations such as the lack of robustness and accuracy. Recently, the multitarget tracking with sensor networks constructed by multiple sensors [8]–[10] has received much attention. The sensors have sensing, processing and communication abilities. Each sensor generates its local measurements about the targets and they cooperatively estimate the unknowns of interest.

The performance of the sensor information fusion will depend on the collaboration strategy. In the centralized solution [11], all nodes send their local measurements either directly or via a multi-hop relay to a fusion center (FC), and the FC is responsible for the state estimation. It can obtain the optimal estimate since it uses all measurements from all nodes. However, the centralized solution usually requires energy-intensive communications over large distances. It is also less robust to the unreliable network conditions such as sensor node failures. Distributed solution [12] is an attractive alternative. Each sensor only communicates with its neighbors in the distributed solution and there is no FC in the networks. Each sensor performs a local filter and interacts with others to calculate a global state estimate. The distributed solution is in general more robust, may require fewer communications, and allows parallel processing. The final estimation results are available at all nodes.

There are basically two types of distributed solution for sensor information fusion, namely, consensus and diffusion strategies. In the consensus strategies [13]–[18], some quantities that are related to the measurements of each node are exchanged among neighbors by using consensus

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

iterations [19]. The goal is to achieve a consensus among all nodes for computing a desired, e.g., average value. However, a number of consensus iterations at each time step are required such that the consensus of the whole network can be reached, which may require potentially prohibitive communication load between sensors.

Different from the consensus strategies which require the consensus of the whole network to be reached for a number of consensus iterations, the diffusion strategies [20] do not impose this restriction and the consensus iterations are not required. So the estimation performance may be improved and the potentially prohibitive communication load because of the consensus iterations can be avoided. The diffusion strategies at each time step are composed of two phases: an adaptation phase that updates the estimate of each node by using its neighbors' measurements, and a combination phase that fuses the neighbors' estimates. The diffusion strategies may converge faster and reach lower mean-square deviation than the consensus strategies [21].

In this paper, we consider the distributed multitarget tracking by using the diffusion strategies to exchange and fuse the information over networks.

A. RELATED WORKS

The diffusion strategies were originally introduced for the solution of distributed estimation and adaptation problems [20]. They have been adopted to model various forms of distributed estimation problems over networks. Specifically, the basic diffusion based on the least-squares or recursive least-squares methods has been proposed in [22]. In [23], the diffusion Kalman filters were proposed for the linear dynamic state-space models. Many other diffusion algorithms have also been studied for the sequential estimation over time [24]–[27]. For example, Li and Jia [26] proposed a diffusion-based algorithm for the distributed estimation of Markov jump systems and maneuvering target tracking, and it is a distributed single-target tracking problem. The performance analysis of the diffusion strategies has been studied under different conditions [28] as well.

The PHD filter with multiple sensors for multitarget tracking was first derived for the case of two sensors by Mahler [29], [30]. Delande *et al.* [31] generalized it to the case of an arbitrary number of sensors. In [32], Nannuru et al. derived the update equations for the general multisensor CPHD filter as well as a computationally tractable implementation. In [33], Saucan *et al.* derived a multisensor multi-Bernoulli filter for multitarget tracking. These works are basically centralized solution of the multisensor multitarget tracking.

Distributed multisensor multitarget tracking has also been studied. In [34], Üney et al. derived a distributed fusion of PHD, CPHD and Bernoulli filters by combining a generalized version of covariance intersection. In [36], Battistelli et al. developed a consensus CPHD filter that provided a distributed solution, where each node first calculates its local estimate with its own measurements, and then it calls for consensus iterations to achieve global fusion over the network by iterating local fusion among neighbors. More recently, Leonard and Zoubir [37] developed a distributed particle filter implementation of the PHD filter for multitarget tracking. Since a large number of weighted particles are generated at each node and communicated between neighbors for the adaptation step and the combination step in [37], this algorithm requires high computational complexity and communication load.

B. OUR CONTRIBUTIONS

We develop a distributed multitarget tracking algorithm over sensor networks by using the diffusion strategies and the CPHD filter. The proposed distributed algorithm is composed of an adaptation phase and a combination phase. The adaptation phase updates the local estimate of each node with all its neighbors' measurements. It is basically a centralized multitarget tracking with a small set of neighboring nodes. We achieve it based on the general multisensor CPHD filter in [32]. The combination phase improves the local estimate of each node by sharing its estimate with its neighbors. It is a fusion of local estimates among the neighboring nodes. We achieve it based on the generalized version of covariance intersection [34]. Finally, we present a Gaussian mixture implementation of the proposed distributed multitarget tracking algorithm.

The proposed diffusion-based distributed algorithm is somewhat similar to the consensus-based distributed algorithm in [36]. However, compared to the consensus-based algorithm, the proposed algorithm has two differences and advantages.

First, during the adaptation phase, each node in the proposed algorithm updates its local estimate with all its neighbors' measurements (not only its own measurements). It can enrich the statistical knowledge of each node by incorporation of the neighbors' measurements. It can provide more accurate and robust local estimates at each node. Especially, when the detection probability that the nodes detect the targets is low, and if one node does not obtain its own measurements from the targets, the node can still provide correct local estimate about the targets. While, each node in the consensus-based algorithm updates its local estimate only with its own measurements. When the detection probability is low, and if one node does not obtain its own measurements from the targets, the node can not provide correct local estimate about the targets, and the incorrect local estimate will be diffused to the network by the consensus iterations, since the consensus of all local estimates are generally required over network for a number of consensus iterations. When the detection probability that the nodes detect the targets decreases, the performance of consensus-based algorithm will deteriorate, while the proposed algorithm can still provide accurate and robust tracking results.

Second, during the combination phase of the consensus-based algorithm, a number of consensus iterations are required such that the consensus of all the local estimates can be reached over network, while each node in the proposed algorithm fuses its local estimate and all its neighbors' local estimates only once and does not require a number of iterations. So the communication load of the proposed algorithm over networks can be largely reduced compared to the consensus-based algorithm.

On the other hand, compared to the distributed particle filter implementation in [37], our method has much lower computational complexity and communication load since the large number of weighted particles are not required.

C. ORGANIZATION

The rest of this paper is organized as follows. Section II describes the problem formulation and some background. Section III presents the proposed diffusion-based distributed multisensor multitarget tracking algorithm. The Gaussian mixture implementation is presented in Section IV. Some numerical examples are provided in Section V. The paper is concluded in Section VI.

II. PROBLEM FORMULATION AND BACKGROUND

A. SENSOR NETWORK MODEL

We consider a sensor network consisting of a set of R nodes $\mathcal{V} = \{1, \ldots, R\}$. The nodes are geographically dispersed in the surveillance region and each has processing, communication and sensing capabilities. Two nodes can communicate directly with each other if their distance is less than a communication range η_c. The communication structure among nodes is represented using an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. Two nodes r and s, if the edge $(r, s) \in \mathcal{E}$, are said to be connected. The set of nodes connected to a certain node r is called as the neighborhood of node r and is denoted as

$$\mathcal{N}_r = \{s \in \mathcal{V} | (r, s) \in \mathcal{E}\}, \tag{1}$$

where the node r is assumed to be a neighbor of itself. Each node knows its immediate neighbors, but no node knows the global communication structure of the network. It is assumed that the graph or the network is connected, i.e., for any two nodes r and s there exists a sequence of edges (r, t_1), $(t_1, t_2), \ldots, (t_{m-1}, t_m)$, (t_m, s) in $\mathcal{E}$.

B. MULTITARGET TRACKING

We review the multitarget tracking problem formulated in the RFS framework [5]. Suppose that there are N_k targets in the surveillance area at time k and the target states are $x_k^1, \cdots, x_k^{N_k}$ respectively, each taking values in a state space $\mathcal{X} \subset \mathbf{R}^{n_x}$. Then, the multitarget state X_k at time k is defined as the RFS

$$X_k = \{x_k^1, \ldots, x_k^{N_k}\} \in \mathcal{F}(\mathcal{X}), \tag{2}$$

where $\mathcal{F}(\mathcal{X})$ denotes the collections of all finite subsets of $\mathcal{X}$. In the RFS X_k, both the number of targets and the state of each target are not know and must be estimated.

We denote the surviving probability that the target with state $x_k \in X_k$ will continue to exist at time $k + 1$ as $p_v(x_k)$. If the target continues to exist, its state evolves according to the transition distribution $\pi_{k+1|k}(x_{k+1}|x_k)$. Consequently, for a given multitarget state X_k at time k, its behavior at the next time $k + 1$ is described as the RFS

$$X_{k+1} = \left(\bigcup_{x_k \in X_k} S_{k+1|k}(x_k) \right) \bigcup \Gamma_{k+1}. \tag{3}$$

Here the first term $S_{k+1|k}(x_k)$ is the surviving RFS of target that evolves from a target with previous state x_k, which is given by

$$S_{k+1|k}(x_k) = \begin{cases} \{x_{k+1}\}, & \text{with prob. } p_v(x_k), \\ \emptyset, & \text{with prob. } 1 - p_v(x_k), \end{cases} \tag{4}$$

and $x_{k+1} \sim \pi_{k+1|k}(x_{k+1}|x_k)$. The second term $\Gamma_{k+1} = \{b_{k+1}^1, \ldots, b_{k+1}^B\}$ is the RFS of spontaneous births at time $k + 1$. The number B of births is distributed according to a discrete probability distribution $p_b(n)$ and the states b_{k+1}^j are distributed according to a birth density $s_b(x)$.

