sensors

Article

An Efficient Implementation Method for Distributed Fusion in Sensor Networks Based on CPHD Filters

Liu Wang and Guifen Chen *

School of Electronic and Information Engineering, Changchun University of Science and Technology, Changchun 130022, China; wangl95@ccu.edu.cn
* Correspondence: chenguifencclg@163.com

Abstract: A highly efficient implementation method for distributed fusion in sensor networks based on CPHD filters is proposed to address the issues of unknown cross-covariance fusion estimation and long fusion times in multi-sensor distributed fusion. This method can effectively and efficiently fuse multi-node information in multi-target tracking applications. Discrete gamma cardinalized probability hypothesis density (DG-CPHD) can effectively reduce the computational burden while ensuring computational accuracy similar to that of CPHD filters. Parallel inverse covariance intersection (PICI) can effectively avoid solving high-dimensional weight coefficient convex optimization problems, reduce the computational burden, and efficiently implement filtering fusion strategies. The effectiveness of the algorithm is demonstrated through simulation results, which indicate that PICI-GM-DG-CPHD can substantially reduce the computational time compared to other algorithms and is more suitable for distributed sensor fusion.

Keywords: distributed fusion; DG-CPHD; parallel inverse covariance intersection; wave filter

1. Introduction

The fusion estimation of multiple sensors can be divided into two types, centralized fusion and distributed fusion, as shown in Figure 1 [1]. The robustness and flexibility of distributed fusion offer several advantages. Distributed fusion sensors have a low data communication link bandwidth, allowing the efficient transmission of information. Additionally, the scalability of these sensors is excellent, making them adaptable to varying needs. Moreover, distributed fusion enables the acquisition of an optimal fusion algorithm when the correlation conditions are known. It is widely used in multi-sensor and multi-target tracking [2–7]. Scholars have conducted in-depth research on multi-sensor distributed fusion strategies. For example, Yu [8] suggested a method for distributed fusion filtering based on dynamic communication weights to tackle the issue of fusion errors in incomplete node information within distributed sensor networks. The proposed method aims to improve the situation by optimizing the fusion process in a finite time manner. Senel [9] conducted research and analysis in the fields of multi-sensor data fusion and real-time multi-target tracking. Liu [10] proposed a distributed sequential ellipsoidal intersection fusion estimation algorithm. This algorithm combines the local filter estimates obtained at the fusion center in order to achieve more precise fusion localization results. Additionally, it achieves this objective with reduced computational costs. Su [11] proposed an improved adaptive birth model based on pre-segmentation measurement in his research, which solves the problems of measurement origin uncertainty and cardinality overestimation in the distributed fusion of sensor networks. Pal [12] proposed a distributed fusion underwater wireless sensor network using local tracker optimal quantization estimation in his research.

Citation: Wang, L.; Chen, G. An Efficient Implementation Method for Distributed Fusion in Sensor Networks Based on CPHD Filters. *Sensors* **2024**, *24*, 117. https://doi.org/10.3390/s24010117

Academic Editor: Zahir M. Hussain

Received: 16 November 2023
Revised: 15 December 2023
Accepted: 16 December 2023
Published: 25 December 2023

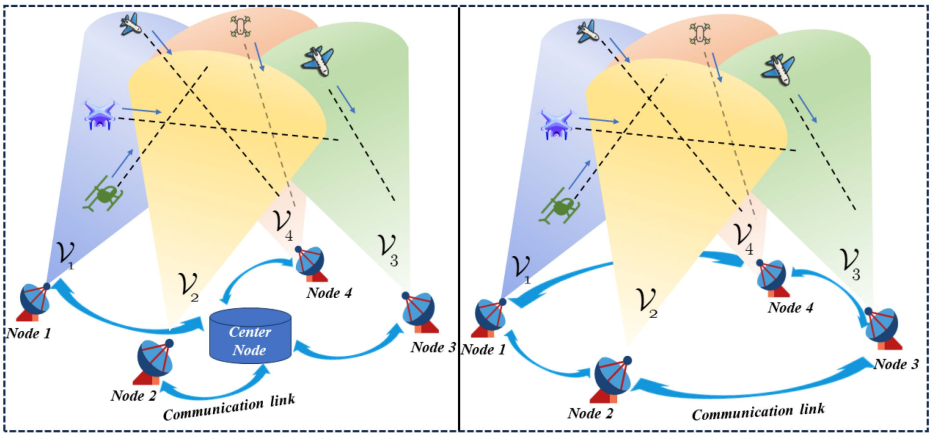


Figure 1. Schematic diagram of centralized sensor network and distributed sensor network.

In traditional distributed sensor networks, it can be assumed that nodes $1, 2, \cdots, l$ are synchronized and the common sampling period is T. $\pi_a(X^a_{kT} \Big| ZX^a_{(1:k)T})$, $a \in l$, represents the posterior test of the sensor. Node 1 needs to send the posterior of the sensor to the next node (node 2) before fusion, as shown in Figure 2. However, in the actual process of distributed fusion in sensor networks, the accuracy of fusion is affected by factors such as the startup time, sampling period, and communication delay. Therefore, it is necessary to align the time and then fuse the predicted posterior set based on fusion criteria. A covariance intersection fusion (CI fusion) algorithm is proposed to address issues related to multi-sensor fusion estimation with unknown cross-covariance. CI fusion has the advantage of being able to combine covariance without taking into account the correlation between nodes [13–18], allowing for a conservative estimation. Scholars have conducted in-depth research on covariance cross-fusion. For example, Sun [19] proposed a distributed estimation approach utilizing inverse covariance intersection. This algorithm was proven to be useful in efficiently addressing challenges related to cross-correlation and dynamic-state estimation in wireless sensor networks. Liu [20] proposed sequential inverse covariance intersection (SICI) fusion Kalman filtering, which avoids the calculation of the cross-covariance between local filters and is less conservative than the previous sequential covariance intersection (SCI) filtering; Sun [21] introduced an algorithm that utilizes inverse covariance intersection (ICI) technology for distributed iterative diffusion estimation. The aim is to address correlation problems with unknown solutions and achieve accurate uncertainty estimates similar to the true mean square error. Hu [22] proposed, in his research, that, by analogy with batch CI fusion, hierarchical covariance intersection (HCI) fusion independent of the hierarchical structure and SCI fusion independent of the fusion order can be proposed, which can avoid the potential negative impact of uncertainty in the fusion process. Wang B [23] introduced a distributed multi-target tracking algorithm using a multi-Bernoulli (MB) filter that is based on the generalized covariance intersection (GCI). Wang L [24] proposed a method in his research that utilizes the inverse covariance cross-fusion algorithm to fuse the signals of various subsystems, solving the convex optimization problem of high-dimensional weight coefficients, reducing the burden of data computation, and minimizing the data fusion time. Park [25] proposed the inverse covariance intersection (ICI) in his research, extending it to multi-sensor and multi-objective tracking systems. However, the above works mainly focus on the research and analysis of the fusion strategy, without considering its efficient implementation. Although the fusion criteria can improve the operation accuracy of the system to a certain extent, doing so may reduce the operation efficiency. The main reasons can be summarized as follows: the complexity of the filtering methods increases with the number of targets, the complexity of the filtering methods increases exponentially with the operation time, the high-dimensional weight coefficient convex optimization problem of the fusion criteria themselves occurs, the reuse of fusion criteria increases the operation time, and the uncertainty caused by the hierarchy and order in the fusion process will affect the efficiency of the algorithm to a certain extent.

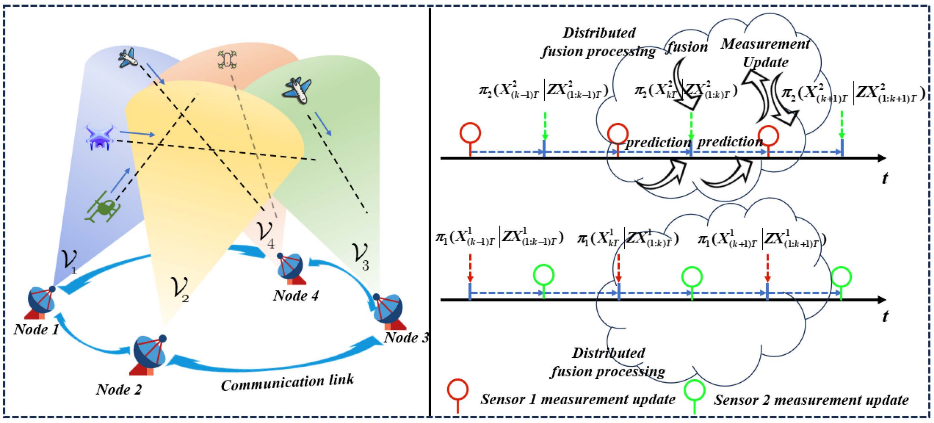

$\mathcal{V}_1$
$\mathcal{V}_2$
$\mathcal{V}_3$
$\mathcal{V}_4$
Node 1
Node 2
Node 3
Node 4
Communication link
Distributed fusion processing
fusion
Measurement Update
$\pi_2(X^2_{(k-1)T} | ZX^2_{(1:k-1)T})$
$\pi_2(X^2_{kT} | ZX^2_{(1:k)T})$
$\pi_2(X^2_{(k+1)T} | ZX^2_{(1:k+1)T})$
prediction
prediction
t
$\pi_1(X^1_{(k-1)T} | ZX^1_{(1:k-1)T})$
$\pi_1(X^1_{kT} | ZX^1_{(1:k)T})$
$\pi_1(X^1_{(k+1)T} | ZX^1_{(1:k+1)T})$
t
Distributed fusion processing
Sensor 1 measurement update
Sensor 2 measurement update

Figure 2. Distributed fusion processing of sensor networks.

