

Research

Open Access



Computation-distributed probability hypothesis density filter

Junjie Wang^{1†}, Lingling Zhao^{1*†}, Xiaohong Su¹, Chunmei Shi² and JiQuan Ma³

Abstract

Particle probability hypothesis density filtering has become a promising approach for multi-target tracking due to its capability of handling an unknown and time-varying number of targets in a nonlinear, non-Gaussian system. However, its computational complexity linearly increases with the number of obtained observations and the number of particles, which can be very time consuming, particularly when numerous targets and clutter exist in the surveillance region. To address this issue, we present a distributed computation particle probability hypothesis density (PHD) filter for target tracking. It runs several local decomposed particle PHD filters in parallel while processing elements. Each processing element takes responsibility for a portion of particles but all measurements and provides local estimates. A central unit controls particle exchange among the processing elements and specifies a fusion rule to match and fuse the estimates from different local filters. The proposed framework is suitable for parallel implementation. Simulations verify that the proposed method can significantly accelerate and maintain a comparative accuracy compared to the standard particle PHD filter.

Keywords: Multi-target tracking, Probability hypothesis density filter, Particle filter, Distributed computation

1 Introduction

Multi-target filtering is a class of dynamic state estimation problems in which the object of interest is a finite set that consists of a random number of elements and the values of individual elements [1]. Classical approaches, such as the nearest neighbor (NN) [2], the joint probabilistic data association filter (JPDA) [3], and multi-hypothesis tracking (MHT) [4], are based on the framework of filter and data association. Recently, considerable work has been devoted to random finite set (RFS)-based approximations, such as the probability hypothesis density (PHD) [5, 6], the cardinalized PHD (CPHD) [7, 8], and the multiple-target multi-Bernoulli (MeMBer) filter [9]. These methods avoid the data association problem and provide the set-valued estimations of target states.

Among these filters based on RFS, the MeMBer filter is more suitable for low-clutter environments. The CPHD filter propagates both the intensity of the RFS and

the posterior cardinality distribution; consequently, it provides a more accurate target number estimation but a lower efficiency. For the PHD filter, a particle PHD filter for nonlinear and/or non-Gaussian MTT problems which are more suitable for general scenarios was proposed in [1]. Moreover, another implementation is a closed-form solution [6] with assumptions on linear Gaussian systems, which is called the GM-PHD filter. The GM-PHD filter provides the best efficiency in general but is constrained to linear Gaussian systems, whereas the particle PHD filter can adapt nonlinear or non-Gaussian MTT problems even in dense-clutter environments. However, the particle PHD filter suffers from high computational complexity because hundreds of thousands of particles are required. To solve the high computational complexity problem, Hong et al. [10] proposed a new update model for the particle PHD filter that is suitable for hardware implementation. In the paper [11], a data-driven particle PHD filter was developed for real-time multi-target tracking of nonlinear/non-Gaussian systems in dense-clutter environments. Using gating technology, Li et al. [12] accelerated the PHD filter to some extent. A pipeline approach was also introduced into the particle PHD filter in [13]. In contrast to the aforementioned methods, we attempt to

*Correspondence: zhaoll@hit.edu.cn

†Equal contributor

¹School of Computer Science and Technology, Harbin Institute of Technology, XiZhida Road, 150001 Harbin, China

Full list of author information is available at the end of the article

improve the efficiency of the particle PHD filter through the use of a distributed architecture, which is an effective solution for accelerating its computation to satisfy the time demands in MTT and some distributed hardware such as wireless sensor networks (WSNs) [14, 15]. The existing distributed PHD/CPHD filter is generally based on such an architecture in which each sensor runs a local PHD/CPHD filter with some local measurements, called distributed sensing PHD (DSPHD) filter [16, 17].

In this paper, we propose a distribution implementation for the particle PHD filter using multiple processing elements (PEs) and a central unit (CU). The CU transfers all measurements to all the PEs first, and then each PE runs a local particle PHD filter and provides target state estimation with labels independently and in parallel. To match the local estimations from different PEs, we replace the general particle PHD filter with the decomposed particle PHD filter [18] at each PE, which can help associate estimated states according to their corresponding measurements. Finally, the global estimates are obtained on the CU through the association and fusion of the local estimated states from different PEs. Two fusion rules are presented for different tracking scenarios with clutter. Moreover, a particle exchange strategy is exploited to eliminate particle degradation.

Compared to the DSPHD filter exploiting local measurements and information exchange to estimate the global states, our algorithm provides a different type of allocation of measurements. In our algorithm, each PE runs with all sensor measurements but only a subset of particles, whereas in the general distributed PHD/CPHD filter, each PE runs with some sensor measurements rather than all measurements. In the DSPHD filters, the local means and covariances of the state estimates are exchanged among PEs and the CU. Since the local filters use the different particles and measurements, they do not speak the same language. Therefore, they need to agree on consensus through some mechanisms. In contrast, our distributed computational PHD filter shares the same measurements but only uses a subset of particles in each PE; therefore, their languages are compatible with each other, and thus, PEs can exchange information of the particle-weight couples.

The main contributions of this paper consist of the following three parts:

First, we propose a distributed architecture for the PHD filter, which makes the update and resampling step in the particle PHD filter in parallel in part and significantly accelerates the particle PHD filter. This architecture is valid not only for the particle PHD filter but also for the GM-PHD filter.

Second, we exploit the decomposed PHD filter to extract the estimated states and obtain their corresponding measurement labels, which guarantees the association

and fusion of states from different PEs. Furthermore, we present two rules for the fusion of local state estimations, considering the influence of clutter on the PHD filter.

Finally, based on the architecture and algorithm of the proposed distributed particle PHD filter, the real-time performance is enhanced while the tracking accuracy is comparable to a traditional particle PHD filter.

The remainder of this paper is organized as follows. The standard particle PHD filter is briefly described in Section 2. In Section 3, we present and analyze our distributed particle PHD filter in detail. Simulation results are presented in Section 4. Section 5 provides the conclusions.

2 Background

2.1 The PHD filter

The PHD filter was initially developed in the framework of finite set statistics (FISST) [5]. The PHD function D_{Ξ} is the first-order moment of the random finite set (RFS) Ξ and can be defined as

$$D_{\Xi}(x) \equiv E[\delta_{\Xi}(x)] = \int \delta_X(x) P_{\Xi}(dX) \quad (1)$$

where $\delta_{\Xi}(x) = \sum_{x \in \Xi} \delta_x$ is the random density representation of Ξ . P_{Ξ} is the probability distribution of the RFS Ξ . $E[\cdot]$ is the expectation operator. The PHD D_{Ξ} of Ξ is a unique function on space E , unless it is on a set of measure zero. The PHD has the properties that [19], the integral over a measurable subset $S \subseteq E$, $\int_S D_{\Xi}(x) \lambda(dx)$ is the expected number of targets. In addition, the peaks of the PHD function provide the estimates of the target states.