For a given multitarget state X_k at time k, the rth sensor outputs a set of measurements, which account for detection uncertainty and clutter. We denote the detection probability that the rth sensor detects the target with state x_k as $p_{d,r}(x_k)$, and the probability of a missed detection as $q_{d,r}(x_k) = 1 - p_{d,r}(x_k)$. If it is detected, the measurement is characterized by a likelihood function $\xi_{r,k}(z_{r,k}|x_k)$. Consequently, the multitarget measurements at the rth sensor are given by

$$Z_{r,k} = \left(\bigcup_{x_k \in X_k} \Theta_{r,k}(x_k) \right) \bigcup C_{r,k}. \tag{5}$$

Here the first term $\Theta_{r,k}(x_k)$ is the RFS of measurements generated by the target with state x_k, which is given by

$$\Theta_{r,k}(x_k) = \begin{cases} \{z_{r,k}\}, & \text{with prob. } p_{d,r}(x_k), \\ \emptyset, & \text{with prob. } 1 - p_{d,r}(x_k), \end{cases} \tag{6}$$

and $z_{r,k} \sim \xi_{r,k}(z_{r,k}|x_k)$. The second term $C_{r,k}$ is the RFS of clutter measurements as $C_{r,k} = \{c_{r,k}^1, \ldots, c_{r,k}^{m_{r,k}}\}$. For the rth node, the number of clutters $m_{r,k}$ is distributed according to a discrete probability distribution $p_{r,c}(n)$ and the clutters $c_{r,k}^j$ are generated according to a clutter density $s_{r,c}(x)$. Denote $Z_k \triangleq \{Z_{1,k}, \cdots, Z_{R,k}\}$ as the measurements from all sensors at time k.

We denote the transition of multitarget state X_k from time k to $k+1$ by a multitarget transition density $f_{k+1|k}(X_{k+1}|X_k)$, which describes the births and deaths of targets, and the time motion of surviving targets. The multitarget measurement Z_k is described by a multitarget likelihood $g_k(Z_k|X_k)$, which characterizes the target generated measurements and clutters. The goal is to construct the multitarget posterior density $p_k(X_k|Z_{1:k})$, where $Z_{1:k} \triangleq \{Z_1, \ldots, Z_k\}$ is the cumulative measurements from all sensors up to time k. The multitarget posterior is propagated with the Bayesian recursion as

$$p_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X) p_{k-1}(X|Z_{1:k-1}) \delta X, \tag{7}$$

$$p_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X)p_{k|k-1}(X|Z_{1:k-1})\delta X}, \quad (8)$$

where $\int \cdot \delta X$ denotes the set integral defined by

$$\int p(X)\delta X = p(\emptyset)+\sum_{n=1}^{\infty}\frac{1}{n!}\int p(\{x_1, \ldots, x_n\})dx_1 \ldots dx_n. \quad (9)$$

However, the multitarget Bayes recursion usually results in the multiple integrations because of the set integral, so it is intractable in most practical applications. Some computationally cheap approximations have been derived, such as the PHD filer [6] and the CPHD filter [7]. The PHD is defined as the first order moment of the multitarget density, also called as intensity function, $\nu_k(x)$, which gives the expected number of targets when integrated over the region R^{n_x} as

$$\int_{R^{n_x}} \nu_k(x)dx = E\{|X_k \cap R^{n_x}|\}. \quad (10)$$

The PHD and CPHD filter construct the prediction and update equations in terms of the intensity function. The PHD filter assumes that the number of targets is Poisson-distributed, while the CPHD filter uses a general cardinality distribution. Besides the intensity function, the CPHD filter also propagates the cardinality distribution of targets. The original PHD and CPHD filter are only derived for the case of single sensor.

III. DIFFUSION-BASED DISTRIBUTED TRACKING

The proposed diffusion-based distributed algorithm are composed of two phases at each time step, i.e., adaptation phase and combination phase. The adaptation phase updates the local estimate of a node with all its neighbors' measurements, which can enrich the statistical knowledge of each node. Even when the detection probability that the nodes detect the targets is low, and if one node does not obtain its own measurements from the targets, the node can still provide correct local estimate about the targets. The combination phase diffuses and merges the neighbors' local estimates at each node. The diffusion of local estimates among neighbors can further improve the estimation performance of the whole network. Our method adopts the similar information processing procedure as the diffusion Kalman filters in [23]–[28].

Our distributed algorithm is based on the CPHD filter, which propagates the cardinality distribution and the intensity function of the multitarget state with the Bayesian recursion over time.

Suppose that each node $r \in \mathcal{V}$ at time $k-1$ has its estimate of the posterior cardinality distribution, denoted as $\rho_{r,k-1}(n)$, and its estimate of posterior intensity function, denoted as $\nu_{r,k-1}(x)$. With the arrival of new measurement Z_k of the sensor network at time k, each node r constructs its estimate of the posterior cardinality distribution and intensity function, denoted as $\rho_{r,k}(n)$ and $\nu_{r,k}(x)$, respectively. The process will be described below in terms of the Bayesian recursion, which includes three phases, i.e., prediction, adaptation and combination.

A. PREDICTION PHASE

From its previous posteriors $\rho_{r,k-1}(n)$ and $\nu_{r,k-1}(x)$ at time $k-1$, each node $r \in \mathcal{V}$ in the network at time k calculates the predicted cardinality distribution, denoted as $\rho_{r,k|k-1}(n)$, and the predicted intensity function, denoted as $\nu_{r,k|k-1}(x)$. All nodes can perform the prediction phase independently and in parallel because no information exchange is required in this phase. So the prediction step for the multisensor case is same as that for the case of single sensor.

The predicted cardinality distribution at the node r is given by [7]

$$\rho_{r,k|k-1}(n) = \sum_{j=0}^{n} p_b(n-j)\chi_{r,k}(j), \quad (11)$$

where

$$\chi_{r,k}(j) = \sum_{h=j}^{\infty} C_h^j \frac{\langle \nu_{r,k-1}, p_v\rangle^j \langle \nu_{r,k-1}, 1-p_v\rangle^{h-j}}{\langle \nu_{r,k-1}, 1\rangle^h} \times \rho_{r,k-1}(h), \quad (12)$$

and C_h^j is the binomial coefficient $\frac{h!}{j!(h-j)!}$, $\langle \cdot, \cdot \rangle$ is the inner product between two real-value functions f and g defined as

$$\langle f, g\rangle = \int f(x)g(x)dx. \quad (13)$$

The predicted intensity function is given by [7]

$$\nu_{r,k|k-1}(x)=s_b(x)+\int \pi_{k|k-1}(x|w)p_v(w)\nu_{r,k-1}(w)dw, \quad (14)$$

The normalized predicted intensity function is denoted as

$$\mu_{r,k|k-1}(x) = \frac{\nu_{r,k|k-1}(x)}{\bar{n}_{r,k|k-1}}, \quad (15)$$

where $\bar{n}_{r,k|k-1} = \sum_{n=1}^{\infty} n\rho_{r,k|k-1}(n)$.

B. ADAPTATION PHASE

The adaptation phase updates the local estimate of each node with all its neighbors' measurements (not only its own measurements). The aim is to enrich the statistical knowledge of each node by incorporation of the neighbors' measurements. Denote $Z_{\mathcal{N}_r,k} \triangleq \{Z_{s,k} : s \in \mathcal{N}_r\}$ which includes the measurements of all neighbors of the node r at time k. The node r updates its local estimate with $Z_{\mathcal{N}_r,k}$ and its local prior information. It is actually a centralized multisensor multitarget tracking problem with a small set of neighboring nodes.

A multisensor PHD filter was first proposed in [29] and [30] for two sensors. It was extended to an arbitrary number of sensors in [31]. The update procedure was further generalized to the multisensor CPHD filter in [32] with an efficient approximate implementation. A multisensor multi-Bernoulli filter was also derived in [33] for the multitarget tracking. We will achieve the adaptation phase at each node by means of the multisensor CPHD filter in [32].

The update process of the multisensor CPHD filters involves partitioning the measurement set $Z_{\mathcal{N}_r,k}$ into disjoint subsets. We first introduce notations similar to [32]. For the notation convenience, we rewrite the neighborhood of node r as

$$\mathcal{N}_r = \{1, 2, \ldots, d_r\}, \tag{16}$$

where s $(1 \leq s \leq d_r)$ means the sth neighbor of the node r, and $d_r \triangleq |\mathcal{N}_r|$ is the number of neighbors in $\mathcal{N}_r$. We rewrite the measurement set $Z_{\mathcal{N}_r,k}$ as

$$Z_{\mathcal{N}_r,k} = \{Z_{1,k}, \ldots, Z_{d_r,k}\}, \tag{17}$$

where $Z_{s,k}$ $(1 \leq s \leq d_r)$ denotes the measurements of the sth neighbor of the node r.