Scholars have conducted in-depth research and analysis on the efficient implementation of multi-sensor fusion strategies. For example, Li [26] proposed a real-time fusion algorithm for asynchronous sensor systems and a real-time sequential (RTS) fusion method, RTS-GCI, based on generalized covariance intersection (GCI) to achieve fast sensor filtering. The implementation of distributed fusion based on maximum probability association (MPA-DF), proposed by Li [27,28], and the generalized covariance intersection (GCI) distributed fusion method (referred to as GCI-DF) can achieve efficient filtering to a certain extent. In the above research, by improving the fusion strategy and adjusting the level and order of the fusion process to reduce the computational burden, the operational efficiency of the algorithm could be improved to a certain extent. Therefore, it can be seen that the efficient implementation of a fusion strategy applied to a multi-sensor multi-target system is one of the future development directions of multi-sensor distributed fusion multi-target tracking.

This article proposes an efficient implementation method for the distributed fusion of sensor networks based on cardinalized probability hypothesis density (CPHD) filters, which addresses the issues of unknown cross-covariance fusion estimation and long fusion times in multi-sensor distributed fusion. The parallel inverse covariance intersection (PICI) fusion algorithm is applied in local estimation fusion, effectively addressing the issue of the convex optimization of weight coefficients with a high number of sensors in multi-sensor distributed fusion. Additionally, the PICI fusion algorithm is employed in discrete gamma cardinalized probability hypothesis density (DG-CPHD) and Gaussian mixture discrete gamma cardinalized probability hypothesis density (GM-DG-CPHD), successfully reducing the running time. The simulation results demonstrate the efficacy of the algorithm.

2. Research Background

2.1. GM-CPHD Filter

The multi-objective density function $f(\mathrm{X})$ is a real-valued function $RFS\mathrm{X} = \{x_1, \cdots, x_n\}$, and the multi-objective density function is a characteristic of a random finite set (RFS). The set integral of the multi-objective density function is defined as

$$\int f(\mathrm{X})\delta\mathrm{X} \triangleq f(\phi) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{x_1, \cdots, x_n\}) dx_1 \cdots dx_n \tag{1}$$

where X is the multi-target set and n is the target quantity.

The probability-generating function (PGF) $G[h]$ and probability hypothesis density $D(x)$ are defined as

$$G[h] \triangleq \int f(\mathrm{X}) \prod_{x \in \mathrm{X}} h(x) \delta\mathrm{X} \tag{2}$$

$$D(\mathrm{x}) \triangleq \frac{\delta}{\delta \mathrm{x}} G[h]|_{h=1} \tag{3}$$

2.1.1. CPHD Filter

The CPHD filter [29–31] merges the propagation intensity function and cardinality distribution (which represents the probability distribution of the target number) by considering the clutter RFS as an independent and identically distributed (i.i.d.) process.

Assumption 1. *Both prior and posterior multi-objective random finite sets are i.i.d. clustering processes.*

Assumption 2. *Each target moves independently, and a single target undergoes a Markov transition density $p_{t,k|k-1}(x|x')$*

Assumption 3. *Newly born targets can independently appear in sports scenarios. Each target moves independently, and a single target undergoes Markov transfer density $p_{t,k|k-1}(x|x')$. The density of the newborn target is $b_{t,k|k-1}(X)$, the strength function of the random finite set is expressed as $D(x)$, and the PGFL is $G[h]$.*

Assumption 4. *The sensor detects a single target with state x as $p_{d,k}(x)$at time step k with probability p_d.*

The CPHD filter's definition is based on a multi-object distribution and encompasses processes that are independent and identically distributed. If we assume that the cardinality distribution $p(n)$ of the point process is $|\mathrm{X}| = n$, then the PGFL $G[h]$ and the probability assumption density $D(x)$ of CPHD can be stated as

$$f(\mathrm{X}) \triangleq n! \cdot p(n) \cdot f(\mathrm{x}_1) \cdots f(\mathrm{x}_n) \tag{4}$$

$$G[h] = \sum_{n=0}^{\infty} p(n) (\int h(u) \cdot f(u) du)^{n} \tag{5}$$

$$D(\mathrm{x}) \triangleq \frac{\delta}{\delta x} G[h]|_{h=1} = f(x) \sum_{n=1}^{\infty} n \cdot p(n) \tag{6}$$

1. Prediction of CPHD

$$p_{k,t|t-1}(n) = \sum_{j=0}^{n} p_b(n-j) \sum_{h=j}^{\infty} \binom{h}{j} p_{k,t}^{j} (1-p_{k,t})^{h-j} p_{t-1|t-1}(h) \tag{7}$$

$$D_{k|k-1}(x) = D_b(x) + \int {}_X p_s(x') p_t(x|x') D_{k|k-1}(x') dx' \tag{8}$$

$$\hat{N}_{k|k-1} = \hat{N}_{a,k} + \hat{N}_{b,k} \tag{9}$$

Here, $p_s(x')$ is a known objective transition function with a previous state; $p_t(x|x')$ is the survival probability of the target with a previous state; $\hat{N}_{a,k}$ is the expected number of new goals; and $\hat{N}_{b,k}$ is the expected number of goals that survive in time step $k-1$.

2. Updates to CPHD

$$p_{k,t|t}(n) = \frac{\mathcal{L}_t^0(d_{k,t|t-1}(\cdot), y_t, n) p_{k,t|t-1}(n)}{\sum_{i=0}^{\infty} \mathcal{L}_t^0(d_{k,t|t-1}(\cdot), y_t, i) p_{k,t|t-1}(i)} \tag{10}$$

$$d_{k,t|t}(x) = \mathcal{L}_{yt}(x) d_{k,t|t-1}(x) \tag{11}$$

Here, 1 is the generalized likelihood function.

3. The complexity of CPHD

The complexity of PHD recursion is $O(mn_{\max})$, while the CPHD algorithm can be seen as evaluating the elementary pair production function m + 1 times, with complexity of $\mathcal{O}(m^3) \sim \mathcal{O}|Z|^3 n_{\max})$, where m is the number of evaluations and $n_{\max}$ is the maximum number of targets. When $n_{\max} > m$, CPHD can be considered as $(m^3 + n_{\max} m^2 - \frac{m^2}{2}) \sim \mathcal{O}(n_{\max} m^2)$.

2.1.2. GM-CPHD Filter

Based on the above conditions, the recursive form of GM-CPHD can be given as follows.