The PHD filter consists of two steps: the *prediction* step and the *update* step. It is used to recursively estimate the combined posterior density using multi-target transition density $f_{k|k-1}(\cdot)$ and measurement likelihood $g_k(\cdot)$. Assuming that the RFS is Poisson, it has been shown that the recursion propagating the PHD $D_{k|k}$ of the multi-target posterior $p_{k|k}$ follows [5]

$$D_{k|k} = (\Psi_k \circ \Phi_{k|k-1}) D_{k-1|k-1} \quad (2)$$

where \circ represents the composition of functions, $\Phi_{k|k-1}$ is the prediction operator, and Ψ_k is the update operator. They are defined as follows:

$$(\Phi_{k|k-1}\alpha)(x) = \gamma_k + \int f_{k|k-1}(x, \xi) \alpha(\xi) \lambda(d\xi) \quad (3)$$

$$(\Psi_k\alpha)(x) = \left[1 - P_D(x) + \sum_{z \in Z_k} \frac{\psi_{k,z}(x)}{\kappa_k(z) + \langle \psi_{k,z}, \alpha \rangle} \right] \alpha(x) \quad (4)$$

where $P_D(\cdot)$ is the probability of detection, $\kappa_k(\cdot)$ denotes the intensity function of clutter at time k , $\gamma_k(\cdot)$ denotes

the intensity function of the random finite set of spontaneous birth targets, $e_{k|k-1}(\xi)$ denotes the survival probability with state ξ ; $c_k(z)$ as the clutter probability density, and $b_{k|k-1}(\cdot|\xi)$ denotes the PHD of the RFS $B_{k|k-1}(\{\xi\})$ spawned by a target with previous state ξ .

$$\begin{aligned}\phi_{k|k-1}(x, \xi) &= e_{k|k-1}(\xi)f_{k|k-1}(x|\xi) + b_{k|k-1}(x|\xi) \\ \psi_{k,z}(x) &= P_D(x)g_k(z|x) \\ \kappa(z) &= \lambda_k c_k(z) \\ \langle f, g \rangle &= \int f(x)g(x)\lambda(dx)\end{aligned}$$

2.2 Particle PHD filter

As an approximate implementation of the PHD filter, the particle PHD filter is composed of three steps (for detailed parameter notation, refer to [1]):

At time $k > 0$, let L_k and Jk denote the number of surviving particles and birthed particles, respectively. q_k denotes the importance function.

1) Prediction step.

For $i = 1, \dots, L_{k-1}$, sample $\tilde{x}_k^i \sim q_k(\cdot|x_{k-1}^i, Z_k)$ and compute the predicted weights

$$\tilde{w}_{k|k-1}^i = \frac{\phi_{k|k-1}(x_k^i, x_{k-1}^i)}{q_k(x_k^i|x_{k-1}^i, Z_k)} w_{k-1}^i \quad (5)$$

For $i = L_{k-1} + 1, \dots, L_{k-1} + Jk$, sample $\tilde{x}_k^i \sim p_k(\cdot|Z_k)$ and compute the weights of newborn particles

$$\tilde{w}_{k|k-1}^i = \frac{\gamma_k(x_k^i)}{p_k(x_k^i|Z_k)} \frac{1}{Jk} \quad (6)$$

2) Update step.

For each $z \in Z_k$, compute

$$C_k(z) = \sum_{j=1}^{L_{k-1}+Jk} \psi_{k,z}(\tilde{x}_k^j) \tilde{w}_{k|k-1}^j * \tilde{x}_k^j \quad (7)$$

For $i = 1, \dots, L_{k-1} + Jk$, update weights

$$\tilde{w}_k^i = \left[v(\tilde{x}_k^i) + \sum_{z \in Z_k} \frac{\psi_{k,z}(\tilde{x}_k^i)}{\kappa_k(z) + C_k(z)} \right] \tilde{w}_{k|k-1}^i \quad (8)$$

3) Resampling step.

Compute the total target number $N_k = \sum_{j=1}^{L_{k-1}+Jk} \tilde{w}_k^j$

and resample $\{\tilde{x}_k^i, \tilde{w}_k^i/N_{k,i=1}^{L_{k-1}+Jk}\}$ to obtain

$$\{x_k^i, w_k^i/N_k\}_{i=1}^{L_k}$$

As with the particle filter, the application of the particle PHD filter is limited by its computational complexity, which is primarily caused by resampling. At the same time, the computational complexity is also caused by the update, in which all particles participate.

3 Distributed computation particle PHD filter

The particle PHD filter has the capacity of handling nonlinear non-Gaussian dynamics inherent from the particle filter. However, it also suffers from a high computational cost, which limits its application in real-time systems. To improve its efficiency, we propose a distributed computation particle PHD filter (DCPPHD). This method is motivated by the distributed resampling with non-proportional allocation (DRNA) proposed by Bolic et al. [20]. The underlying concept of the DRNA is to attain parallelism by resampling independently for different subsets of particles. In a multicore computer with K processing elements (PEs), $n = 1, \dots, K$, a particle filter is performed locally by each PE. To keep balance of aggregated weights, PEs exchange parts of particles with each other. Although DRNA has provided an effective distribution framework, there are still challenges in the distribution implementation of the particle PHD filter. First, more than one target generally exists in the surveillance region; thus, each PE provides a local state estimation set rather than a single estimation. How to distinguish these state estimations among PEs, that is, which estimations originate from the same target, is a key issue to address. Furthermore, after associating state estimations, it is necessary to design a fusion method in order to obtain the global estimations. To address these problems, in the proposed DCPPHD, we exploit the decomposed PHD filter [18] to obtain labeled state estimations corresponding to the measurements, which gives rise to the direct association of the estimated states. The global estimated states can be obtained from the labeled local state estimations. Moreover, some fusion strategies are presented according to different tracking scenarios.

3.1 General structure

In the proposed DCPPHD, particles are divided into groups to make the update and resampling steps parallel between groups. Each PE only contains a group of particles and runs a particle PHD filter independently until they exchange parts of particles among each other. A central unit (CU) takes responsibility for transferring measurements to each PE, fusing the local estimations from PEs and providing the global state estimations. Assume that K PEs exist and that each PE runs a local particle PHD filter with M particles. Therefore, the total number of particles of the DCPPHD is $N = MK$. The CU transfers all measurements to K PEs. For all the PEs, after the prediction, update and resampling steps, each local particle PHD filter provides local estimations (target number and states) to the CU in parallel. In particular, resampling is conducted locally without interacting with the particles in other PEs. After receiving these local estimations from the different PEs, the CU will associate and fuse them according to the proposed rules. To avoid the degeneracy

of the particles in local PEs, a particle exchange step is utilized, in which each PE mutually interchanges part of its particles to another PE.

The structure of the DCPPHD with four PEs is shown in Fig. 1. At each time step, the CU broadcasts all the measurements to all PEs; then, each PE runs its local decomposed particle PHD filter and transmits its local estimated states and corresponding index set $\{\zeta_k^l, I_k^l\}$ to the CU, where I_k^l denotes the index of the measurement that is related to ζ_k^l . Subsequently the CU can associate the local estimated states based on the measurement index to construct global estimates. Following the local estimation step, each PE exchanges L particles with its neighboring PEs.

Note that the particle exchange step begins once the local estimations have been submitted to the CU.

3.2 Algorithm of DCPPHD filter

The algorithm of the DCPPHD filter primarily involves four steps and runs on PEs and the CU: local filter, local estimation, particle exchange on PEs, and global estimation on the CU. The details of each step are summarized in the following.