Let $W_s \subset Z_{s,k}$ be a measurement subset that contains at most one measurement from the measurements $Z_{s,k}$ of the sth neighbor, i.e., $|W_s| \leq 1$. To keep the expression compact, we drop the time index k in W_s. Then an ordered subset of measurement set is constructed as $W_{1:d_r} \triangleq \{W_1, \ldots, W_{d_r}\}$, where each subset includes at most one measurement from each neighbor. It can be interpreted as the measurements made by all neighbors in the set $\mathcal{N}_r$ from one same target. Each subset $W_{1:d_r}$ can also be associated with a set of indices as $T_{W_{1:d_r}} = \{(s,l) | z^l_{s,k} \in W_s, s = 1, \ldots, d_r\}$, while (s,l) means that $z^l_{s,k}$ is the lth measurement in $Z_{s,k}$ of the sth neighbor. It is said that two subsets $W^i_{1:d_r} \triangleq \{W^i_1, \ldots, W^i_{d_r}\}$ and $W^j_{1:d_r} \triangleq \{W^j_1, \ldots, W^j_{d_r}\}$ are disjoint if $W^i_s \bigcap W^j_s = \emptyset$ for $s = 1, \ldots, d_r$. For a number of n disjoint subsets $W^1_{1:d_r}$, $\cdots$, $W^n_{1:d_r}$, we define $V = \{V_1, \ldots, V_{d_r}\}$, where $V_s = Z_{s,k} \backslash (\cup^n_{j=1} W^j_s)$ $(1 \leq s \leq d_r)$. The set V can be interpreted as the collection of clutter points made by all neighbors.

For a given integer $n \geq 0$, we define

$$P = \{W^1_{1:d_r}, \ldots, W^n_{1:d_r}, V\} \tag{18}$$

as a partition of the measurements $Z_{\mathcal{N}_r,k}$. Each element of the partition P can be interpreted as the measurements generated by one target (i.e., each $W^j_{1:d_r}$ for one target), or generated by the clutter (i.e., V for the clutter). Let $\mathcal{P}$ denote the set of all partitions P of $Z_{\mathcal{N}_r,k}$.

With the predicted cardinality distribution $\rho_{r,k|k-1}(n)$ and intensity function $\mu_{r,k|k-1}(x)$ as well as the neighbors' measurements $Z_{\mathcal{N}_r,k}$, the update equation of the cardinality distribution and the intensity function at the rth node are given by [32]

$$\widetilde{\nu}_{r,k}(x) = \left[\alpha_0 \prod_{s=1}^{d_r} q_{d,s}(x) + \sum_{P\in\mathcal{P}} \alpha_P \left(\sum_{W\in P} \varrho_W(x)\right)\right] \times \mu_{r,k|k-1}(x), \tag{19}$$

$$\widetilde{\rho}_{r,k}(n) = \left[\frac{\displaystyle\sum_{\substack{P\in\mathcal{P}\\|P|\leq n+1}} k_p \frac{n!}{(n-|P|+1)!}\gamma^{n-|P|+1} \prod_{W\in P} d_W}{\displaystyle\sum_{P\in\mathcal{P}} k_p M^{(|P|-1)}(\gamma) \prod_{W\in P} d_W}\right] \times \rho_{r,k|k-1}(n). \tag{20}$$

where

$$\varrho_W(x) = \frac{\left(\displaystyle\prod_{(s,l)\in T_W} p_{d,s}(x)\xi_{s,k}(z^l_{s,k}|x)\right) \displaystyle\prod_{(t,*)\notin T_W} q_{d,t}(x)}{D}, \tag{21}$$

and the denominator D in the above equation has the form

$$D = \int \mu_{r,k|k-1}(x) \left(\prod_{(s,l)\in T_W} p_{d,s}(x)\xi_{s,k}(z^l_{s,k}|x)\right) \times \prod_{(t,*)\notin T_W} q_{d,t}(x)dx. \tag{22}$$

The quantities α_0, α_P, γ, k_p and d_W have cumbersome expressions, which are given in Appendix, and $M(t)$ is the probability generating function of the predicted cardinality distribution defined as

$$M(t) = \sum_{n=0}^{\infty} t^n \rho_{r,k|k-1}(n), \tag{23}$$

and its vth order derivative $M^{(v)}(t) = \frac{d^v M}{dt^v}(t)$. To keep the expressions compact, we drop the index in $W_{1:d_r}$ and write it as W in the above equations.

C. COMBINATION PHASE

We denote the estimated multitarget density of the sth neighbor of node r as $\widetilde{p}_{s,k}(X)$ after the adaptation phase. If the multitarget state is modeled as an i.i.d cluster process, the multitarget density will has the form as [5]

$$\widetilde{p}_{s,k}(X) = |X|!\widetilde{\rho}_{s,k}(|X|) \prod_{x\in X} \widetilde{\mu}_{s,k}(x), \tag{24}$$

where $\widetilde{\mu}_{s,k}(x)$ is the normalized intensity function of $\widetilde{\nu}_{s,k}(x)$ in (19). These estimates can be shared by the node r because of the direct communication link between nodes r and s.

During the combination phase, the node r improves its local estimate by fusing these shared estimates. It is a multisensor multitarget fusion problem over the set of neighboring nodes. There have been several interesting contributions about the multisensor multitarget fusion, such as the generalized covariance intersection for multitarget density in [38].

We adopt the Kullback-Leibler optimal fusion, which is to construct a posterior intensity at the node r, denoted as $p_{r,k}(X)$, whose divergence from the posterior $\widetilde{p}_{s,k}(X)$ of all the neighbors $s \in \mathcal{N}_r$ is minimal. The weighted Kullback-Leibler divergence (KLD) for multitarget density is defined as [34], [35]

$$\sum_{s\in\mathcal{N}_r} \varepsilon_s D_{KL}(p_{r,k}||\widetilde{p}_{s,k}), \tag{25}$$

where

$$D_{KL}(p_{r,k}||\widetilde{p}_{s,k}) = \int p_{r,k}(X) \log \frac{p_{r,k}(X)}{\widetilde{p}_{s,k}(X)} \delta X, \tag{26}$$

and the weights ε_s satisfy $\varepsilon_s \geq 0$ and $\sum_{s\in\mathcal{N}_r} \varepsilon_s = 1$.

The notion of KLD has been analyzed in the context of single target density, and it was extended to multitarget density in [34] and [35]. It is a consistent information-theoretic relative entropy measure. The weighting parameters ε_s may reflect the statistical properties, e.g., their covariances, of the multitarget density of the nodes. We will discuss the selection of the value of weights ε_s in Section IV-E.

In the KLD optimal fusion for the multitarget density, the multitarget density $p_{r,k}(X)$ that minimizes the weighted KLD in (25) has the form [36]

$$p_{r,k}(X) = \frac{\prod\limits_{s\in\mathcal{N}_r}[\widetilde{p}_{s,k}(X)]^{\varepsilon_s}}{\int \prod\limits_{s\in\mathcal{N}_r}[\widetilde{p}_{s,k}(X)]^{\varepsilon_s}\delta X}. \quad (27)$$

It is the exponential mixture of the densities. It has shown that it can prevent double counting in arbitrary network topologies [39].

If the multitarget state is modeled as an i.i.d cluster process, it has been shown that the fusion density in (27) has the form as [34]

$$p_{r,k}(X) = |X|!\rho_{r,k}(|X|)\prod_{x\in X}\mu_{r,k}(x), \quad (28)$$

where

$$\mu_{r,k}(x) = \frac{\prod\limits_{s\in\mathcal{N}_r}[\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s}}{\int \prod\limits_{s\in\mathcal{N}_r}[\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s}dx}, \quad (29)$$

$$\rho_{r,k}(n) = \frac{\prod\limits_{s\in\mathcal{N}_r}[\widetilde{\rho}_{s,k}(n)]^{\varepsilon_s}\left\{\int \prod\limits_{s\in\mathcal{N}_r}[\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s}dx\right\}^n}{\sum\limits_{j=0}^{\infty}\prod\limits_{s\in\mathcal{N}_r}[\widetilde{\rho}_{s,k}(j)]^{\varepsilon_s}\left\{\int \prod\limits_{s\in\mathcal{N}_r}[\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s}dx\right\}^j}. \quad (30)$$

From the equations (29) and (30), the fused cardinality distribution $\rho_{r,k}(n)$ and the intensity function $\mu_{r,k}(x)$ at the node r can be directly computed from the cardinality distribution $\widetilde{\rho}_{s,k}(n)$ and the intensity function $\widetilde{\mu}_{s,k}(x)$ of all its neighbors $s \in \mathcal{N}_r$.

IV. GAUSSIAN MIXTURE IMPLEMENTATION

We derive a closed-form solution to the distributed multisensor multitarget tracking for the special case of linear Gaussian dynamical and measurement models. We consider the following assumptions on the targets and sensors.

A.1: Each target evolves independently and follows a linear Gaussian dynamical model as

$$\pi_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; A_k x_k, Q_k), \quad (31)$$

where A_k is the state transition matrix and Q_k is the process noise covariance.

A.2: Each sensor generates measurements independently and follows a linear Gaussian measurement model as

$$\xi_{r,k}(z_{r,k}|x_k) = \mathcal{N}(z_{r,k}; B_{r,k}x_k, R_{r,k}), \quad (32)$$

where $B_{r,k}$ is the measurement matrix and $R_{r,k}$ is the measurement noise covariance.