1. Prediction of GM-CPHD

If we make the assumption that the posterior intensity at time $k-1$ follows a Gaussian mixture in the given format, then

$$D'_{k|k-1}(x) = \sum_{i=1}^{J_{k-1}} \omega^i_{k-1} \mathcal{N}(x; m^i_{k-1}, P^i_{k-1}) \tag{12}$$

The predicted k-time intensity is also a Gaussian mixture, given by the following equation:

$$D'_{k|k-1}(x) = d = D'_{s,k|k-1}(x) + \gamma_k(x) \tag{13}$$

$$D'_{s,k|k-1}(x) = p_{s,k} \sum_{j=1}^{J_{k-1}} \omega^j_{k-1} \mathcal{N}(x; m^j_{k-1}, P^j_{k-1}) \tag{14}$$

$$m^j_{s,k|k-1} = F_{k-1} m^j_{k-1} \tag{15}$$

$$P^j_{s,k|k-1} = Q_{k-1} + F_{k-1} P^j_{k-1} F^T_{k-1} \tag{16}$$

$$D'_{\beta,k|k-1}(x) = \sum_{j=1}^{J_{k-1}} \sum_{l=1}^{J_{\beta,k}} \omega^j_{k-1} \omega^l_{\beta,k} \mathcal{N}(x; m^{j,l}_{\beta,k|k-1}, P^{j,l}_{\beta,k|k-1}) \tag{17}$$

$$m^{j,l}_{\beta,k|k-1} = F^l_{\beta,k-1} m^j_{k-1} + d^l_{\beta,k-1} \tag{18}$$

$$P^{j,l}_{\beta,k|k-1} = Q^l_{\beta,k-1} + F^l_{\beta,k-1} P^j_{\beta,k-1} (F^l_{\beta,k-1})^T \tag{19}$$

2. Update for GM-CPHD

Assuming that the predicted strength has a Gaussian mixture form, it is

$$D'_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} \omega^i_{k|k-1} \mathcal{N}(x; m^i_{k|k-1}, P^i_{k|k-1}) \tag{20}$$

The predicted k-time intensity is also a Gaussian mixture, given by the following equation:

$$D'_k(x) = (1 + p_{d,k}) D'_{k|k-1}(x) + \sum_{z \in Z_k} D_{d,k}(x; z) \tag{21}$$

At this point,

$$D'_{d,k}(x; z) = \sum_{j=1}^{J_{k|k-1}} \omega^j_k \mathcal{N}(x; m^j_{k|k}, P^j_{k|k}) \tag{22}$$

$$\omega_k^j(z) = \frac{p_{d,k}\omega_{k|k-1}^j q_k^j(z)}{\kappa_k(z) + p_{d,k}\sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^l q_k^l(z)} \tag{23}$$

$$m_{k|k}^j(z) = m_{k|k-1}^j + K_k^j(z - H_k m_{k|k-1}^j) \tag{24}$$

$$P_{k|k}^j = [I - K_k^j H_k]P_{k|k-1}^j \tag{25}$$

$$K_k^j = P_{k|k-1}^j H_k^T (H_k P_{k|k-1}^j H_k^T + R_k)^{-1} \tag{26}$$

3. The complexity of GM-CPHD

At this point, the complexity of CPHD recursion is $\mathcal{O}(m^3) \sim \mathcal{O}|Z|^3 n_{\max})$, and the complexity of GM-CPHD will increase significantly as the number of targets increases.

2.2. Discrete Gamma CPHD Filter

2.2.1. DG-CPHD Filter

1. Prediction of DG-CPHD

Flávio [32,33] aimed, in his research, to model the cardinality distribution in CPHD filtering as a discrete gamma distribution, while ensuring cardinality accuracy and variance, reducing the complexity of CPHD filtering and achieving fast filtering. The DG-CPHD filtering process assumes that the states of multiple objectives follow a clustering process that is both independent and identically distributed. The distribution of cardinality is modeled using a discrete gamma distribution. The unique characteristic of this discretized gamma distribution is that it enables straightforward calculations for the approximation of the first-order and second-order matrices of the posterior cardinality distribution. This approximation method effectively bypasses the limitations imposed by binomial filters.

The assumptions for the discrete gamma CPHD filter are

$$p_\Xi(X) = dGamma_{\alpha,\beta}(n) \cdot n!\prod_{i=1}^{n} p(x_i) \tag{27}$$

where 1 is the spatial probability density of each object.

The predicted intensity function is

$$D''_{k|k-1}(x) = D''_b(x) + \int {}_X p_s(x')p_t(x|x')D''_{k|k-1}(x')dx' \tag{28}$$

The parameters of the posterior discrete gamma cardinality distribution at time $k-1$, with a ratio of α_{k-1} (shape) and β_{k-1} (rate), are

$$\alpha_{k|k-1} = \frac{\mu_{N,k|k-1}^2}{\sigma_{N,k|k-1}^2} \tag{29}$$

$$\beta_{k|k-1} = \frac{\mu_{N,k|k-1}}{\sigma_{N,k|k-1}^2} \tag{30}$$

$$\hat{N}_{k|k-1} = \hat{N}_{a,k} + \hat{N}_{b,k} = \mu_{N,k|k-1} \tag{31}$$

$$\sigma_{N,k|k-1}^2 = \hat{N}_{a,k} + \hat{N}_{b,k} + \langle p_s, \varsigma\rangle^2 \alpha_{k-1}\beta_{k-1}^{-1}(\beta_{k-1}^{-1} - 1) \tag{32}$$

The primary distinction lies in how the estimated distribution of cardinality is approximated. In the case of the DG-CPHD filter, this approximation is represented by the discrete gamma cardinality parameters α_{k-1} and β_{k-1}.

2. Update of DG-CPHD

$$D''_{k|k-1}(x) = \frac{p_d(x)}{\mu_{N,k|k+1}} \sum_{z \in Z_k} \frac{l_d(x)}{\lambda c(z)} \Theta_k[Z_k \backslash \{z\}] D''_{k|k-1}(x) + \frac{p_d(x)}{\mu_{N,k|k+1}} \Theta_k[Z_k] D''_{k|k-1}(x) \tag{33}$$

$$\alpha_{k|k-1} = \frac{\mu^2_{N,k}}{\sigma^2_{N,k}} \tag{34}$$

$$\beta_{k|k-1} = \frac{\mu_{N,k}}{\sigma^2_{N,k}} \tag{35}$$

Define $\mu_{N,k|k-1}$ as $\alpha_{k|k-1}\beta^{-1}_{k|k-1}$

$$\mu_{N,k} = \theta_{1,0} + \theta_{0,1} \cdot \langle p_d, \varsigma \rangle \tag{36}$$

$$\sigma^2_{N,k} = \theta_{2,0} - \theta_{1,0} + 2\theta_{1,1} \cdot \langle p_d, \varsigma \rangle + \theta_{0,2} \cdot \langle p_d, \varsigma \rangle^2 - (\mu^2_{N,k} - \mu_{N,k}) \tag{37}$$

$$\Theta_k[Z] \triangleq \frac{\sum_{j=0}^{|Z|} \hat{G}^{j+1}(\langle q_d, \varsigma \rangle) \sigma_{m=|Z|,j}(\frac{\langle q_d l_{z1}, \varsigma \rangle}{\lambda c(z_1)}, \cdots, \frac{\langle q_d l_{zm}, \varsigma \rangle}{\lambda c(z_m)})}{\sum_{i=0}^{m_k} \hat{G}^{i}(\langle q_d, \varsigma \rangle) \sigma_{m=|Z_k|,i}(\frac{\langle q_d l_{zk1}, \varsigma \rangle}{\lambda c(z_{1k})}, \cdots, \frac{\langle q_d l_{zkm}, \varsigma \rangle}{\lambda c(z_{km})})} \tag{38}$$

$$\theta_{u,v} \triangleq \frac{\sum_{j=0}^{m_k} j^u \hat{G}^{j+v}(\langle q_d, \varsigma \rangle) \sigma_{m=|Z_k|,i}(\frac{\langle q_d l_{zk1}, \varsigma \rangle}{\lambda c(z_{1k})}, \cdots, \frac{\langle q_d l_{zkm}, \varsigma \rangle}{\lambda c(z_{km})})}{\sum_{i=0}^{m_k} \hat{G}^{i}(\langle q_d, \varsigma \rangle) \sigma_{m=|Z_k|,j}(\frac{\langle q_d l_{zk1}, \varsigma \rangle}{\lambda c(z_{1k})}, \cdots, \frac{\langle q_d l_{zkm}, \varsigma \rangle}{\lambda c(z_{km})})} \tag{39}$$

The primary distinction lies in how the estimated distribution of cardinality is approximated. In the case of the DG-CPHD filter, this approximation is represented by the discrete gamma cardinality parameters α_{k-1} and β_{k-1}.

3. The complexity of DG-CPHD

The computational complexity of CPHD is $O(m^3 n \max)$, and the DG-CPHD filter is constrained to using a finite number of terms, $\overline{n}_{\max}$, to approximate derivatives of the probability-generating function (PGF) $G[h]$. The results demonstrate that $\overline{n}_{\max}$ is not as prone to sensitivity to the number of target-generated measurements.

References [32,33] show that when the value of λ increases, the complexity of both the CPHD and DG-CPHD filters is primarily influenced by the total number of measurements. This means that as the number of false alarms increases, the running times of these filters increase at a sub-exponential rate, with a maximum limit of $O(m^3)$ operations.