3.2.1 Local filter and local labeled state estimation

To identify the set-valued state estimations from different PEs, we use the decomposed PHD filter to find the relationship between estimated states and measurements rather than the standard particle PHD filter. For the decomposed PHD filter, the estimated PHD is decomposed into several sub-PHDs in the weight domain. This

concept comes from the fact that the multi-target PHD $D_{t|t}$ can be rewritten as follows:

$$D_{t|t}(x|Z_{1:t}) = \sum_{z \in Z_t} \Delta D_{t|t}(x|z) + \Delta D_{t|t}(x|\phi) \tag{9}$$

where

$$\Delta D_{t|t}(x|z) = \frac{P_D L_t(z|x) D_{t|t-1}(x)}{\kappa_t(z) + \int P_D L_t(z|\zeta) D_{t|t-1}(\zeta) d\zeta} \tag{10}$$

and

$$\Delta D_{t|t}(x|\phi) = (1 - P_D) D_{t|t-1} \tag{11}$$

where $\Delta D_{t|t}(x|\phi)$ denotes the intensity function from the target with no measurement received. From Eq. (9), PHD can be considered as the sum of $M_t + 1$ sub-PHDs corresponding to every single target and every sub-PHD is relevant to a measurement received at time t . Consequently, the target states can be directly estimated from the sub-PHDs and each target state is labeled by its corresponding measurement. The local filter and local labeled state estimation algorithm is as follows:

1. Step 1: Prediction. At time $t - 1$, assume that the particle set $\{x_{t-1}^{(n,m)}, w_{t-1}^{(n,m)}\}$, $n = 1, \dots, K, m = 1, \dots, L_{t-1}^n$ is available where L_{t-1}^n denotes the number of particles in the n th PE. For the n th group and $m = 1, \dots, L_{t-1}^n$, sample $\{x_t^{(n,m)}\}$ from q_t for surviving targets and sample $\{x_t^{(n,m)}\}$, $m = L_{t-1}^n + 1, \dots, L_{t-1}^n + Jk/K$, from $p(x)$ for newborn targets.

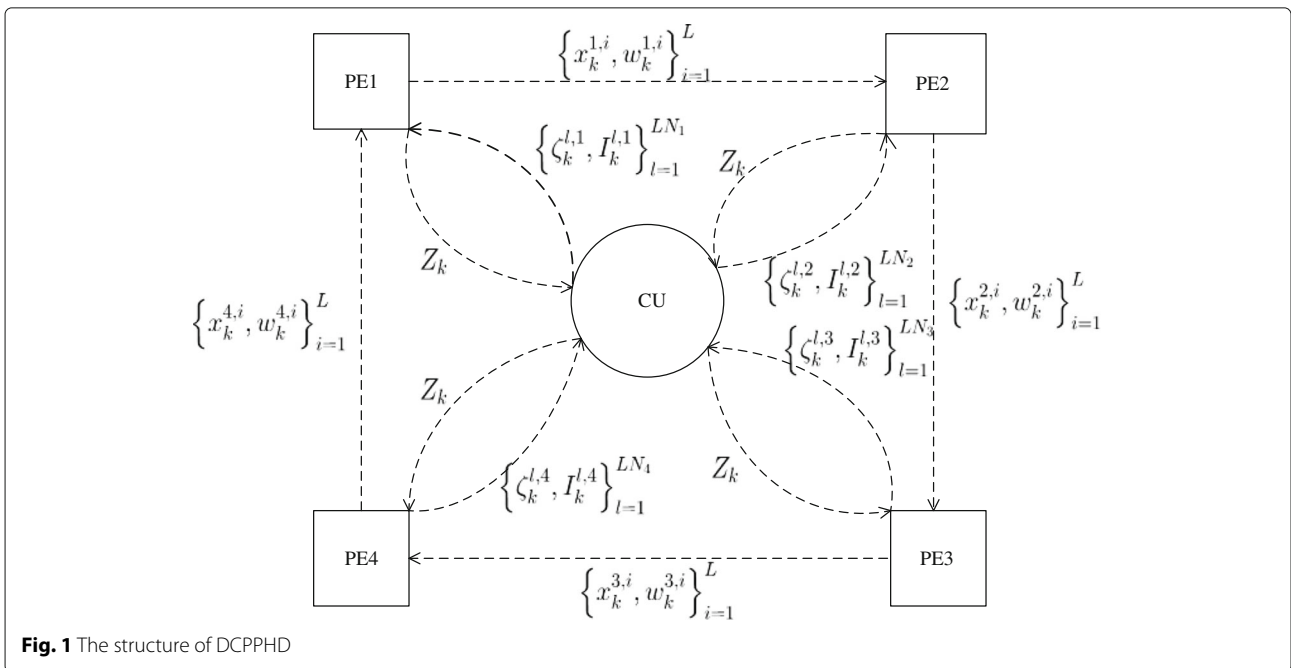


Fig. 1 The structure of DCPPHD

2. Step 2: Update. Let Z_t denote the measurement set. For each group $n, n = 1, \dots, K$ and each observation $z_p \in Z_t, p = 1, \dots, M_t$,

$$C_k(z_{t,p}) = \sum_{i=1}^{L_{t-1}^n + J_t} \psi_{t,z_{t,p}}(\tilde{x}_t^{i,n}) \tilde{w}_{t|t-1}^{(i,n)} \tilde{x}_t^{(i,n)} \quad (12)$$

$$G_t^{i,p,n} = \frac{\psi_{t,z}(\tilde{x}_t^{i,n})}{\kappa(z_{t,p}) + C_t(z_{t,p})} \quad (13)$$

then calculate the sub-weight of each particle for observations $z_{t,p}$

$$\Delta \tilde{w}_t^{i,p,n} = G_t^{i,p,n} \tilde{w}_{t|k-1}^{i,n} \quad (14)$$

Additionally, the particle sub-weight for obtained target without measurements is

$$\Delta \tilde{w}_t^{i,p,0} = v(\tilde{x}_t^{i,n}) \tilde{w}_{t|t-1}^{i,n} \quad (15)$$

Based on formula (9), particle weights can be computed by

$$\tilde{w}_t^{i,n} = \sum_{p=1}^{M_t} \Delta \tilde{w}_t^{i,p,n} + \Delta \tilde{w}_t^{i,p,0} \quad (16)$$

3. Step 3: Local estimation. For each measurement $z_{t,p}, p = 1, \dots, M_t$, compute the sum of $\Delta \tilde{w}_t^{i,p,n}$ relevant to $z_{t,p}$ in group n ,

$$\Delta W_t^{n,p} = \sum_{i=1}^{M+J_t} \Delta \tilde{w}_t^{i,p,n}. \quad (17)$$

Then, the sum of sub-weight $\Delta W_t^{n,0}$ corresponding to targets without observations is

$$\Delta W_t^{n,0} = \sum_{i=1}^{M+J_t} \Delta \tilde{w}_t^{i,0,n}. \quad (18)$$

The target number can be estimated by

$$LT_t^n = \text{round} \left(\sum_{i=1}^{i=L_{t-1}^n + J_k/K} \tilde{w}_t^{i,n} \right) \quad (19)$$

where $\text{round}(\sum_{i=1}^{i=L_{t-1}^n + J_k/K} \tilde{w}_t^{i,n})$ is the nearest integer to $\sum_{i=1}^{i=L_{t-1}^n + J_k/K} \tilde{w}_t^{i,n}$. Find the LT_t^n largest sum weight $\Delta W_t^{n,p}$ and the index set I_t^n relevant to $\Delta W_t^{n,p}$. The local estimated target state can be calculated by $\zeta_{t,l} = w_t^{i,l,n} \tilde{x}_t^{i,n}$ where l in index set,

$$w_t^{i,l,n} = \frac{\Delta \tilde{w}_t^{i,l,n}}{\sum_{i=1}^{M+J_k} \Delta \tilde{w}_t^{i,p,n}} \quad (20)$$

When the n th PE obtains a local estimation set $\zeta_t^l, l = 1, \dots, LT_t^n$, the PEs transmit the pair $\{\zeta_t^{l,n,p}, I_t^{l,p,n}\}$ to the CU, where $I_t^{l,p,n}$ denotes the state

estimation $\zeta_t^{l,n,p}$ from the $n - th$ PE coming from the measurement $z_{t,p}$.