A.3: The survival probability of each target is state independent

$$p_v(x) = p_v. \quad (33)$$

A.4: The detection probability at each sensor is state independent

$$p_{d,r}(x) = p_{d,r}. \quad (34)$$

A.5: The intensity of the birth is a Gaussian mixture of the form

$$s_b(x) = \sum_{i=1}^{N_b}\omega_b^{(i)}\mathcal{N}(x; m_b^{(i)}, P_b^{(i)}). \quad (35)$$

It is easy to prove the following standard results for Gaussian densities that will be used in the following derivation.

R.1: Given A, Q, m, and P with appropriate dimensions, where Q and P are positive definite, then [40], [41]

$$\int \mathcal{N}(x; Aw, Q)\mathcal{N}(w; m, P)dw = \mathcal{N}(x; Am, Q+APA^T). \quad (36)$$

R.2: Given B, R, m, and P with appropriate dimensions, where R and P are positive definite, then [40], [41]

$$\mathcal{N}(z; Bx, R)\mathcal{N}(x; m, P) = q(z)\mathcal{N}(x; \widetilde{m}, \widetilde{P}), \quad (37)$$

where $q(z) = \mathcal{N}(z; Bm, R + BPB^T)$, $\widetilde{m} = m + K(z - Bm)$, $\widetilde{P} = (I - KB)P$ and $K = PB^T(BPB^T + R)^{-1}$.

R.3: The power of a Gaussian component is a Gaussian component [36]

$$[\alpha\mathcal{N}(x; m, P)]^w = \alpha^\omega k(\omega, P)\mathcal{N}(x; m, P/\omega), \quad (38)$$

where $k(\omega, P) = \frac{[\det(2\pi P/\omega)]^{1/2}}{[\det(2\pi P)]^{\omega/2}}$.

R.4: The product of two Gaussian components is a Gaussian component [36]

$$[\alpha_1\mathcal{N}(x;m_1,P_1)][\alpha_2\mathcal{N}(x;m_2,P_2)] = \alpha_{12}\mathcal{N}(x;m_{12},P_{12}), \quad (39)$$

where $P_{12} = (P_1^{-1}+P_2^{-1})^{-1}$, $m_{12} = P_{12}(P_1^{-1}m_1+P_2^{-1}m_2)^{-1}$ and $\alpha_{12} = \alpha_1\alpha_2\mathcal{N}(m_1 - m_2; 0, P_1 + P_2)$.

A. GAUSSIAN MIXTURE IMPLEMENTATION OF THE PREDICTION PHASE

Suppose that the posterior intensity $\nu_{r,k-1}(x)$ of the node r at time $k - 1$ is a Gaussian mixture of the form

$$\nu_{r,k-1}(x) = \sum_{i=1}^{N_{r,k-1}}\omega_{r,k-1}^{(i)}\mathcal{N}(x; m_{r,k-1}^{(i)}, P_{r,k-1}^{(i)}). \quad (40)$$

Then with the equations (31), (33), (35) and (36), the predicted intensity $\nu_{r,k|k-1}(x)$ in (14) is also a Gaussian mixture of the form

$$\nu_{r,k|k-1}(x) = \sum_{i=1}^{N_{r,k|k-1}} \omega^{(i)}_{r,k|k-1}\mathcal{N}(x; m^{(i)}_{r,k|k-1}, P^{(i)}_{r,k|k-1}), \quad (41)$$

where $N_{r,k|k-1} = N_b + N_{r,k-1}$, and

$$\omega^{(i)}_{r,k|k-1} = \omega^{(i)}_b, m^{(i)}_{r,k|k-1} = m^{(i)}_b, P^{(i)}_{r,k|k-1} = P^{(i)}_b, \quad (42)$$

for $i = 1, \ldots, N_b$, and

$$\omega^{(i+N_b)}_{r,k|k-1} = p_v\omega^{(i)}_{r,k-1}, \quad m^{(i+N_b)}_{r,k|k-1} = A_k m^{(i)}_{r,k-1}, \quad (43)$$

$$P^{(i+N_b)}_{r,k|k-1} = Q_k + A_k P^{(i)}_{r,k-1}A^T_k, \quad (44)$$

for $i = 1, \ldots, N_{r,k-1}$.

Moreover, from the posterior cardinality distribution $\rho_{r,k-1}(n)$ of the node r at time $k-1$ and the equation (33), the predicted cardinality distribution $\rho_{r,k|k-1}(n)$ in (11) can be simplified as

$$\rho_{r,k|k-1}(n)=\sum_{j=0}^{n} p_b(n-j)\sum_{h=j}^{\infty} C^j_h p^j_v(1-p_v)^{h-j}\rho_{r,k-1}(h). \quad (45)$$

From the above results, the normalized predicted intensity function $\mu_{r,k|k-1}(x)$ in (15) is also a Gaussian mixture form and we denote it as

$$\mu_{r,k|k-1}(x)= \sum_{i=1}^{N_{r,k|k-1}} \varpi^{(i)}_{r,k|k-1}\mathcal{N}(x; m^{(i)}_{r,k|k-1}, P^{(i)}_{r,k|k-1}). \quad (46)$$

B. GAUSSIAN MIXTURE IMPLEMENTATION OF THE ADAPTION PHASE

With the equations (32) and (37), the expression $\xi_{s,k}(z^l_{s,k}|x)$ $\mu_{r,k|k-1}(x)$ is a Gaussian mixture form as

$$\xi_{s,k}(z^l_{s,k}|x)\mu_{r,k|k-1}(x)=\sum_{i=1}^{N_{r,k|k-1}}\widetilde{\omega}^{(i)}_{rs,k|k}\mathcal{N}(x;\widetilde{m}^{(i)}_{rs,k|k}, \widetilde{P}^{(i)}_{rs,k|k}), \quad (47)$$

where

$$\widetilde{\omega}^{(i)}_{rs,k|k} = \varpi^{(i)}_{r,k|k-1} \times\mathcal{N}(z^l_{s,k};B_{s,k}m^{(i)}_{r,k|k-1},R_{s,k}+B_{s,k}P^{(i)}_{r,k|k-1}B^T_{s,k}), \quad (48)$$

$$\widetilde{m}^{(i)}_{rs,k|k} = m^{(i)}_{r,k|k-1} + K(z^l_{s,k} - B_{s,k}m^{(i)}_{r,k|k-1}), \quad (49)$$

$$\widetilde{P}^{(i)}_{rs,k|k} = (I - KB_{s,k})P^{(i)}_{r,k|k-1}, \quad (50)$$

$$K = P^{(i)}_{r,k|k-1}B^T_{s,k}(B_{s,k}P^{(i)}_{r,k|k-1}B^T_{s,k} + R_{s,k})^{-1}. \quad (51)$$

Then, under the assumption in (34) that the detection probability is constant, the expression $\varrho_W(x)\mu_{r,k|k-1}(x)$ in (19) is also a Gaussian mixture form, where the integration in (22) can be analytically calculated with the equation (36).

Because the detection probability is constant, the quantities α_0 and α_P in Appendix can be easily calculated. So the update posterior intensity function $\widetilde{\nu}_{r,k}(x)$ in (19) is a Gaussian mixture form. We express it as

$$\widetilde{\nu}_{r,k}(x) = \sum_{i=1}^{\widetilde{N}_{r,k}} \widetilde{\omega}^{(i)}_{r,k}\mathcal{N}(x; \widetilde{m}^{(i)}_{r,k}, \widetilde{P}^{(i)}_{r,k}), \quad (52)$$

and its normalized intensity function as

$$\widetilde{\mu}_{r,k}(x) = \sum_{i=1}^{\widetilde{N}_{r,k}} \widetilde{\varpi}^{(i)}_{r,k}\mathcal{N}(x; \widetilde{m}^{(i)}_{r,k}, \widetilde{P}^{(i)}_{r,k}). \quad (53)$$

However, the number of all partitions P in $\mathcal{P}$ from all possible measurement subsets is prohibitively large, so the analytical implementation is still numerically infeasible. In [32], it has shown an approximation to overcome this limitation, which follows two-step greedy approximation with the Gaussian mixture framework. It does not construct all possible measurement subsets, but only sequentially keeps a few best-scoring measurement subsets.

C. GAUSSIAN MIXTURE IMPLEMENTATION OF THE COMBINATION PHASE

There have been considerable efforts in the area of distributed fusion of Gaussian mixtures, such as the pairwise component covariance intersection [42], pseudo-Chernoff fusion [36], [43], and sigma-point based Chernoff fusion [44]. We achieve the pseudo-Chernoff fusion method in [36] and [43] for the combination phase here because of its simplicity and performance.