2.2.2. GM-DG-CPHD Filter

1. Prediction of GM-DG-CPHD

Based on the above GM-CPHD filtering process, the GM-DG-CPHD prediction process can be written as

$$D'''_{k|k-1}(x) = \sum_{i=1}^{J_{k-1}} \omega^i_{k|k-1} \mathcal{N}(x; m^i_{k|k-1}, P^i_{k|k-1}) \tag{40}$$

$\alpha_{k|k-1}\beta^{-1}_{k|k-1}$ are the same as above, but, at this point,

$$\mu_{N,k|k-1} = \sum_{i=1}^{I_{b,k}} \omega^i_{b,k} + \alpha_{k|k-1}\beta^{-1}_{k|k-1} \tag{41}$$

$$\sigma^2_{N,k|k-1} = \mu_{N,k|k-1} + p_s^2 \alpha_{k|k-1}\beta^{-1}_{k|k-1}(\beta^{-1}_{k|k-1} - 1) \tag{42}$$

2. Update of GM-DG-CPHD

$$D'''_{k|k-1}(x) = \sum_{i=1}^{J_{k-1}} \omega_k^i \mathcal{N}(x; m_k^i, P_k^i) \tag{43}$$

$\alpha_{k|k-1}\beta^{-1}_{k|k-1}$ are the same as above, but, at this point,

$$\mu_{N,k} = \theta_{1,0} + \theta_{0,1} \cdot (1 - p_d) \tag{44}$$

$$\sigma^2_{N,k} = \theta_{2,0} - \theta_{1,0} + 2\theta_{1,1} \cdot (1 - p_d) + \theta_{0,2} \cdot (1 - p_d)^2 - (\mu^2_{N,k} - \mu_{N,k}) \tag{45}$$

$$w^j_{u,k} = \frac{(1 - p_d)\Theta_k[Z_k]}{\alpha_{k|k-1}\beta^{-1}_{k|k-1}} w^j_{u,k|k-1} \tag{46}$$

$$m^j_{u,k} = m^j_{k|k-1} \tag{47}$$

$$P^j_{u,k} = P^j_{k|k-1} \tag{48}$$

3. The complexity of GM-DG-CPHD

Similarly, the complexity of GM-DG-CPHD will increase significantly as the number of targets increases.

2.3. GCI Fusion Strategy

In the fusion of distributed sensor networks, the GCI fusion algorithm is traditionally used. The core of the GCI fusion criterion is to minimize the weighted sum of the Kullback–Leibler divergence (KLD) relative to a given distribution set by fusing a posterior. Under the GCI fusion rule, the common product form of Bayesian rules is replaced by the exponential mixing of geometric means or distributions [34].

$$\pi_\omega(X_{kT} \Big| Z^1_{(1:k)T}, Z^1_{(1:k)T}) = \frac{\pi_1(X^1_{kT} \Big| Z^1_{(1:k)T})^{\omega_1} \pi_2(X^2_{kT} \Big| Z^2_{(1:k)T})^{\omega_2}}{\int \pi_1(X^1_{kT} \Big| Z^1_{(1:k)T})^{\omega_1} \pi_2(X^2_{kT} \Big| Z^2_{(1:k)T})^{\omega_2} \delta X} \tag{49}$$

$$\pi_\omega = \underset{\omega}{\operatorname{argmin}}(1 - \omega)D(\pi|\pi_1) + \omega D(\pi|\pi_2) \tag{50}$$

Here, D is the Kullback–Leibler divergence, and $\omega_1 + \omega_2 = 1$.

However, the disadvantage of the GCI fusion strategy is that it generates many Gaussian components after fusion, increasing the computational burden and complexity. There are also some Gaussian components with low weights that need to be pruned and merged in subsequent calculations.

3. A Fast Fusion Method for Discrete Gamma CPHD Real-Time Sequences

3.1. PICI Fusion Algorithm

Common fusion methods include batch inverse covariance intersection (BICI), sequential inverse covariance intersection (SICI), and parallel inverse covariance intersection (PICI). As shown in Figure 3, BICI fusion enhances convex optimization problems with high-dimensional weight coefficients among the mentioned solutions. On the other hand, SICI fusion eliminates the need to solve high-dimensional weight coefficient convex optimization problems, but, when there are too many sensors, it will consume a lot of time due to multiple sequential fusion.

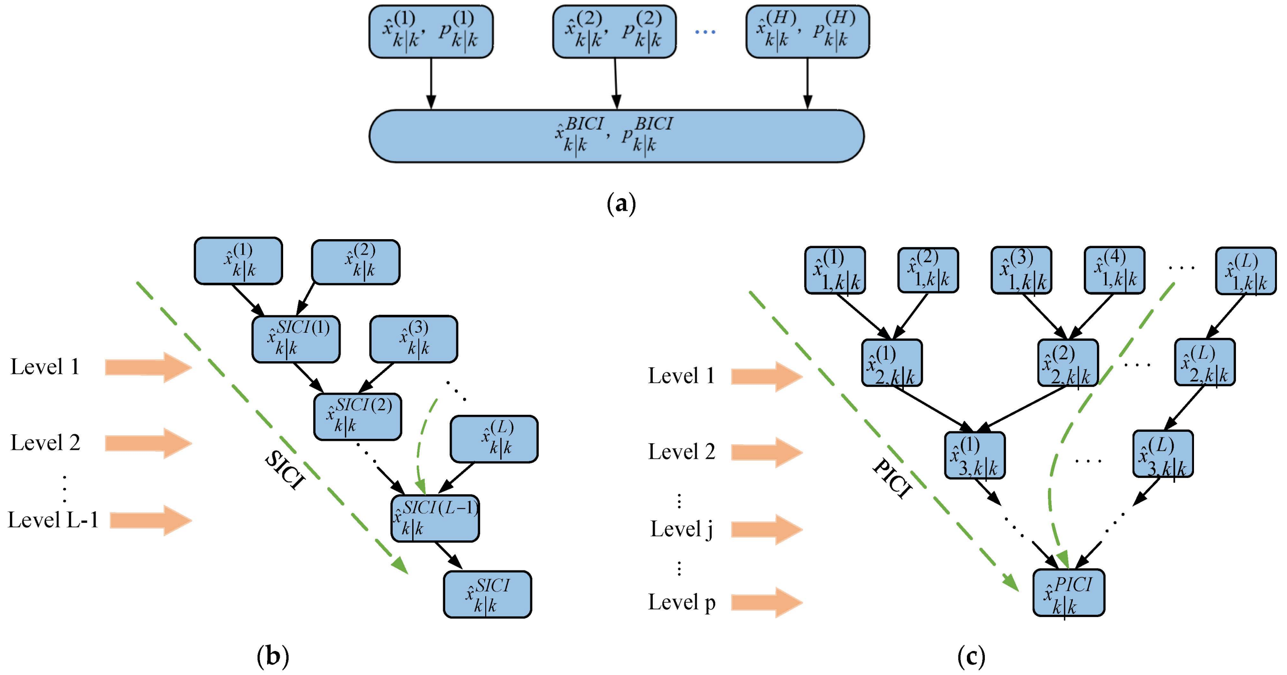


Figure 3. Common fusion strategies: (**a**) BICI integration strategy; (**b**) SICI integration strategy; (**c**) PICI integration strategy.

As the system incorporates multiple sensors, the SICI fusion algorithm necessitates the execution of L-1 pairwise fusions, which can be time-consuming. With a fixed number of sensors L, as the layers increase, the number of ICI fusion devices for the two sensors will progressively decrease in an exponential manner. Therefore, when L is particularly large, the PICI algorithm can save a lot of computational time. In order to reduce the computational time of the system, a parallel covariance intersection fusion algorithm is used for operation. For this purpose, this study uses the PICI fusion strategy to achieve the efficient implementation of distributed fusion in sensor networks.

In the PICI multi-sensor fusion algorithm, the fusion estimation of $\hat{x}^i_{k|k}(i, i+1, \cdots, L)$ at different levels is determined by the subsystem estimation $\hat{x}^{ICI,j-1}_{k|k,2i-1}, \hat{x}^{ICI,j-1}_{k|k,2i}$ and subsystem variance $P^{j-1}_{ICI,2i-1}, P^{j-1}_{ICI,2i}$ of the ICI fusion of two sensors in the previous layer. Therefore, the PICI fusion algorithm is

$$\hat{x}^{ICI,j}_{k|k,i} = K^j_{k,i}\hat{x}^{ICI,j-1}_{k|k,2i-1} + L^j_k\hat{x}^{ICI,j-1}_{k|k,2i} \tag{51}$$

$$P^j_{ICI} = [(P^{j-1}_{ICI,2i-1})^{-1} + (P^{j-1}_{ICI,2i})^{-1} - (\omega^j_i P^{j-1}_{ICI,2i-1} + (1-\omega^j_i)P^{j-1}_{ICI,2i})^{-1}]^{-1} \tag{52}$$

$$K^j_K = P^j_{ICI,i}[(P^{j-1}_{ICI,2i-1})^{-1} - \omega^j_i(\omega^j_i P^{j-1}_{ICI,2i-1}) + (1-\omega^j_i)P^{j-1}_{ICI,2i})^{-1}] \tag{53}$$

$$L^j_{k,i} = P^j_{ICI,i}[(P^{i-1}_{ICI,2i})^{-1} - (1-\omega^j_i)(\omega^j_i P^{i-1}_{ICI,2i-1} + (1-\omega^j_i)P^{i-1}_{ICI,2i})^{-1}] \tag{54}$$

where $i = 1, 2, \cdots, L_j, j = 1, 2, \cdots L_p$, and L_j, L_p represent the numbers of filters in the jth layer and the final number of layers, respectively.