4. Step 4: Resampling.

Each PE calculates its own estimated number of targets LT_t^i from its total particle mass according to Eq. (19). Consequently, the number of particles of the i th PE is updated by

$$\tilde{N}_i = LT_t^i \rho \quad (21)$$

Then, the resampling can be performed locally at all the PEs. Subsequently, normalize the weights of the particles with the sum of weights at each PE.

3.2.2 Global estimation

After receiving the local state estimations $\{\zeta_t^{n,l,p}, I_t^{l,p,n}\}$ from all PEs, the CU will combine the local estimated states and their corresponding measurement index from each group to construct a global estimate. Classify the estimated states with a same measurement index into a group $\zeta_{t,i} = \{\zeta_t^{1,l_1,p}, \zeta_t^{2,l_2,p}, \dots, \zeta_t^{K,l_K,p}\}$, where if the n th PE does not provide an estimated state $\zeta_t^{n,l_n,p}$, let $\zeta_t^{n,l_n,p} = \phi$.

For each estimated state group $\zeta_{t,i}$, calculate a unified state estimation according to a fusion rule. Here, we propose two fusion rules for global estimation under the basic assumption “at most one measurement per target” [21].

Rule 1: Only if $|\zeta_{t,i}| = K$, a global state can be obtained by $\hat{x}_t = \sum_{i=1}^K (\zeta_{t,i})/K$. This rule is designed for tracking scenarios with a higher clutter rate, in which false state estimates from false alarms tend to occur. However, false target state estimates are not always obtained by all PEs; then, rule 1 can help to eliminate some of them because only when all PEs consider an observation from a target will it be considered from a target.

Rule 2: If $|\zeta_{t,i}| \geq K/2$ (half number of PEs), then a global state can be obtained by $\hat{x}_t = \text{mean}(\zeta_{t,i})$. This rule is designed for tracking scenarios with lower clutter rates. Rule 2 relaxes the condition that a target exists because once the majority of PEs consider an observation from a target, the global estimation also considers it as if from a target.

3.2.3 Particle exchange

The particles in the n th PE will tend to degenerate when its aggregate weight becomes negligible relative to the aggregated weights of the other PEs. Then, the n th PE hardly contributes to the approximation of the posterior probability distribution. To keep the PEs valid, neighboring PEs should exchange a portion of particles.

The n th PE transmits P particles to its neighbor. The selected particles from the n th PE to the s th PE can be denoted as $\mathcal{M}_t^{n,s} = \{x_t^{(n,i_1)}, \dots, x_t^{(n,i_p)}\}$. Then, the infor-

mation of particle-weight pairs held by the i th PE after the exchange step is given by

$$X_t^i = X_t^i \cup \mathcal{M}_t^{(s,i)} = \left\{ \hat{x}_t^{(i,j)} \right\}_{j=1}^{N_i+P}$$

$$W_t^i = \left\{ \hat{w}_t^{(i,j)} \right\}_{j=1}^{N_i+P}$$

One of the particle exchange methods is to select the P highest weight particles at each PE to exchange. Let $\hat{p}_i(X_k|Z_k)$ be the local approximation of the posterior pdf, which can be defined as

$$\hat{p}_i = \sum_{j=1}^{N_i} w_k^{i,j} \delta x_k^{i,j} dx \quad (22)$$

where the subscript i indicates that weights and particles are referred to the i th local PHD filter. Let $\hat{w}_k^{i,j}$ denote the weight of the j th particle of the i th PE after the exchange step but prior to resampling. Let $\hat{N}_i = N_i + P$ be the total number of particles of the i -th PE after the exchange step prior to resampling.

It is known that the most representative particles of the $\hat{p}_i(X_k|Z_k)$ are those with the highest weights. We utilize the Kullback-Leibler (KL) divergence to measure the distance between two pdfs. The smaller the KL divergence is, the closer the two pdfs are. Consider the approximated posterior density $\hat{p}'_i(X_k|Z_k)$ computed using only P particles

$$\hat{p}'_i(X_k|Z_k) = \sum_{j=1}^P w_k^{i,j} \delta x_k^{i,j} dx \quad (23)$$

The P particles are those with the highest weights. The KL divergence between $\hat{p}_i(X_k|Z_k)$ and $\hat{p}'_i(X_k|Z_k)$ can be written as:

$$D(\hat{p}_i, \hat{p}'_i) = - \sum_{j=1}^{N_i} w_k^{i,j} \log \sum_{i_j=1}^P w_k^{i,j} \quad (24)$$

The maximization of $\sum_{j=1}^P w_k^{i,j}$ is equivalent to the minimization of $D(\hat{p}_i, \hat{p}'_i)$.

Therefore, the exchange information can be constructed by selecting the particles with the highest weights, to be communicated with other PEs. Thus, we use the P particles with the highest weights.

The global posterior $\hat{p}(X_t|Z_t)$ can be represented as a local posterior $\hat{p}_i(X_t|Z_t)$. We define the cumulative sum of the local KL divergences as:

$$\begin{aligned} \sum_{i=1}^K D(\hat{p}, \hat{p}_i) &= - \sum_{i=1}^K \Omega_t \log \left(\sum_{j=1}^{N_i+P} w_t^{i,j} \right) \\ &= - \sum_{i=1}^K \log \left(\sum_{j=1}^{N_i+P} \frac{w_t^{i,j}}{\Omega} \right) \\ &= - \sum_{j=1}^{N_i+P} \log \left(\frac{\Omega_i}{\Omega} + \sum_{j=N_i}^{N_i+P} \frac{w_t^{i,j}}{\Omega} \right) \end{aligned} \quad (25)$$

where $\Omega = \sum_{i=1}^K \Omega_i$. From Eq. (25), we can infer that the higher the number of exchanged particles is, the closer the local posterior and global posterior are. With $P \rightarrow \infty$, the local posterior is the closest to the $\hat{p}_i(X_t|Z_t)$. However, increasing the number of exchanged particles can lead to a high communication burden. In particular, if we choose the all-to-all network configuration in which the PE can receive other PEs' local approximation posterior, then the all-to-all network can obtain the best estimation compared with other types of networks. The network and the selected particles should compromise between the communication burden and estimation quality. Therefore, we simply select those particles with the highest weights to exchange to balance these two factors.