We first consider to fuse the intensity functions of two neighbors $s \in \mathcal{N}_r$ and $t \in \mathcal{N}_r$. It is assumed that the estimated intensity functions of the node s and t after the adaption phase are

$$\widetilde{\mu}_{j,k}(x) = \sum_{i=1}^{\widetilde{N}_{j,k}} \widetilde{\varpi}^{(i)}_{j,k}\mathcal{N}(x; \widetilde{m}^{(i)}_{j,k}, \widetilde{P}^{(i)}_{j,k}), j = s, t. \quad (54)$$

For two nodes, the expression in (29) is described as

$$\widetilde{\mu}_{st,k}(x) = \frac{[\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s}[\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t}}{\displaystyle\int [\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s}[\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t}dx}. \quad (55)$$

Although $\widetilde{\mu}_{j,k}(x)$ $(j = s, t)$ is a Gaussian mixture form, the power of $\widetilde{\mu}_{j,k}(x)$, i.e., $[\widetilde{\mu}_{j,k}(x)]^{\varepsilon_j}$ in (55), is not a Gaussian mixture form. Under the condition that the cross-products of the different terms in the Gaussian mixture are negligible, the power of the Gaussian mixture is approximated with another Gaussian mixture by using the first order series expansion [43]

$$[\widetilde{\mu}_{j,k}(x)]^{\varepsilon_j} = \left[\sum_{i=1}^{\widetilde{N}_{j,k}} \widetilde{\varpi}^{(i)}_{j,k}\mathcal{N}(x; \widetilde{m}^{(i)}_{j,k}, \widetilde{P}^{(i)}_{j,k})\right]^{\varepsilon_j}$$

$$\approx \sum_{i=1}^{\widetilde{N}_{j,k}} \left[\widetilde{\varpi}^{(i)}_{j,k}\mathcal{N}(x; \widetilde{m}^{(i)}_{j,k}, \widetilde{P}^{(i)}_{j,k})\right]^{\varepsilon_j}$$

$$= \sum_{i=1}^{\widetilde{N}_{j,k}} (\widetilde{\varpi}_{j,k}^{(i)})^{\varepsilon_j} \kappa(\varepsilon_j, \widetilde{P}_{j,k}^{(i)}) \mathcal{N}\left(x; \widetilde{m}_{j,k}^{(i)}, \frac{\widetilde{P}_{j,k}^{(i)}}{\varepsilon_j}\right), \quad (56)$$

where it has applied the result in (38).

Substituting (56) into (55), we will have

$$\widetilde{\mu}_{st,k}(x) = \frac{\sum_{i=1}^{\widetilde{N}_{s,k}} \sum_{j=1}^{\widetilde{N}_{t,k}} \widetilde{\varpi}_{st,k}^{(ij)} \mathcal{N}\left(x; \widetilde{m}_{st,k}^{(ij)}, \widetilde{P}_{st,k}^{(ij)}\right)}{\sum_{i=1}^{\widetilde{N}_{s,k}} \sum_{j=1}^{\widetilde{N}_{t,k}} \widetilde{\varpi}_{st,k}^{(ij)}}, \quad (57)$$

where

$$\widetilde{P}_{st,k}^{(ij)} = \left[\varepsilon_s (\widetilde{P}_{s,k}^{(i)})^{-1} + \varepsilon_t (\widetilde{P}_{t,k}^{(j)})^{-1}\right]^{-1}, \quad (58)$$

$$\widetilde{m}_{st,k}^{(ij)} = \widetilde{P}_{st,k}^{(ij)} \left[\varepsilon_s (\widetilde{P}_{s,k}^{(i)})^{-1} \widetilde{m}_{s,k}^{(i)} + \varepsilon_t (\widetilde{P}_{t,k}^{(j)})^{-1} \widetilde{m}_{t,k}^{(j)}\right], \quad (59)$$

$$\begin{aligned} \widetilde{\varpi}_{st,k}^{(ij)} &= (\widetilde{\varpi}_{s,k}^{(i)})^{\varepsilon_s} (\widetilde{\varpi}_{t,k}^{(j)})^{\varepsilon_t} \kappa(\varepsilon_s, \widetilde{P}_{s,k}^{(i)}) \kappa(\varepsilon_t, \widetilde{P}_{t,k}^{(j)}) \\ &\quad \times \mathcal{N}\left(\widetilde{m}_{s,k}^{(i)} - \widetilde{m}_{t,k}^{(j)}; 0, \frac{\widetilde{P}_{s,k}^{(i)}}{\varepsilon_s} + \frac{\widetilde{P}_{t,k}^{(j)}}{\varepsilon_t}\right), \end{aligned} \quad (60)$$

where it has applied the result in (39).

Moreover, the integral for (30) is approximated as

$$\int [\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s} [\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t} dx \approx \sum_{i=1}^{\widetilde{N}_{s,k}} \sum_{j=1}^{\widetilde{N}_{t,k}} \widetilde{\varpi}_{st,k}^{(ij)}. \quad (61)$$

Sequentially, if there are three neighbors s, t and y of node r, then the fusion in (29) is described as

$$\widetilde{\mu}_{sty,k}(x) = \frac{[\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s} [\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t} [\widetilde{\mu}_{y,k}(x)]^{\varepsilon_y}}{\int [\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s} [\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t} [\widetilde{\mu}_{y,k}(x)]^{\varepsilon_y} dx}. \quad (62)$$

The fusion can be rewritten as

$$\widetilde{\mu}_{sty,k}(x) = \frac{\widetilde{\mu}_{st,k}(x) [\widetilde{\mu}_{y,k}(x)]^{\varepsilon_y}}{\int \widetilde{\mu}_{st,k}(x) [\widetilde{\mu}_{y,k}(x)]^{\varepsilon_y} dx}, \quad (63)$$

and the integral for (30) is expressed as

$$\begin{aligned} &\int [\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s} [\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t} [\widetilde{\mu}_{y,k}(x)]^{\varepsilon_y} dx \\ &\quad = \int [\widetilde{\mu}_{s,k}(x)]^{\varepsilon_s} [\widetilde{\mu}_{t,k}(x)]^{\varepsilon_t} dx \cdot \int \widetilde{\mu}_{st,k}(x) [\widetilde{\mu}_{y,k}(x)]^{\varepsilon_y} dx. \end{aligned} \quad (64)$$

So with the pairwise fusion, the fusion can be easily extended to $d_r \geq 3$ neighbors.

After the fusion, the number of Gaussian components will increase without bound. It can be simplified with a pruning and merging step in [40]. The basic idea is to discard components with negligible weights and merge components that are close together.

We denote the final estimate of the normalized intensity of node r as

$$\mu_{r,k}(x) = \sum_{i=1}^{N_{r,k}} \varpi_{r,k}^{(i)} \mathcal{N}(x; m_{r,k}^{(i)}, P_{r,k}^{(i)}). \quad (65)$$

With the estimated cardinality distribution $\rho_{r,k}(n)$ in (30), the intensity function $\nu_{r,k}(x)$ of node r can be easily calculated from $\mu_{r,k}(x)$, and we denote it as

$$\nu_{r,k}(x) = \sum_{i=1}^{N_{r,k}} \omega_{r,k}^{(i)} \mathcal{N}(x; m_{r,k}^{(i)}, P_{r,k}^{(i)}). \quad (66)$$

D. MULTITARGET STATE EXTRACTION

With the estimated cardinality distribution $\rho_{r,k}(n)$, the node r can estimate the number of targets via a maximum a posterior (MAP) estimator [41] as

$$\widehat{n}_{r,k} = \arg\max_{n} \rho_{r,k}(n). \quad (67)$$

Then, the state extraction in [41] is implemented to extract the $\widehat{n}_{r,k}$ local maxima of the intensity function $\nu_{r,k}(x)$ as the estimated states of the $\widehat{n}_{r,k}$ targets.

E. CHOOSING THE WEIGHTING PARAMETERS

The weighting parameters ε_s in (27) control the relative weighting on $\widetilde{p}_{s,k}(X)$. Motivated by the original covariance intersection [45] which is for the case of Gaussian distribution, we present a suboptimal non-iterative method for fast choosing the weighting parameters in the case of Gaussian mixture.

From the intensity function $\widetilde{\nu}_{r,k}(x)$ in (52) before the combination phase, each component $\mathcal{N}(x; \widetilde{m}_{r,k}^{(i)}, \widetilde{P}_{r,k}^{(i)})$ in the Gaussian mixture can be integrated as the estimate of one target with weight $\widetilde{\omega}_{r,k}^{(i)}$. Since the trace of covariance matrix $\widetilde{P}_{r,k}^{(i)}$ provides a scalar measure of the estimation uncertainty of the target, the estimation uncertainty if the intensity function $\widetilde{\nu}_{r,k}(x)$ in Gaussian mixture form is used to estimate the states of multiple targets can be expressed as

$$\Lambda_i \triangleq \sum_{i=1}^{\widetilde{N}_{r,k}} \widetilde{\omega}_{r,k}^{(i)} \mathrm{tr}(\widetilde{P}_{r,k}^{(i)}). \quad (68)$$

The constraint for choosing the weighting parameters will be selected such that $\Lambda_s = \Lambda_t$ implies $\varepsilon_s = \varepsilon_t$, and $\Lambda_s / \Lambda_t \to 0$ implies $\varepsilon_t = 0$ [45]. The constraint is satisfied by

$$\varepsilon_s \Lambda_s - \varepsilon_t \Lambda_t = 0, \quad s, t \in \mathcal{N}_r. \quad (69)$$

The largest linearly independent subset in the above system can be described as

$$\varepsilon_s \Lambda_s - \varepsilon_{s+1} \Lambda_{s+1} = 0, \quad s = 1, \ldots, d_r - 1. \quad (70)$$

In addition, the nonnegative weighting parameters $\varepsilon_1, \ldots, \varepsilon_{d_r}$ satisfy

$$\varepsilon_1 + \cdots + \varepsilon_{d_r} = 1. \quad (71)$$

Applying Cramer's rule, the weighting parameters that satisfy the linear system (70) and (71) are

$$\varepsilon_s = \frac{1/\Lambda_s}{\sum_{i=1}^{d_r} 1/\Lambda_i}. \quad (72)$$

F. THE COMPLEXITY OF ALGORITHMS

We compare the communication load of the proposed diffusion-based distributed algorithm with that of the consensus-based distributed algorithm in [36].