For this reason, the initial value can be considered as

$$\hat{x}^{ICI,1}_{k|k,i} = \hat{x}^i_{k|k}, P^1_{ICI,i} = P^i_{k|k}, L_1 = L \tag{55}$$

The performance indicators for the minimization of $\omega \in [0,1]$ are

$$\min J == \min_{\omega_1 \in [0,1]} tr\left\{ \left[(\mathrm{P}^1_{k|k})^{-1} + (\mathrm{P}^2_{k|k})^{-1} - (\omega_1 \mathrm{P}^1_{k|k} + (1-\omega_1)(\mathrm{P}^2_{k|k})^{-1} \right]^{-1} \right\} \tag{56}$$

$$\min J = \min_{\omega_i^j \in [0,1]} tr\left\{ \left[(\mathrm{P}^{j-1}_{ICI,2i-1})^{-1} + (\mathrm{P}^{j-1}_{ICI,2i})^{-1} - (\omega_i^j \mathrm{P}^{j-1}_{ICI,2i+1} + (1-\omega_i^j)(\mathrm{P}^{j-1}_{ICI,2i})^{-1} \right]^{-1} \right\} \tag{57}$$

In order to validate the effectiveness of the PICI fusion algorithm, this study chose an L value of 5 and conducted a comparative analysis on the effects of SCI, SICI, PCI, PICI, BCI, and BICI through covariance ellipse validation. The results of the comparison of various covariance intersection fusion algorithms can be seen in Figure 4. From this, it can be seen that the PICI fusion algorithm has higher fusion accuracy than other algorithms.

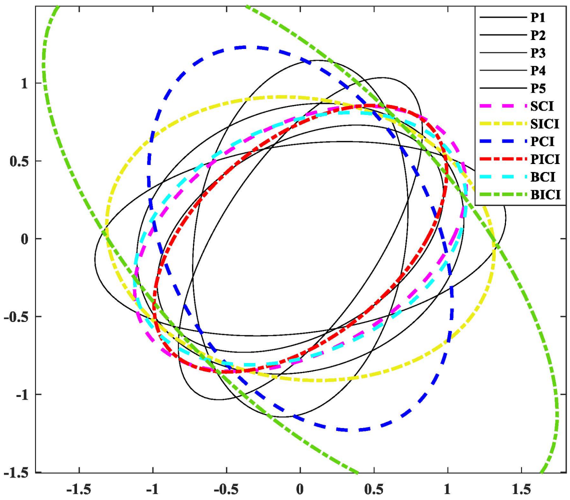


Figure 4. Comparison results of different covariance intersection fusion algorithms.

3.2. Implementation of PICI-GM-DG-CPHD Algorithm

For the convenience of calculation, according to references [24,25], PICI is rewritten as

$$\mathrm{x}_{PICI} = \mathrm{K}_{PICI}\mathrm{x}_A^{j-1} + \mathrm{L}_{PICI}\mathrm{x}_B^{j-1} \tag{58}$$

$$\begin{aligned} \mathrm{P}_{PICI} &= (\mathrm{P}_A^{j-1})^{-1} + (\mathrm{P}_B^{j-1})^{-1} - (\omega_{PICI}\mathrm{P}_A^{j-1} + (1-\omega_{PICI})\mathrm{P}_B^{j-1}) - 1)^{-1} \\ &= ((\mathrm{P}^{j-1}_{A,PICI})^{-1} + (\mathrm{P}^{j-1}_{B,PICI})^{-1})^{-1} \end{aligned} \tag{59}$$

From the above Equations (58) and (59), it can be seen that the fusion results of the PICI algorithm at the next level are determined by the fusion results of the previous level. To this end, PICI and ICI can be determined differently in terms of $\mathrm{P}_A^{-1}, \mathrm{P}_B^{-1}$, and PICI needs to be divided into $P^{A(M)}_{Lp,k|k}, P^{B(m)}_{Lp,k|k}$ types at different levels. When the number of sensors L is even, then $m = \frac{L}{2}$, and when the number of sensors L is odd, then $m = \frac{L-1}{2}$.

$$(\mathrm{P}^{(y)}_{A,ICI})^{-1} \triangleq (\mathrm{P}^{(y)}_A)^{-1} - \omega_{ICI}(\omega_{ICI}\mathrm{P}^{(y)}_A + (1-\omega_{ICI})\mathrm{P}^{(y)}_B)^{-1} \tag{60}$$

$$(\mathrm{P}^{y}_{B,ICI})^{-1} \triangleq (\mathrm{P}^{y}_B)^{-1} - \omega_{ICI}(\omega_{ICI}\mathrm{P}^{y}_B + (1-\omega_{ICI})\mathrm{P}^{y}_{\mathrm{A}})^{-1} \tag{61}$$

Here, $y = 1, 2, \cdots, m$.

Considering the existence of multiple fusion scenarios in PICI, the results of different sensor combinations can be set as

$$\hat{\mathrm{x}}_{(L)}^{PICI} = \sum_{i=1}^{L} M_i^{PICI} \hat{x}_i \tag{62}$$

$$P_{(L)}^{PICI} = \sum_{i=1}^{L} N_i^{PICI} P_i \tag{63}$$

The weighting matrix is represented by

$$M_i^{PICI} = M_{i(L)}^{PICI} \cdots M_{i(2)}^{PICI} M_{i(1)}^{PICI} \tag{64}$$

$$N_i^{PICI} = N_{i(L)}^{PICI} \cdots N_{i(2)}^{PICI} N_{i(1)}^{PICI} \tag{65}$$

For this purpose, PICI-GM-CPHD can be represented as follows. As shown in Algorithm 1.

Algorithm 1: PICI-GM-CPHD filtering algorithm process

```
Perform GM-DG-CPHD filtering processing:
        Input: $\left\{\omega_{k-1}^i, m_{k-1}^i, P_{k-1}^i\right\}\alpha_{k-1}, \beta_{k-1} Z_k = \{z_1, z_2, \cdots, z_{mk}\}$
Prediction Step
        for $i = 1, 2, \cdots I_{b,k}$ do
        Define the correlation coefficients to predict the intensity of new students' goals in Formula (40),
        and predict the intensity of new students' goals
        $I_{k|k-1} := I_{b,k} + I_{k-1}$
        for $i = I_{b,k} + 1, \cdots, I_{k|k-1}$ do
        Redefine the correlation coefficient to predict the strength of new targets in Formula (40)
        and predict the strength of surviving targets
        Predict cardinality parameters based on Formulas (34), (35), (41), (42), (44), (45)
        End for
Measurement Update Step
        For $i = 1, 2, \cdots I_{k|k-1}$ do
        According to Formulas (43)–(48), update the intensity of newly detected targets and the
        prediction base parameters
        Output:$\left\{\omega_k^i, m_k^i, P_k^i\right\}\alpha_k, \beta_k$
        End for
For $l = 1 : L$($L$ is number of sensors);
        Calculate PICI fusion weight using covariance intersection;
        Replace the covariance of the probability density a of $\mathrm{P}_{\mathrm{A}}^y, \mathrm{P}_{\mathrm{B}}^y$ single sensor with
        $\begin{cases} \mathrm{P}_{A,ICI}^y = \mathrm{P}_A^y + \frac{\omega_{ICI}}{1-\omega_{ICI}} \mathrm{P}_A^y (\mathrm{P}_B^y)^{-1} \mathrm{P}_A^y \\ \mathrm{P}_{B,ICI}^y = \mathrm{P}_B^y + \frac{\omega_{ICI}}{1-\omega_{ICI}} \mathrm{P}_B^y (\mathrm{P}_A^y)^{-1} \mathrm{P}_B^y \end{cases}$
        Different calculations $P_{L_P,k|k}^{A(m)}, P_{L_P,k|k}^{B(m)}$;
        Calculate the next level's fusion result based on the previous level's fusion result;
        Determine the PICI fusion results of multiple sensors based on Equations (64) and (65);
        Calculate different PICI-GM-CPHD weighting matrices
End for
Estimate extraction.
```