DCPPHD

At time $t = 0$

Initialization

for each PE, $n = 1, \dots, K$:

a) Draw $x_0^{(n,i)}$, for $i = 1, \dots, L_t^n$, from the prior pdf $p(x_0)$ and assign equal weights, $w_0^{(n,i)} = \frac{1}{L_t^n}$.

b) Set $\chi_0^{(n)} = \left\{ x_0^{(n,i)} \right\}_{i=1}^{L_t^n}$, $\mathcal{W}_0^n = \left\{ w_0^{(n,i)} \right\}_{i=1}^{L_t^n}$.

At time $t > 0$, for $n = 1, \dots, K$:

Local filter

for $i = 1, \dots, L_{t-1}^{(n)}$ do

$\tilde{x}_t^{(i,n)} \sim q(\cdot | x_{t-1}^{(i,n)}, z_t)$.

end for

for $i = L_{t-1}^{(n)} + 1, \dots, L_{t-1}^{(n)} + Jk/K$ do

$\tilde{x}_t^{(i,n)} \sim p(\cdot | z_t)$.

end for

for $i = 1, \dots, L_{t-1}^{(n)} + Jk/K$ do

weight update

$w_t^{(i,n)} = [v(\tilde{x}_t^{(i,n)}) + \sum_{z \in Z_t} \frac{\psi_{t,z}(\tilde{x}_t^{(i,n)})}{\kappa_k(z) + C_t(z)}] \tilde{w}_{t|t-1}^i$

end for

estimate $N_t(n)$ and $\zeta(n)$ as in 3

resample the $L_{t-1}^{(n)} + Jk/K$ particles into $N_t(n)\rho$ particles

and set $L_t^{(n)} = N_t(n)\rho$

Global estimation

According to the 3.2.2, get the global estimation.

Particles exchange

neighbor PEs exchange particles, as described in 3.2.3,

to obtain the sets $\left\{ \chi_t^{(n)}, \mathcal{W}_t^n \right\}$.

3.3 Analysis of DCPPHD

The efficiency of improvement using parallel computing can be roughly estimated using Amdahl's law. According to Amdahl's law [22], the acceleration of DCPPHD is defined by

$$S(K) = \frac{1}{p + (1-p)/K}$$

where $S(K)$ is the theoretical acceleration achieved for K PEs, where p is the proportion of the algorithm that sequentially executed. For a completely parallelizable algorithm ($p = 0$), the DCPPHD has a theoretically maximum achievable acceleration limited to K . Noted that Amdahl's law only represents a theoretically predicted acceleration. In practical cases, the extra time introduced by the parallel computing should also be taken into consideration.

In the case of the DCPPHD filter, the parallel operations take a higher percentage; thus, the DCPPHD algorithm has the potential to achieve good acceleration.

3.4 Remark

The distributed computational PHD filter can be introduced to the jump Markov systems (JMS) with the transition probability being a constant instead of a variable following some distribution. For the jump Markov systems, several versions of the PHD filter have been proposed: (1) Augmented PHD filter and its extension for the linear jump Markov systems and the nonlinear jump Markov models proposed by Pasha et al [23]; (2) the generalized pseudo Bayesian (GPB) PHD and best-fitting Gaussian (BFG) PHD filter proposed in [24] and [25], respectively. These methods can be extended to the distributed version based on the mechanism proposed in our paper. For example, in the augmented PHD filter, multiple PHD filters work independently under different models, consequently, each PHD filter can be converted into our DCPPHD filter running in parallel on each PE, which does not influence the key idea of the augmented PHD filter in essence. On the other hand, for the JMSPHD filter propose in [24] and [25], the same approximation from the jump Markov system to a single model by GPB or BFG can be also used on CU in our DCPPHD before filtering. Therefore, our DCPPHD filter can be applied for the JMS under the assumption that the transition probability is a constant. However, for the semi-Markovian jump systems mentioned in the papers [26] and [27], the transition probabilities with sojourn-time subject to Weibull distribution or Laplace distribution, Gaussian distribution, etc. is time-varying, which does not satisfy the assumption for the current JMS PHD filter; thus, our DCPPHD cannot be applied for the semi-Markovian jump systems directly since transition probabilities are unknown and need to be estimated at each time. The details about the augmented PHD filter, GPB PHD filter, and BFG PHD filter refer to [23–25], respectively.

As one kind of the typical distributed systems, multi-agent systems have been widely used in many applications [28–31]. The PHD filter including both of the proposed DCPPHD and DSPHD can be applied for the multi-agent systems with switching network topologies and periodic sampling by an extension to the consensus PHD filter. However, all of them are not suitable in the case of stochastic switches of multi-agent systems because of the fluctuation of the deterministic parameters and the timescales of communication and observation. To solve this problem, the mean square and almost sure convergence can be introduced into the PHD filter for the periodic multi-agent systems. In reference [32], the problems of random link failures, stochastic communication noises, and Markovian switching topologies are addressed and the mean square and almost sure convergence are established. However, for the multi-agent systems with event-based sampling mentioned in references [33] and [34], since the non-periodic event-based sampling is exploited, the DSPHD filter and our algorithm cannot be applied directly as they are taken only if some events happen rather than with a fixed sampling interval, which does not satisfy the model assumptions of the PHD filter. Therefore, extra approximation or other mechanism are required to adapt the PHD filter to such multi-agent systems. One possible solution may be developed according to the idea proposed in paper [35]. The author proposed a modified Kalman filter for target tracking combined with the send-on-delta sampling [36]. When no measurements are received, the last received measurement with a dynamic measurement noise can be used to update step.

4 Simulation results

In this section, we provide the results from experiments on simulation data to evaluate the performance of the proposed DCPPHD filter. The DCPPHD filter is compared to two traditional particle PHD filters with different particle numbers. The number of particles in DCPPHD is the same as that of the first particle PHD filter (PHD1). The numbers of particles used in the entire DCPPHD filter and PHD1 both are K times larger than that of another particle PHD filter (PHD2), i.e., the number of particles in PHD2 is equal to the number of particles in one PE of the DCPPHD filter. The purpose of the comparison with PHD2 is to verify whether the performance of the DCPPHD filter can degenerate to one PE. The algorithm is programmed with a MATLAB code running on an HP Z600 Workstation with 2x Intel Xeon processors at 2.53 GHz with 6-GB ram, running Windows 7 Professional 64 b.

4.1 Model

For simplicity, we consider a two-dimensional tracking scenario. The dynamic state model is expressed in the following form:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} T^2/2 & 0 \\ 1 & 0 \\ 0 & T^2/2 \\ 0 & 1 \end{bmatrix} \omega_k \quad (26)$$

where $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]$ is the target state vector at time k and $T = 1$ is the sampling period. $[x_k, y_k]$ is the position, and $[\dot{x}_k, \dot{y}_k]$ is the velocity. $w_k = [\omega_k^x, \omega_k^y]$ is the vector of independent zero-mean Gaussian white noise with standard deviations of $\begin{bmatrix} 0.025 & 0 \\ 0 & 4 \end{bmatrix}$. There is only a signal sensor in the scenario and the target-originated measurements are given by

$$\mathbf{z}_k = [R, \theta]^T = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan \frac{x_k}{y_k} \end{bmatrix} + \begin{bmatrix} v_k^R \\ v_k^\theta \end{bmatrix} \quad (27)$$

where v_k^R and v_k^θ are mutually independent zero-mean Gaussian white noise with variance $\sigma_R = 1m^2, \sigma_\theta = 0.01rad^2$. Clutter κ_k is uniformly distributed over the surveillance region with a Poisson distribution with a rate of λ_k . The probability of target survival is 0.9, and the detection probability is $P_{D,k} = 1$. The intensity of target birth is $\mathcal{N}(\cdot, \bar{x}, Q)$, where $\mathcal{N}(\cdot, \bar{x}, Q)$ denotes a normal density with mean \bar{x} and covariance Q . In this experiment, $\bar{x} = [0, 3, 0, -3]^T$ and $Q = \text{diag}[10; 1; 10; 1]$. The rates of clutter range over 3 different levels ($\lambda = 0, 10, 50$).