During the combination phase of the consensus-based algorithm, a number of consensus iterations are required such that the consensus of all the local estimates can be reached over networks. At each iteration step of the consensus iterations, each node transmits its normalized intensity function, with the same form as $\mu_{r,k}(x)$ in (65) to its neighbors, which includes the scale $\varpi_{r,k}^{(i)}$, the vector $m_{r,k}^{(i)} \in \mathbf{R}^{n_x}$, and the symmetric matrix $P_{r,k}^{(i)} \in \mathbf{R}^{n_x \times n_x}$ for $i = 1, \ldots, N_{r,k}$ (totally $[1 + n_x + n_x(n_x + 1)/2] \times N_{r,k}$ scales), as well as the cardinality distribution $\rho_{r,k}(n)$. So the total communication load of consensus iterations is $[1+n_x+n_x(n_x+1)/2] \times N_{r,k} \times N_c$ scales and N_c times of $\rho_{r,k}(n)$, where N_c denotes the number of consensus iterations. While, for the proposed algorithm, each node transmits its normalized intensity function and the cardinality distribution to its neighbors only once, so its total communication load is $[1+n_x+n_x(n_x+1)/2] \times N_{r,k}$ scales, and one time of $\rho_{r,k}(n)$.

During the adaptation phase, each node in the proposed algorithm needs transmit its measurements to its neighbors and receive the measurements from its neighbors. It is the added communication load compared to the consensus-based algorithm because each node in the consensus-based algorithm updates its local estimate only with its own measurements. Since each node r has its estimate about the multitarget state at each time step, it can compute the predicted intensity $\mu_{r,k|k-1}(x)$ in (46) before the adaptation phase. Then it constructs a measurement validation region [46] from $\mu_{r,k|k-1}(x)$ as

$$\mathcal{R}_{r,k} = \bigcup_{i=1}^{N_{r,k|k-1}} \mathcal{R}_{r,k}^{(i)}, \tag{73}$$

$$\mathcal{R}_{r,k}^{(i)} = \{z \in Z_{r,k} : (z - \bar{z}_{r,k}^{(i)})^T (S_{r,k}^{(i)})^{-1} (z - \bar{z}_{r,k}^{(i)}) \leqslant \gamma\}, \tag{74}$$

where γ is the gate threshold, $\bar{z}_{r,k}^{(i)} = B_{r,k} m_{r,k|k-1}^{(i)}$ is the predicted measurement mean and $S_{r,k}^{(i)} = R_{r,k} + B_{r,k} P_{r,k|k-1}^{(i)} B_{r,k}^T$ is the predicted measurement covariance. For example, if the threshold is selected as $\gamma = 9$, the gate probability [46] will be $p_g \approx 0.99$. The node r only transmits its measurements that are in $\mathcal{R}_{r,k}$ to its neighbors. If the clutter intensity is not high, there is usually at most one measurement in the measurement validation region corresponding to each target. So each node usually transmits at most $n_y \times N_k$ scales to its neighbors, where the n_y is the dimension of measurements and N_k is the number of targets. The added communication load during the adaptation phase of the proposed algorithm will be limited.

V. NUMERICAL RESULTS

We investigate the performance of the proposed diffusion-based distributed algorithm by numerical results. Multitarget tracking is considered over a surveillance area of 1000m×1000m. $R = 16$ sensors are deployed in the surveillance area. The communication structure of the sensors is depicted in Figure 1.

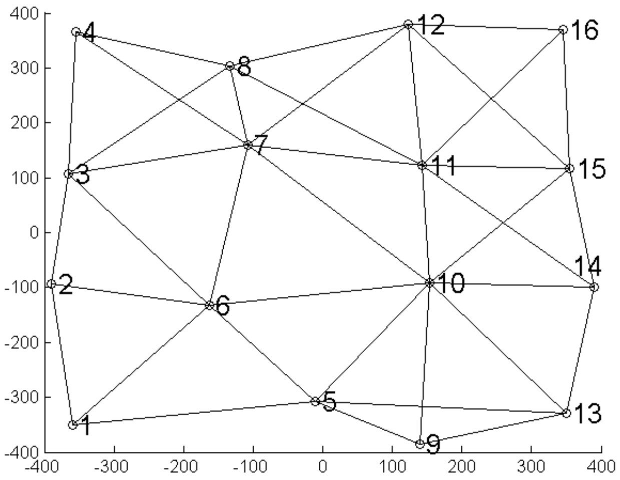


FIGURE 1. Sensor network and communication structure.

The target state is denoted by $x_k \triangleq (p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k})^T$, where $p_k \triangleq (p_{x,k}, p_{y,k})^T$ represents the target position, and $v_k \triangleq (\dot{p}_{x,k}, \dot{p}_{y,k})^T$ represents the target velocity in the (x, y) coordinate system at the time k. The target dynamics are described by the constant velocity model as [47]

$$x_k = A_k x_{k-1} + u_k, \tag{75}$$

where $A_k = I_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, the process noise u_k is with zero mean and covariance $Q_k = \sigma_u^2 I_2 \otimes \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}$, $T = 1$(s) is the sampling time period, I_2 is the 2×2 identity matrix, and $\otimes$ is the Kronecker operator.

The measurement of the rth node satisfies the linear model

$$z_{r,k} = B_{r,k} x_k + v_{r,k}, \tag{76}$$

where $B_{r,k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, and the measurement noise $v_{r,k}$ is with zero mean and covariance $R_{r,k} = \sigma_{v,r}^2 I_2$.

We investigate and compare the following algorithms:

1) Independent algorithm (IA). Each node independently tracks the multitarget state with its own measurements.
2) Consensus-based distributed algorithm (CDA) in [36]. Each node first tracks the multitarget state with its own measurements and its prior statistical knowledge, and then the estimates are fused by the consensus iterations over the network.
3) Local algorithm (LA). Each node tracks the multitarget state with all its neighbors' measurements.
4) The proposed diffusion-based distributed algorithm (DDA). Each node first tracks the multitarget state with all its neighbors' measurements and its prior statistical knowledge, and then the estimates are fused among the neighbors.

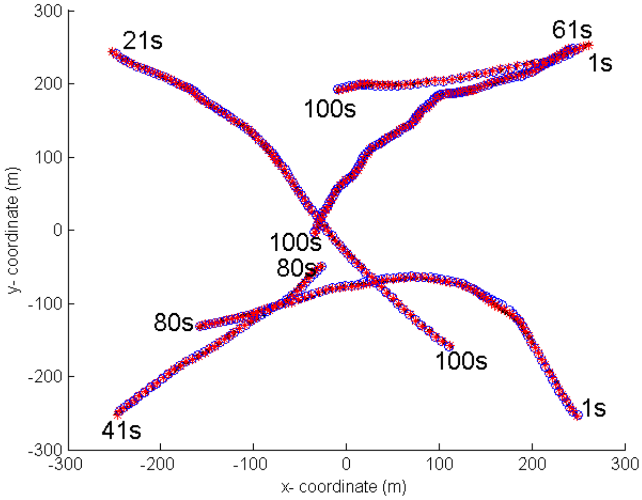


FIGURE 2. Simulated target trajectory and estimated target trajectories of one Monte Carlo run. The simulated trajectory is denoted as "·". The estimated trajectories by the CDA and DDA are denoted as "∘" and "⋆", respectively.

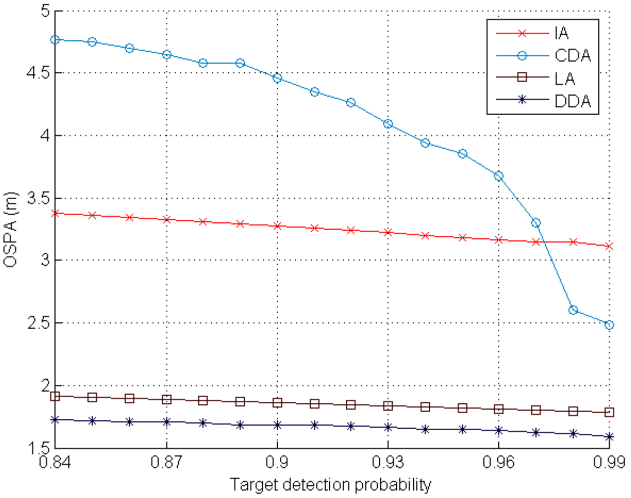


FIGURE 3. OSPA performances of different algorithms versus target detection probability.

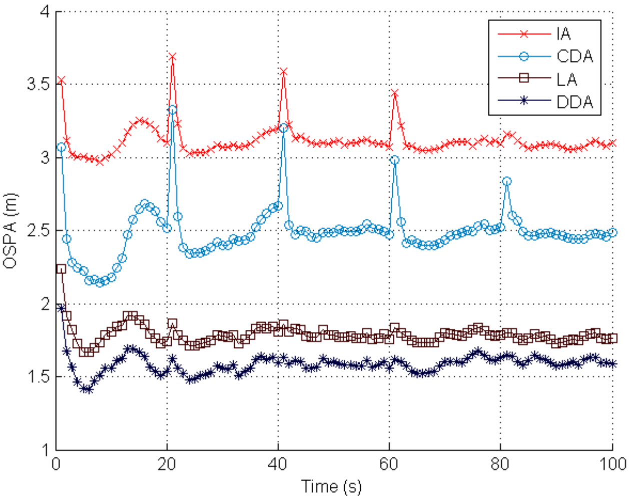


FIGURE 4. OSPA performances of different algorithms versus time step. The target detection probability is set to $p_{d,r} = 0.99$.