4. Modeling and Simulation

To verify the effectiveness of the algorithm, this study applies it to the constant velocity (CV) model for validation. The linear model CV model is $CV(\mathrm{x}) = (x\ \vartheta\ y\ v)^{\top}$, where the four state variables are x as the horizontal axis, ϑ as the angle with the x-axis (counterclockwise as positive), y as the vertical axis, and v as the linear velocity.

$$F_k = \begin{bmatrix} I_2 & \Delta I_2 \\ O_2 & I_2 \end{bmatrix} \tag{66}$$

$$Q_K = \sigma_v^2 \begin{bmatrix} \frac{\Delta^4}{4} I_2 & \frac{\Delta^3}{2} I_2 \\ \frac{\Delta^3}{2} I_2 & \Delta^2 I_2 \end{bmatrix} \tag{67}$$

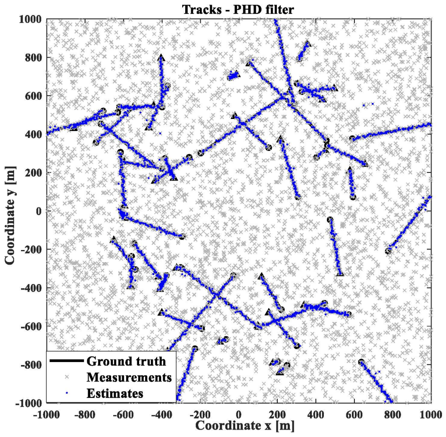
$$H_k = [I_2 \quad O_2],\ R_k = \sigma_\varepsilon^2 I_2 \tag{68}$$

We configure the GM-DG-CPHD filter for a sensor with a detection probability of 0.9 and a survival probability of 0.99. The uniform clutter of the moving target has a Poisson average rate of 5, and the birth density is located at (±1000 m, ±1000 m). All simulations are conducted using 120 Monte Carlo experiments, with the GM-DG-CPHD parameters set as follows: truncation threshold $T = 10^{-5}$, merging threshold $U = 2$, and maximum allowable number of Gaussian terms $J_{\max} = 100$. The tracking performance of the moving targets is evaluated using the optimal sub-pattern assignment (OSPA) distance.

4.1. Comparison Results of Multiple Filtering Methods

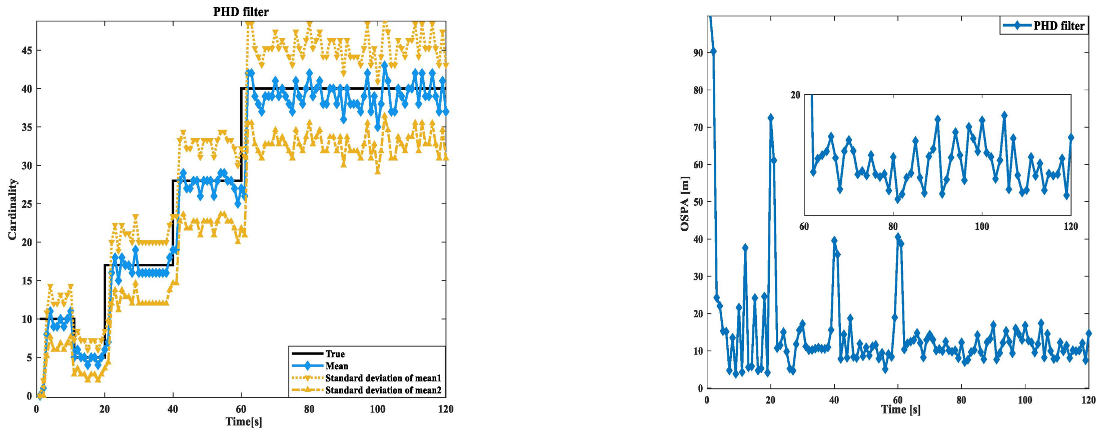
Firstly, the effects of the PHD, CPHD, and DG-CPHD filtering methods are compared, as shown in Figures 5–7. At the same time, in order to avoid the potential impact of moving target motion trajectories on the detection results, this study uses randomly generated target motion trajectories to verify the tracking of multiple target motion results using different filtering methods. We conduct an investigation and analysis in terms of the following aspects. From the results of the three filtering methods shown in Figures 5–7, it can be seen that the DG-CPHD filtering effect is similar to that of CPHD filtering. However, it is significantly higher than the PHD filtering effect, which means that in terms of the filtering effect, $CPHD \approx DGCPHD > PHD$.

Next, the running times of the PHD, CPHD, and DG-CPHD filtering methods are compared, and 40 algorithm runs are selected for comparison, as shown in Figure 8. The above comparison results suggest that the DG-CPHD filtering effect is similar to the CPHD filtering effect, but the calculation time can be significantly shorter than the CPHD filtering operation time. It can be seen that the DG-CPHD filter can be widely used in systems with high requirements for the calculation time and low requirements for the filtering effect. From the running time of the three filtering methods shown in Figure 8, CPHD has the longest filtering time, PHD has the shortest filtering time, and DG-CPHD has a running time between the two. In other words, from the perspective of the filtering effect, $t_{CPHD} > t_{DGCPHD} > t_{PHD}$.



(**a**)

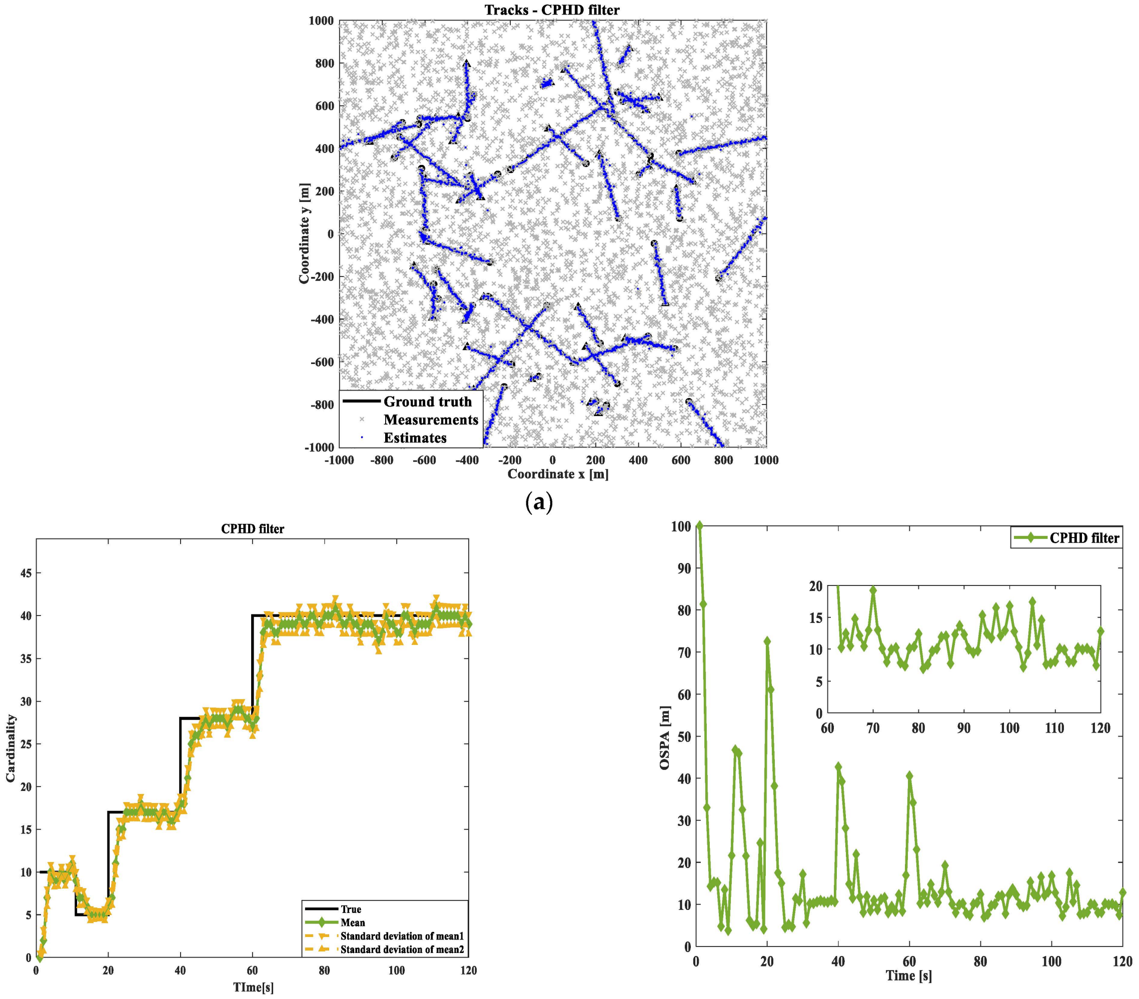
Figure 5. *Cont.*



(**b**)

(**c**)

Figure 5. Comparison of PHD filtering results: (**a**) target tracking path result; (**b**) cardinal distribution results; (**c**) OSPA error results.