The surveillance region is $[-\pi, \pi] \times [-1000, 1000]$ rad m. There are eight PEs in the experiment. The initial number of particles is 2000.

Every existing or newborn target is assigned 200 particles in each group.

4.2 Simulation results

We run 100 independent simulations of the model given in Subsection 4.1.

The true trajectories of the five targets and the estimated positions by PHD1 and DCPPHD are plotted in Figs. 2a–c, respectively. The red points denote the estimated positions. As shown, the estimated positions of the

DCPPHD and PHD1 are similar, and they are all close to the true tracks.

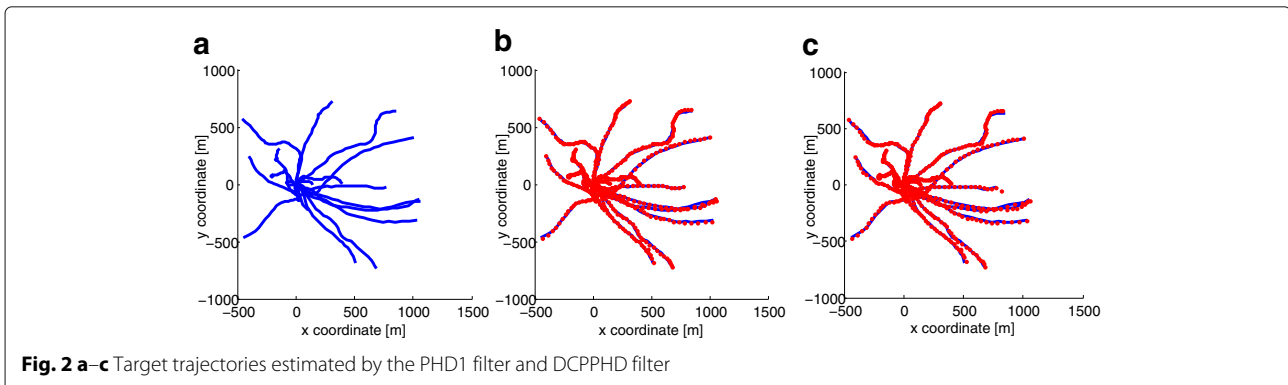
The true target number and the average estimated target number by DCPPHD, PHD1, and PHD2 at each scan are presented in Fig. 3. It can be observed that under the same simulation conditions, PHD1 and the DCPPHD filter achieve similar estimation accuracy, whereas PHD2 has the largest estimated error.

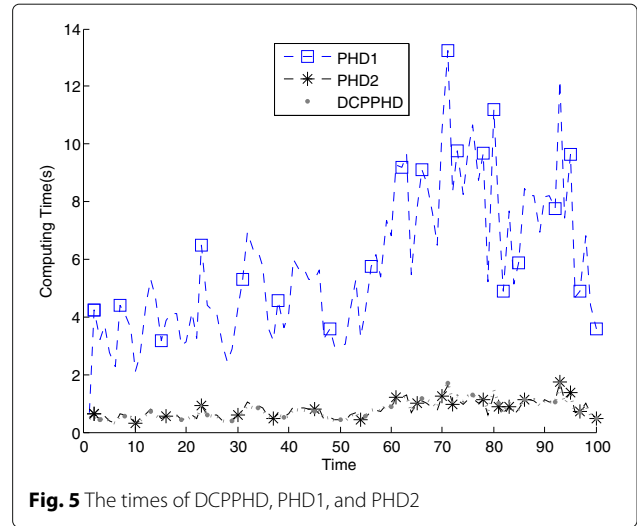
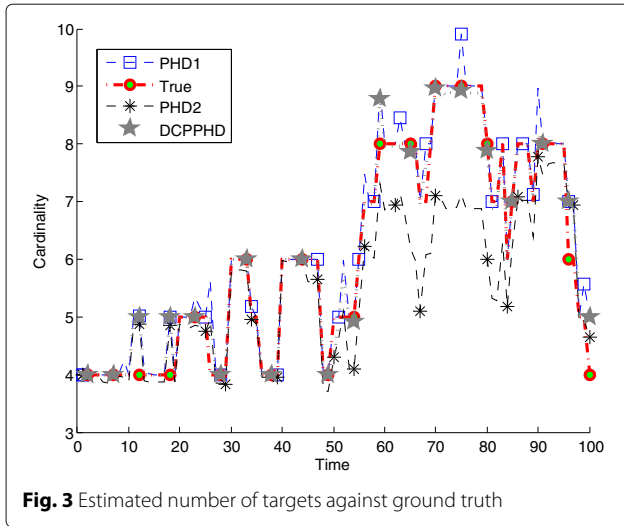
We use the optimal sub-pattern assignment (OSPA) as a multi-target miss-distance metric [37]. The parameters in OSPA are set as $p = 1$ and $c = 100$ in our evaluation. Figure 4 shows OSPA distances of the DCPPHD, PHD1, and PHD2. The times of these three methods are illustrated in Fig. 5. From Fig. 5, it is observed that the time of DCPPHD is considerably less than that of PHD1 and a slightly more than that of PHD2. Note that the time of DCPPHD is the execution time in theory.

Table 1 shows the average results over 100 Monte Carlo runs including the time, mean of OSPA, and standard deviation (std) of OSPA. One objective with the simulations is to assess the potential efficiency of DCPPHD. For this purpose, we record the following times

- T_{cp} is the average execution used by the operations that cannot be implemented in parallel in a DCPPHD. It is calculated according to the sum time of global estimation and particle exchange.
- T_{pi} is the average execution time of the operations that can be implemented in parallel in a DCPPHD.
- T_{si} is the average execution time of the implementation of the particle PHD filter. For DCPPHD, it is calculated according to $T_{si} = T_{cp} + T_{pi}/K$.

As shown in Table 1, the mean OSPA distances of the DCPPHD filter and PHD1 are less than that of PHD2 at different clutter rates, proving that the DCPPHD filter has comparable accuracy with PHD1 and outperforms PHD2. This result indicates that DCPPHD can achieve the same efficiency level of PHD1 but will not degenerate to the PHD2 level, i.e., one PE of DCPPHD. Although each PE





has the same number of particles as PHD2, the estimation results using the fusion strategy on eight PEs are better than the result of each PE.

The approximate times consumed by these approaches are also listed in Table 1, which shows that DCPPHD has great potential ($T_{pi}/K \gg T_{cp}$) for also achieving in a shorter execution time, producing the same level of accuracy as the traditional particle PHD filter. In different clutter environments, the theoretically maximum achievable accelerations are 3.38, 5.03, and 6.68, respectively.