In the simulation scenario, the survival probability of target is set to $p_v = 0.98$. The process noise variance in (75) is $\sigma_u^2 = 0.5$. The parameters of the birth intensity $s_b(x)$ in (35) are $N_b = 4$, $\omega_b^{(1)} = \cdots = \omega_b^{(4)} = 0.005$, $m_b^{(1)} = (250, -4, 250, -4)^T$, $m_b^{(2)} = (250, -4, -250, 4)^T$, $m_b^{(3)} = (-250, 4, 250, -4)^T$, $m_b^{(4)} = (-250, 4, -250, 4)^T$, and $P_b^{(1)} = \cdots = P_b^{(4)} = \mathrm{diag}((25, 1, 25, 1)^T)$. Clutter is modeled as a Poisson process with the mean $\lambda_c = 20$ and uniform spatial distribution over the surveillance area. The number of consensus iterations for the CDA is set to $N_c = 6$.

We repeat independent Monte Carlo runs to evaluate the performance of these algorithms. The length of time interval in each Monte Carlo run is set to $K = 100$(s). We assume that no target exists at the initial time $k = 0$. Two targets appear at the time $k = 1$, three targets appear at the time $k = 21$, $k = 41$ and $k = 61$ respectively, and two targets disappear at the time $k = 81$. Figure 2 presents the simulated target trajectory in the (x, y) coordinate system of one Monte Carlo run. The estimated target trajectories by the CDA and DDA are also depicted in Figure 2, where the measurement noise variance in (76) is set to $\sigma_{v,r}^2 = 10$ and the target detection probability at each node is set to $p_{d,r} = 0.99$. From Figure 2, both the DDA and CDA can give accurate estimation results, including the number of targets and the state of each target.

Next, we investigate the performance of these algorithms for different values of target detection probability. The tracking performance is evaluated in terms of the optimal subpattern assignment (OSPA) metric [48]. The parameters for the OSPA metric are set to $p = 1$ and cutoff parameter $c = 5$. The measurement noise variance in (76) is set to $\sigma_{v,r}^2 = 10$. The target detection probability at each node is set to $p_{d,r} = 0.84, 0.85, \cdots, 0.99$. The time-average OSPA performances for different algorithms, which are averaged over $G = 100$ Monte Carlo runs, are shown in Figure 3. From Figure 3, the DDA outperforms the LA, and both outperform the CDA and IA. By incorporation of the neighbors' measurements, it can largely improve the performance of algorithms. It is also noted that the performance of CDA quickly deteriorates as the target detection probability decreases and it is even worse than the IA. For the CDA, when the detection probability is low, and if one node does not obtain its own measurements from the targets, the node can not provide correct local estimate about the targets since the node updates its local estimate only with its own measurements. Moreover, the incorrect local estimate will be diffused to the network by the consensus iterations, since the consensus of all local estimates are generally required over network for a number of consensus iterations. If several nodes provide incorrect local estimate, the estimation performance of the whole network will deteriorate.

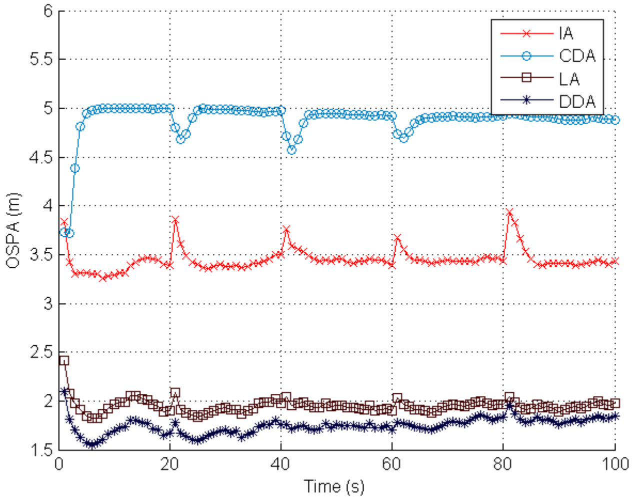


FIGURE 5. OSPA performances of different algorithms versus time step. The target detection probability is set to $p_{d,r} = 0.80$.

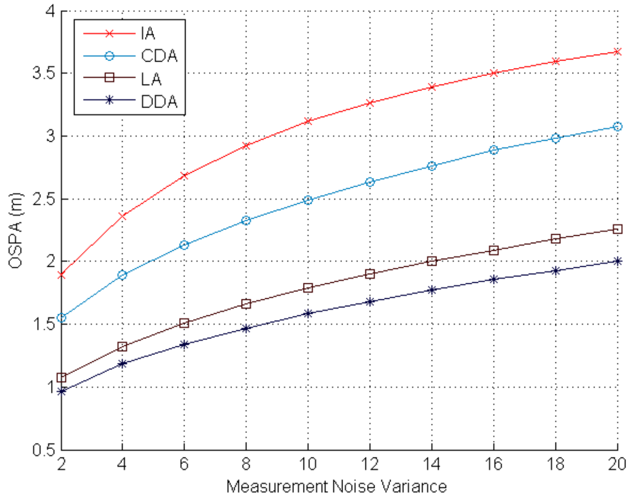


FIGURE 6. OSPA performances of different algorithms versus measurement noise variance. The target detection probability is set to $p_{d,r} = 0.99$.

Third, we investigate the OSPA performance of different algorithms for different time step k. The measurement noise variance $\sigma_{v,r}^2$ is selected to $\sigma_{v,r}^2 = 10$. Two values of target detection probability, i.e., $p_{d,r} = 0.80$ and $p_{d,r} = 0.99$, are considered. The OSPA performances of different algorithms with $p_{d,r} = 0.99$ are shown in Figure 4. The OSPA performances of different algorithms with $p_{d,r} = 0.80$ are shown in Figure 5. From Figure 4 and Figure 5, the DDA outperforms the LA, CDA and IA. From Figure 4, the CDA can provide the accurate tracking results with the target detection probability $p_{d,r} = 0.99$. But from Figure 5, the CDA can not provide the accurate tracking results if the target detection probability decreases to $p_{d,r} = 0.80$. The performance of CDA is even worse than that of the IA, which confirms again that the incorrect local estimates at some nodes, because of the low target detection probability, will be diffused to the network for a number of consensus iterations, and the estimation performance of the whole network will deteriorate.

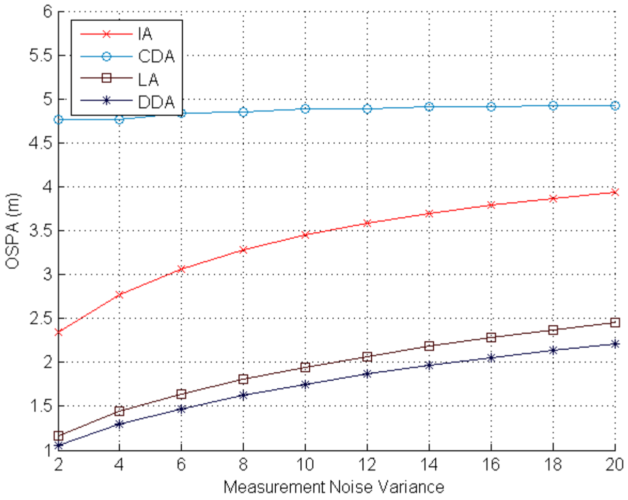


FIGURE 7. OSPA performances of different algorithms versus measurement noise variance. The target detection probability is set to $p_{d,r} = 0.80$.

Finally, we investigate the time-average OSPA performance of different algorithms for different values of measurement noise variance $\sigma_{v,r}^2$. The values of $\sigma_{v,r}^2$ are selected to $2, 4, \ldots, 20$. The time-average OSPA performances of different algorithms with $p_{d,r} = 0.99$ are shown in Figure 6. The time-average OSPA performances of different algorithms with $p_{d,r} = 0.80$ are shown in Figure 7. From Figure 6, the CDA can provide the accurate tracking results with the target detection probability $p_{d,r} = 0.99$. But from Figure 7, the performance of CDA will deteriorate if the target detection probability decreases to $p_{d,r} = 0.80$, and it is even worse than that of the IA.

From Figure 4-Figure 7, the DDA clearly outperforms the LA, which illustrates that the combination step in the DDA can clearly improve the estimation results of each node. Both the DDA and LA largely outperform the CDA and IA, which shows that the incorporation of the neighbors' measurements can largely improve the estimation performance of each node.

VI. CONCLUSION

We developed a diffusion-based distributed multitarget tracking algorithm over sensor network. The proposed distributed algorithm is composed of two phases: an adaptation phase that updates the estimate of each node by using all its neighbors' measurements, and a combination phase that fuses the neighbors' local estimates. A Gaussian mixture implementation of the proposed algorithm are also provided. Numerical results for different target detection probability and measurement noise variance confirmed the effectiveness of the proposed algorithm. Compared to the consensus-based distributed algorithm, the proposed algorithm can provide more accurate and robust estimation results, especially when the detection probability that the sensors detect the targets is low.