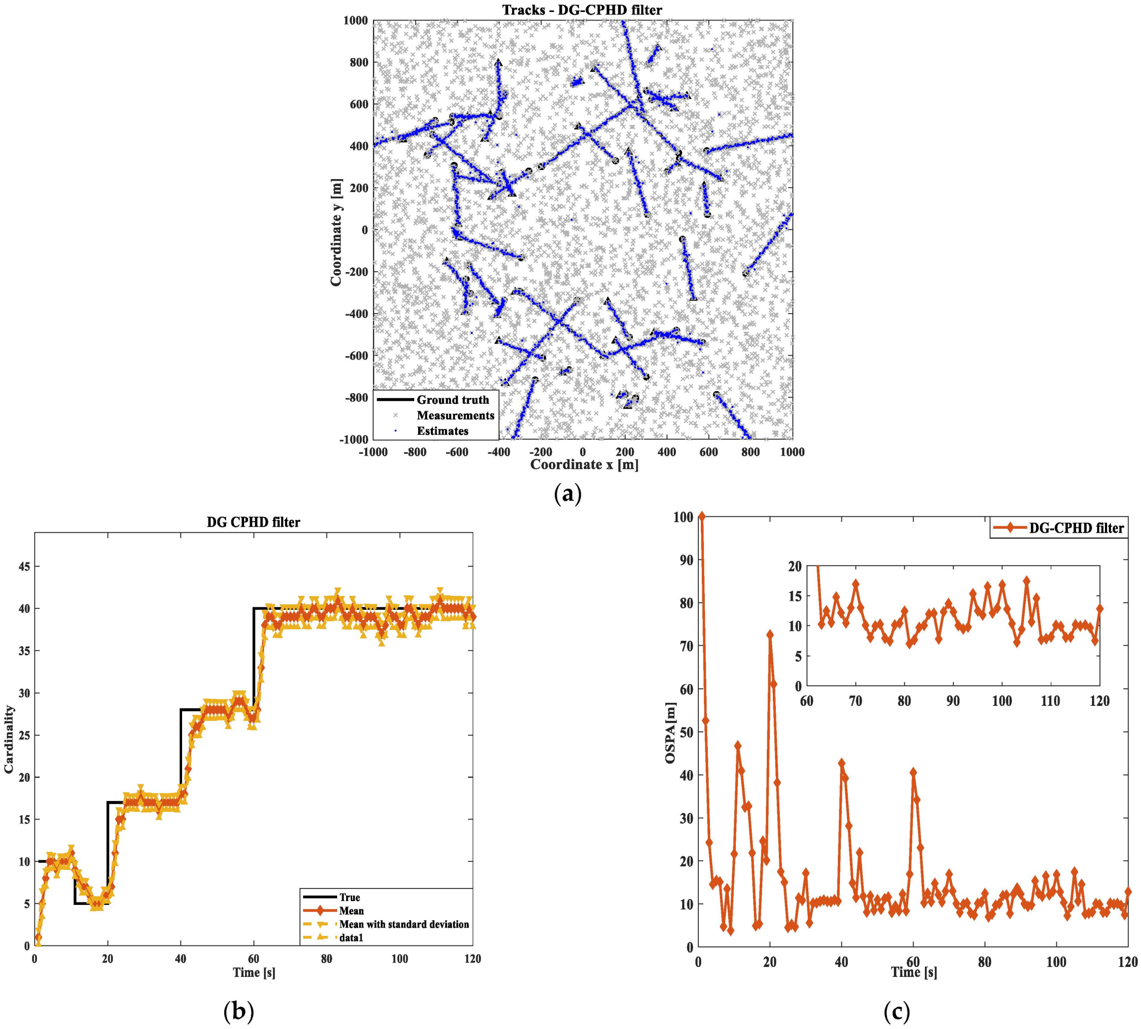


(**a**)

(**b**)

(**c**)

Figure 6. Comparison of CPHD filtering results: (**a**) target tracking path result; (**b**) cardinal distribution results; (**c**) OSPA error results.



(**a**)

(**b**)

(**c**)

Figure 7. Comparison of GM-CPHD filtering results: (**a**) target tracking path result; (**b**) cardinal distribution results; (**c**) OSPA error results.

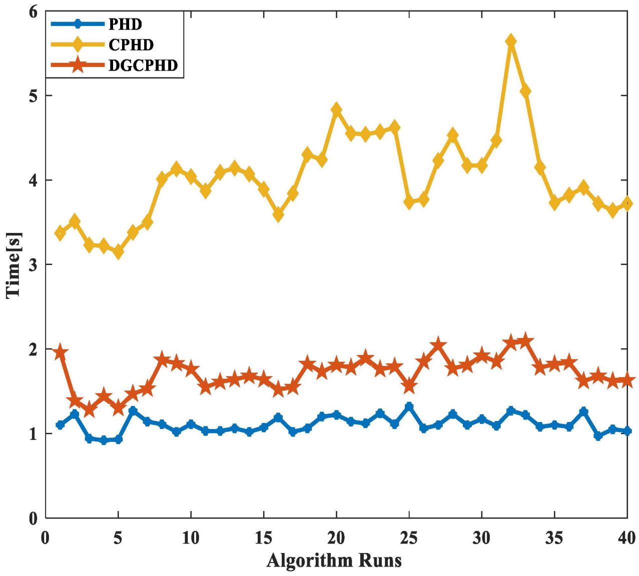


Figure 8. Comparison of running time results of three filtering methods.

The running time results when tracking different numbers of moving targets using the three filtering methods are compared, as shown in Figure 9. From the results, it can be seen that CPHD and GM-CPHD filtering increase the running time significantly with an increasing number of runs; PHD and GM-PHD filtering have a smaller increase in running time as the number of runs increases. The increase in the operating time of DG-CPHD is between the two. Thus, from the perspective of the effect of different runs over time, $t'_{CPHD} > t'_{DGCPHD} > t'_{PHD}$. In order to further verify the effectiveness of different filtering methods with different numbers of runs over the running time, the ratio of the running time to the number of runs was calculated for the three filtering methods. From the results shown in Figure 10, it can be seen that the ratio of the running time of CPHD and GM-CPHD filtering to the tracking of different numbers of moving targets has a relatively large effect. The ratio of the PHD and GM-PHD filtering operation times to the operation frequency results is relatively small. The comparison of the ratio of the DG-CPHD filter's running time to the number of runs is between the two. Thus, from the comparison results of the ratio of the filtering running time to the number of runs, it can be seen that $t''_{CPHD} > t''_{DGCPHD} > t''_{PHD}$.

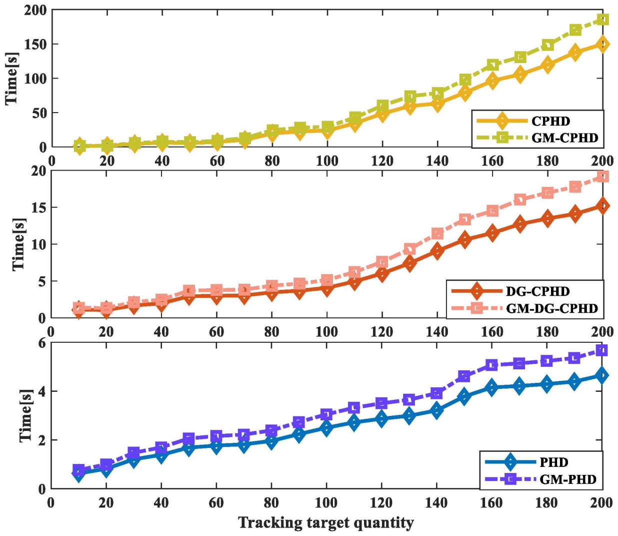


Figure 9. Three filtering methods used to track the running time results with different numbers of moving targets.