4.3 Compared to the parallel particle PHD filter

In the proposed work of [38], all measurements at each filtering iteration will be separated and sent to each PE. Each PE runs an independent filter and obtains conditionally independent local estimates; from this aspect, it can

be viewed as a distributed algorithm. We compared our algorithm with this parallel PHD algorithm.

Figure 6 presents the OSPA metric results over 100 Monte Carlo trials for $r = 10$ with both algorithms. The results of both filters are very close. The average computing times of these algorithms are presented in Fig. 7.

It can be observed that our algorithm provides a comparable OSPA with the parallel PHD filter, which indicates that our filter has the similar accuracy to the parallel PHD filter. Although the execution times of these two filters are similar, note that the parallel PHD filter uses more PEs than our method, which suggests that our method can become more efficient if more PEs are used. However, the data communication of the parallel PHD filter is significantly larger than that of our method because it not only needs to broadcast all particles from the CU to all PEs and collect all the updated particles back to the CU but also broadcasts one measurement to one PE at each iteration, all of which cause significant communication

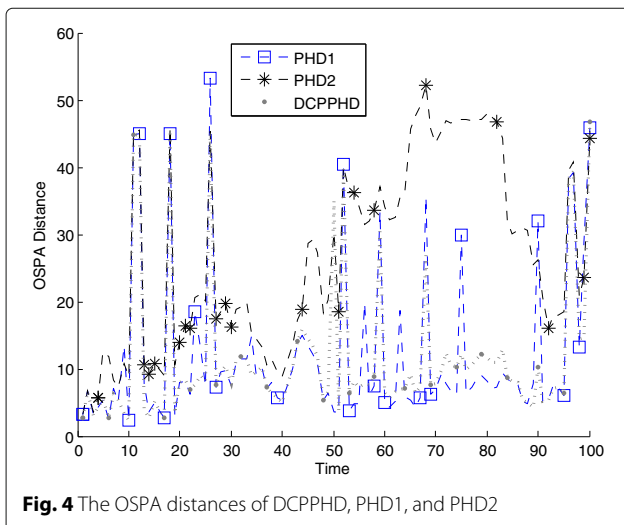


Table 1 The statistical data of simulation results at different rates of clutter

Method	λ	T_{pi} (s)	T_{cp} (s)	T_{si} (s)	Mean of OSPA	std of OSPA
PHD1	0	-	-	228.8114	6.6557	2.4451
PHD2	0	-	-	28.1430	9.2824	4.7546
DCPPHD	0	28.6594	6.9299	35.5893	7.1225	3.1121
PHD1	10	-	-	575.7523	11.7520	11.5318
PHD2	10	-	-	76.9675	29.9813	13.5204
DCPPHD	10	76.8237	7.0468	83.8705	11.4810	10.6069
PHD1	50	-	-	1542.2	30.1807	21.2412
PHD2	50	-	-	198.2446	32.2304	20.5954
DCPPHD	50	202.1936	5.8935	208.0871	25.3664	17.6369

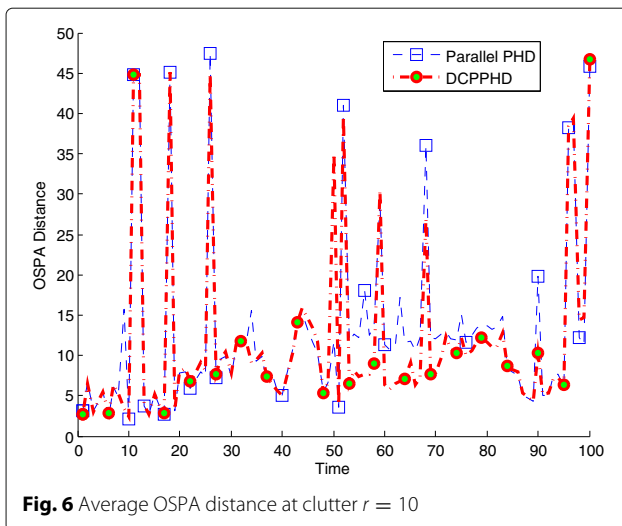


Fig. 6 Average OSPA distance at clutter $r = 10$

overhead. Our method simply allocates all the measurements to all the PEs, and the particles do not need to transit between the CU and PEs. Only part of the particles transit between different PEs simultaneously; therefore, the communication cost overhead of our method is much considerably.

5 Conclusions

This paper presents a distributed scheme of the particle PHD filter with several PEs and a CU. PEs run local decomposed particle PHD filters independently until part of the particles need to exchange among them; then, the CU associates and fuses the local state estimations submitted by all PEs. This architecture and fusion strategy can make the parallelization of local particle PHD filters possible and provides a comparable filtering accuracy with the sequential particle PHD filter. The advantage of the DCPPHD filter over the particle PHD filter lies in that the

DCPPHD filter can be implemented in parallel. The simulation results verify the performance of the DCPPHD filter in terms of accuracy and efficiency.

Possible topics for future work include the following. (1) Various methods that can be applied for the exchange step, such as random exchange [20], exchange depending on the divergence between the empirical distribution, and the complete particle population or exchange of the representative particles to further improve the filter's performance. (2) Implementation of the DCPPHD algorithm in practical WSNs, particularly under the constraints in the real-time operation and the communication capabilities (compared to a centralized PHD filter).

Acknowledgements

This paper was supported by the National Natural Science Foundation of China (NSFC) under Grants 61175027 and 61305013, by the Fundamental Research Funds for the Central Universities (Grant No. HIT.NSRIF.2014071), and by the Research Fund for the Doctoral Program of Higher Education of China (No. 20132302120044).

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Computer Science and Technology, Harbin Institute of Technology, XiZhida Road, 150001 Harbin, China. ²Department of Mathematic, Harbin Institute of Technology, XiZhida Road, 150001 Harbin, China. ³School of Computer Science and Technology, Harbin Institute of Technology, Xuefu Road, 150001 Harbin, China.

Received: 14 October 2015 Accepted: 1 November 2016

Published online: 24 November 2016

References

1. BN Vo, S Singh, A Doucet, Sequential Monte Carlo methods for multi-target filtering with random finite sets. *IEEE Trans. Aerosp. Electron. Syst.* **41**(4), 1224–1245 (2008)
2. RA Singer, JJ Stein, in *Decision and Control, 1971 IEEE Conference On. An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems*, vol. 10 (IEEE, Washington D.C, 1971), pp. 171–175
3. TE Fortmann, Y Bar-Shalom, M Scheffe, Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Ocean. Eng.* **8**(3), 173–184 (1983)
4. SS Blackman, Multiple hypothesis tracking for multiple target tracking. *IEEE Aerosp. Electron. Syst. Mag.* **19**(1), 5–18 (2004)
5. R Mahler, Multitarget bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.* **39**(4), 1152–1178 (2003)
6. B-N Vo, W-K Ma, The gaussian mixture probability hypothesis density filter. *IEEE Trans. Signal Process.* **54**(11), 4091–4104 (2006)
7. R Mahler, PHD filters of higher order in target number. *IEEE Trans. Aerosp. Electron. Syst.* **43**(4), 1523–1543 (2007)
8. BT Vo, BN Vo, A Cantoni, Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Trans. Signal Process.* **55**(7), 3553–3567 (2007)
9. B-T Vo, B-N Vo, A Cantoni, The cardinality balanced multi-target multi-bernoulli filter and its implementations. *IEEE Trans. Signal Process.* **57**(2), 409–423 (2009)
10. S Hong, L Wang, Z-G Shi, KS Chen, Simplified particle PHD filter for multiple-target tracking: algorithm and architecture. *Prog. Electromagn. Res.* **120**, 481–498 (2011)
11. Y Zheng, Z Shi, R Lu, S Hong, X Shen, An efficient data-driven particle PHD filter for multitarget tracking. *IEEE Trans. Ind. Inform.* **9**(4), 2318–2326 (2013)