The complexity of the algorithms are also analyzed which shows that the proposed algorithm has lower communication load.

APPENDIX

For the partition P in (18), it denotes

$$|P|_j = \sum_{i=1}^{n} |\{z \in W^i_{1:d_r} : z \in Z_{j,k}\}| \tag{77}$$

as the number of measurements from the jth neighbor ($1 \leq j \leq d_r$) that are generated by the targets.

For the jth neighbor, it defines $C_j(t)$ as the probability generating function of the clutter cardinality distribution $p_{j,c}(n)$

$$C_j(t) = \sum_{n=0}^{\infty} t^n p_{j,c}(n), \tag{78}$$

and its vth order derivative as $C_j^{(v)}(t) = \frac{d^v C_j}{dt^v}(t)$.

The quantities γ, k_p and d_W has the form as [32]

$$\gamma = \int \mu_{r,k|k-1}(x) \prod_{j=1}^{d_r} q_{d,j}(x)dx, \tag{79}$$

$$k_p = \prod_{j=1}^{d_r} C_j^{(|Z_{j,k}|-|P|_j)}(0), \tag{80}$$

$$d_W = \frac{\int \mu_{r,k|k-1}(x) G_W(x) \prod_{j:(j,*)\notin T_W} q_{d,j}(x)dx}{\prod_{(j,l)\in T_W} s_{j,c}(z^l_{j,k})}, \tag{81}$$

where $G_W(x) = \prod_{(j,l)\in T_W} p_{d,j}(x)\xi_{j,k}(z^l_{j,k}|x)$, $z^l_{j,k} \in Z_{j,k}$ is the lth measurement of the jth neighbor at time k, and $s_{j,c}(\cdot)$ is the clutter density for the jth neighbor. To keep the expressions compact, we drop the index in $W_{1:d_r}$ and write it as W in the above equations.

The quantities α_0 and α_P have the form as [32]

$$\alpha_0 = \frac{\sum_{P\in\mathcal{P}} \left(k_P M^{(|P|)}(\gamma) \prod_{W\in P} d_W \right)}{\sum_{P\in\mathcal{P}} \left(k_P M^{(|P|-1)}(\gamma) \prod_{W\in P} d_W \right)}, \tag{82}$$

$$\alpha_P = \frac{k_P M^{(|P|-1)}(\gamma) \prod_{W\in P} d_W}{\sum_{Q\in\mathcal{P}} \left(k_Q M^{(|Q|-1)}(\gamma) \prod_{W\in W} d_W \right)}. \tag{83}$$

REFERENCES

[1] D. L. Hall, C.-Y. Chong, J. Llinas, and M. Liggins, II, *Distributed Data Fusion for Network-Centric Operations*. Boca Raton, FL, USA: CRC Press, 2012.

[2] M. Xie, W. Yi, T. Kirubarajan, and L. Kong, "Joint node selection and power allocation strategy for multitarget tracking in decentralized radar networks," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 729–743, Feb. 2018.

[3] A. Dehghan and M. Shah, "Binary quadratic programing for online tracking of hundreds of people in extremely crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 568–581, Mar. 2018.

[4] M. Wang, H. Ji, Y. Zhang, and X. Hu, "A Student's t mixture cardinality-balanced multi-target multi-bernoulli filter with heavy-tailed process and measurement noises," *IEEE Access*, vol. 6, pp. 51098–51109, 2018.

[5] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwell, MA, USA: Artech House, 2007.

[6] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[7] R. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, Oct. 2007.

[8] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT, USA: YBS Publishing, 2011.

[9] S. He, H.-S. Shin, and A. Tsourdos, "Multi-sensor multi-target tracking using domain knowledge and clustering," *IEEE Sensors J.*, vol. 18, no. 19, pp. 8074–8084, Oct. 2018.

[10] T. Zhang, H. Li, L. Yang, W. Liu, and R. Wu, "Multi-radar bias estimation without *a priori* association," *IEEE Access*, vol. 6, pp. 44616–44625, 2018.

[11] F. Zhao and L. J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Amsterdam, The Netherlands: Morgan Kaufmann, 2004.

[12] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 61–81, Jan. 2013.

[13] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Consensus-based distributed particle filtering with distributed proposal adaptation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3029–3041, Jun. 2014.

[14] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 1410–1415, May 2015.

[15] 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.

[16] A. Keshavarz-Mohammadiyan and H. Khaloozadeh, "Consensus-based distributed unscented target tracking in wireless sensor networks with state-dependent noise," *Signal Process.*, vol. 144, pp. 283–295, Mar. 2018.

[17] Y. Yu, "Distributed multimodel Bernoulli filters for maneuvering target tracking," *IEEE Sensors J.*, vol. 18, no. 14, pp. 5885–5896, Jul. 2018.

[18] Y. Yu and Y. Liang, "Consensus-based distributed detection and tracking in clutter with random existence of target," *Signal Process.*, vol. 158, pp. 66–78, May 2019.

[19] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[20] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[21] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[22] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[23] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[24] K. Dedecius, J. Reichl, and P. M. Djurić, "Sequential estimation of mixtures in diffusion networks," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 197–201, Feb. 2015.

[25] K. Dedecius and P. M. Djurić, "Sequential estimation and diffusion of information over networks: A Bayesian approach with exponential family of distributions," *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1795–1809, Apr. 2017.

[26] W. Li and Y. Jia, "Distributed estimation for Markov jump systems via diffusion strategies," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 448–460, Feb. 2017.

[27] F. Pang, K. Doğançay, and Q. Zhang, "Distributed detection of Gauss–Markov signals using diffusion Kalman filtering," *Signal Process.*, vol. 153, pp. 368–378, Dec. 2018.

[28] S. Xie and L. Guo, "Analysis of distributed adaptive filters based on diffusion strategies over sensor networks," *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 3643–3658, Nov. 2018.

[29] R. Mahler, "The multisensor PHD filter: I. General solution via multitarget calculus," *Proc. SPIE*, vol. 7336, May 2009, Art. no. 73360E.

[30] R. Mahler, "The multisensor PHD filter: II. Erroneous solution via 'Poisson magic,'" *Proc. SPIE*, vol. 7336, May 2009, Art. no. 73360D.

[31] E. Delande, E. Duflos, P. Vanheeghe, and D. Heurguier, "Multi-sensor PHD: Construction and implementation by space partitioning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, USA, May 2011, pp. 3632–3635.

[32] S. Nannuru, S. Blouin, M. Coates, and M. Rabbat, "Multisensor CPHD filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1834–1854, Aug. 2016.

[33] A.-A. Saucan, M. J. Coates, and M. Rabbat, "A multisensor multi-bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5495–5509, Oct. 2017.

[34] M. Üney, D. E. Clark, and S. J. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 521–531, Jun. 2013.

[35] M. Üney, J. Houssineau, E. Delande, S. J. Julier, and D. Clark, "Fusion of finite set distributions: Pointwise consistency and global cardinality," *IEEE Trans. Aerosp. Electron. Syst.*, to be published.

[36] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 508–520, Jun. 2013.

[37] M. R. Leonard and A. M. Zoubir, "Multi-target tracking in distributed sensor networks using particle PHD filters," *Signal Process.*, vol. 159, pp. 130–146, Jun. 2019.

[38] D. Clark, S. Julier, R. Mahler, and B. Ristić, "Robust multi-object fusion with unknown correlation," in *Proc. Sensor Signal Process. Defense Conf. (SSPD)*, London, U.K., 2010, pp. 1–5.

[39] S. J. Julier, "Fusion without independence," in *Proc. IET Seminar Target Tracking Data Fusion, Algorithms Appl.*, 2008, pp. 3–4.

[40] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[41] B. T. Vo, B. N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, Jul. 2007.

[42] B. Upcroft, L. L. Ong, S. Kumar, M. Ridley, T. Bailey, H. Durrant-Whyte, and S. Sukkarieh, "Rich probabilistic representations for bearing only decentralised data fusion," in *Proc. 7th Int. Conf. Inf. Fusion*, 2005, pp. 1–8.

[43] S. J. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *Proc. 9th Int. Conf. Inf. Fusion*, 2006, pp. 1–8.

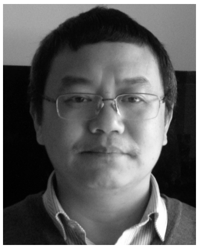
[44] M. Gunay, U. Orguner, and M. Demirekler, "Chernoff fusion of Gaussian mixtures based on sigma-point approximation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2732–2746, Dec. 2016.

[45] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proc. 5th Int. Conf. Inf. Fusion*, 2002, pp. 901–904.

[46] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[47] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[48] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.



YIHUA YU (M'13) received the B.S., M.S., and Ph.D. degrees from the School of Mathematical Sciences, Peking University, Beijing, China, in 1997, 2000, and 2003, respectively.

He is currently an Associate Professor with the School of Science, Beijing University of Posts and Telecommunications. His research interests include statistical signal processing for digital communication, linear and nonlinear Bayesian estimation, and tracking in wireless networks.

YUAN LIANG received the B.S. and Ph.D. degrees from Peking University, Beijing, China, in 1998 and 2003, respectively.

She is currently an Associate Professor with the School of Humanities and Social Science, Beijing Institute of Technology. Her research interests include Bayesian inference, and linear/nonlinear filtering and its applications.