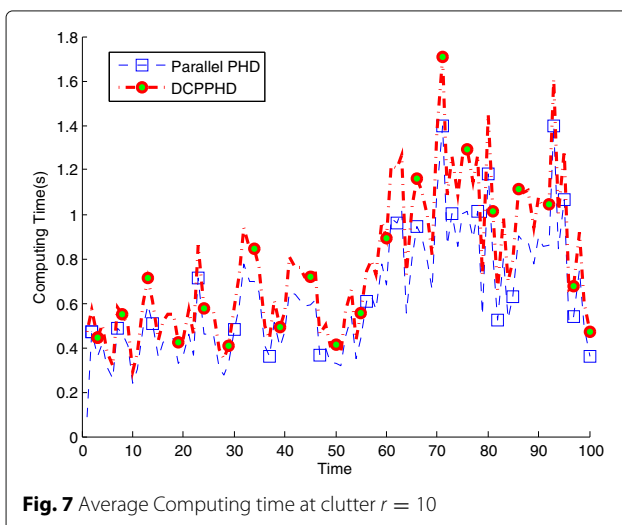


Fig. 7 Average Computing time at clutter $r = 10$

12. T Li, S Sun, TP Sattar, High-speed sigma-gating SMC-PHD filter. *Sig. Process.* **93**(9), 2586–2593 (2013)
13. Z-G Shi, Y Zheng, X Bian, Z Yu, Threshold-based resampling for high-speed particle PHD filter. *Prog. Electromagn. Res.* **136**, 369–383 (2013)
14. L Liu, X Zhang, H Ma, in *INFOCOM 2009, IEEE*. Dynamic node collaboration for mobile target tracking in wireless camera sensor networks (IEEE, Rio de Janeiro, 2009), pp. 1188–1196
15. X Zhang, Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking. *IEEE Trans. Autom. Control.* **56**(10), 2429–2444 (2011)
16. G Battistelli, L Chisci, C Fantacci, A Farina, A Graziano, Consensus CPHD filter for distributed multitarget tracking. *IEEE J. Sel. Top. Sign. Process.* **7**(3), 508–520 (2013)
17. G Battistelli, L Chisci, C Fantacci, A Farina, RP Mahler, in *Signal processing, sensor/information fusion, and target recognition XXIV*. Distributed fusion of multitarget densities and consensus PHD/CPHD filters (SPIE, Baltimore, 2015), pp. 94740–94740
18. L Zhao, P Ma, X Su, H Zhang, in *Information Fusion (FUSION), 2010 13th Conference On*. A new multi-target state estimation algorithm for PHD particle filter (IEEE, Edinburgh, 2010), pp. 1–8
19. K Panta, B-N Vo, S Singh, A Doucet, in *Signal processing, sensor fusion, and target recognition XIII*. Probability hypothesis density filter versus multiple hypothesis tracking, vol. 5429 (SPIE, Orlando, 2004), pp. 284–295
20. M Bolic, PM Djuric, S Hong, Resampling algorithms and architectures for distributed particle filters. *IEEE Trans. Sign. Process.* **53**(7), 2442–2450 (2005)
21. R Streit, The probability generating functional for finite point processes, and its application to the comparison of phd and intensity filters. *J. Adv. Inf. Fusion.* **8**, 119–132 (2013)
22. JL Gustafson, Reevaluating amdahl's law. *Commun. ACM.* **31**(5), 532–533 (1988)
23. SA Pasha, B-N Vo, HD Tuan, W-K Ma, A gaussian mixture PHD filter for jump Markov system models. *IEEE Trans. Aerosp. Electron. Syst.* **45**(3), 919–936 (2009)
24. C Ouyang, H-b Ji, Z-q Guo, Extensions of the SMC-PHD filters for jump Markov systems. *Sign. Process.* **92**(6), 1422–1430 (2012)
25. W Li, Y Jia, Gaussian mixture PHD filter for jump Markov models based on best-fitting Gaussian approximation. *Sign. Process.* **91**(4), 1036–1042 (2011)
26. Y Wei, J Qiu, S Fu, Mode-dependent nonrational output feedback control for continuous-time semi-Markovian jump systems with time-varying delay. *Nonlinear Anal. Hybrid Syst.* **16**, 52–71 (2015)
27. Y Wei, J Qiu, HR Karimi, M Wang, Filtering design for two-dimensional Markovian jump systems with state-delays and deficient mode information. *Inf. Sci.* **269**, 316–331 (2014)
28. F Previtali, L Iocchi, in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2015 IEEE International Conference On*. Ptracking: distributed multi-agent multi-object tracking through multi-clustered particle filtering (IEEE, 2015), pp. 110–115
29. AT Kamal, JH Bappy, JA Farrell, AK Roy-Chowdhury, Distributed multi-target tracking and data association in vision networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(7), 1397–1410 (2016)
30. R Claessens, A de Waal, P de Villiers, A Penders, G Pavlin, K Tuyls, in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems AAMAS '15*. Multi-agent target tracking using particle filters enhanced with context data: (demonstration) (ACM, Istanbul, 2015), pp. 1933–1934
31. C Lin, Z Lin, R Zheng, G Yan, G Mao, Distributed source localization of multi-agent systems with bearing angle measurements. *IEEE Trans. Autom. Control.* **61**(4), 1105–1110 (2016)
32. Q Zhang, J-F Zhang, Distributed parameter estimation over unreliable networks with Markovian switching topologies. *IEEE Trans. Autom. Control.* **57**(10), 2545–2560 (2012)
33. Y Fan, L Liu, G Feng, Y Wang, Self-triggered consensus for multi-agent systems with zero-free triggers. *IEEE Trans. Autom. Control.* **60**(10), 2779–2784 (2015)
34. Y Fan, J Yang, Average consensus of multi-agent systems with self-triggered controllers. *Neurocomputing.* **177**, 33–39 (2016)
35. JW Marck, J Sijts, in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference On*. Relevant sampling applied to event-based state-estimation (IEEE, Venice, 2010), pp. 618–624
36. YS Suh, VH Nguyen, YS Ro, Modified Kalman filter for networked monitoring systems employing a send-on-delta method. *Automatica.* **43**(2), 332–338 (2007)
37. D Schuhmacher, B-T Vo, B-N Vo, A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Sign. Process.* **56**(8), 3447–3457 (2005)
38. T Li, S Sun, M Bolić, JM Corchado, Algorithm design for parallel implementation of the SMC-PHD filter. *Sign. Process.* **119**, 115–127 (2016)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