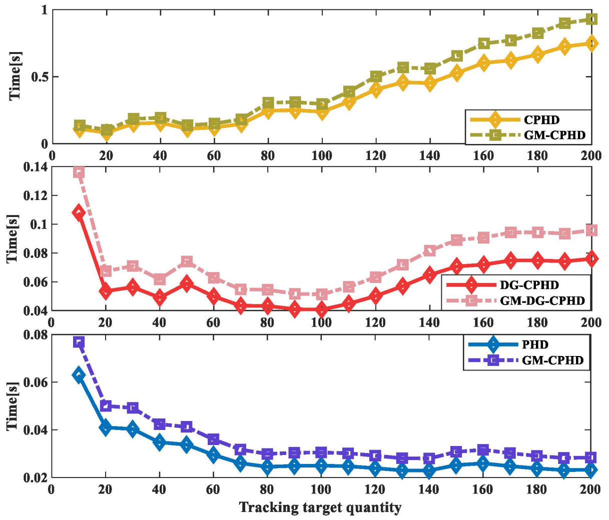


Figure 10. Results of the ratio of running time of three filtering methods to different numbers of moving targets.

From the above results, it can be seen that the effectiveness of DG-CPHD filtering and GM-DG-CPHD filtering is between that of PHD and CPHD filtering, but the running time can be between that of the other methods and is less affected by the number of runs. In other words, DG-CPHD filtering is suitable for systems that do not require high filtering performance but require a good operating time. For this purpose, DG-CPHD filtering and GM-DG-CPHD filtering are selected for research and analysis in this study.

4.2. Comparison Results of Multiple Fusion Methods

For comparative analysis, 20 linear Gaussian measurement models are selected. The initial state of the linear Gaussian measurement model target is shown in Table 1.

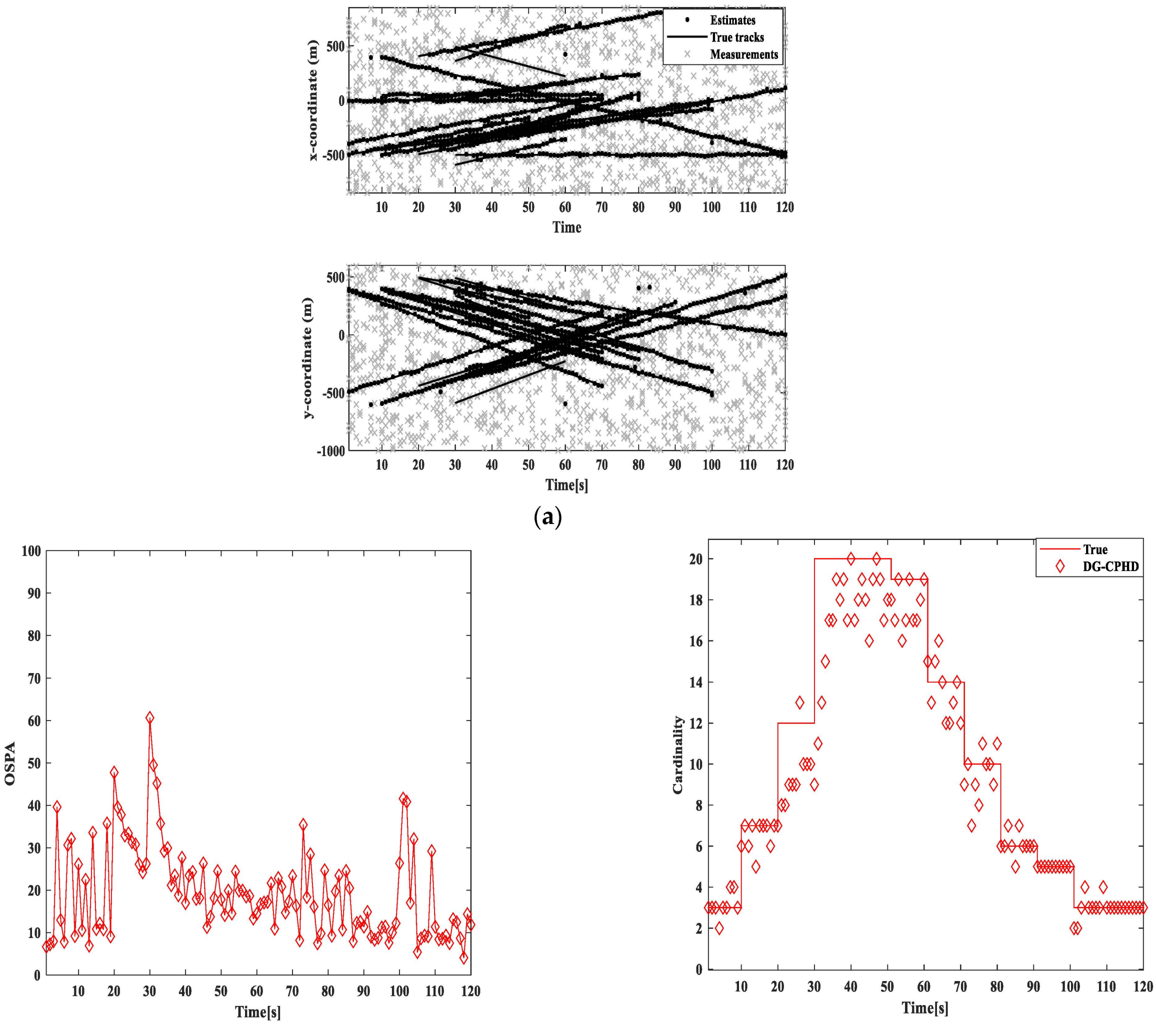
Table 1. Twenty linear Gaussian measurement models for the motion state of moving targets.

Target	Initial State	Appearing Frame	Disappearing Frame
1	[0; 0; 400; −12]	1	70
2	[−500; 5; 400; −9]	1	100
3	[−400; 6; −500; 10]	1	70
4	[400; −8; −600; 10]	10	120
5	[−500; 5; 400; −9]	10	70
6	[50; 0; 400; −10]	10	70
7	[−450; 7; 400; −6]	10	50
8	[−450; 7; −450; 10]	20	60
9	[400; 6; −500; 11]	20	90
10	[−500; 6; 500; −5]	20	120
11	[0; 4; 350; −8]	20	80
12	[−400; 4; 500; −10]	20	100
13	[−400; 9; −400; 12]	30	80
14	[350; 11; −400; 8]	30	60
15	[−600; 8; 500; −9]	30	60
16	[0; 5; 350; −12]	30	60
17	[−400; 5; 300; −10]	30	80
18	[−500; 0; −400; 8]	30	120
19	[500; −9; −600; 12]	30	60
20	[−450; 10; 400; −6]	30	80

Firstly, two algorithms, GM-DG-CPHD and PICI-GM-DG-CPHD, are implemented, as shown in Figures 11 and 12. From the implementation effects of GM-DG-CPHD and PICI-GM-DG-CPHD in Figures 11 and 12, it can be seen that the two algorithms can effectively achieve multi-objective motion tracking filtering for linear systems.

Secondly, the fusion effects of several methods, namely GM-DG-CPHD (single sensor 1), GICI-GM-DG-CPHD, SICI-GM-DG-CPHD, BICI-GM-DG-CPHD, and PICI-GM-DG-CPHD, are compared, as shown in the figures. From the 12 comparison results, it can be seen that the PICI-GM-DG-CPHD effect is second only to the GICI-GM-DG-CPHD effect. Compared with the SICI-GM-DG-CPHD and BICI-GM-DG-CPHD methods, it is more similar to the effect of GM-DG-CPHD (single sensor 1).

Finally, after verifying the comparison results of several filters' effects, it is also necessary to verify the method running times of the fusion algorithms, as shown in Figure 13. We compare the running times of the GM-DG-CPHD (single sensor 1), GICI-GM-DG-CPHD, SICI-GM-DG-CPHD, BICI-GM-DG-CPHD, and PICI-GM-DG-CPHD methods, and we randomly select 40 algorithm runs for comparison, as shown in Figure 14. The above comparison results indicate that the filtering time of PICI-GM-DG-CPHD is between that of SICI-GM-DG-CPHD and BICI-GM-DG-CPHD, which is suitable for the efficient implementation of distributed sensor fusion.

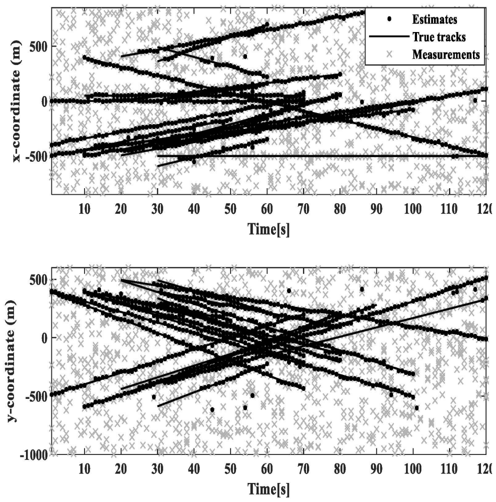


(**a**)

(**b**)

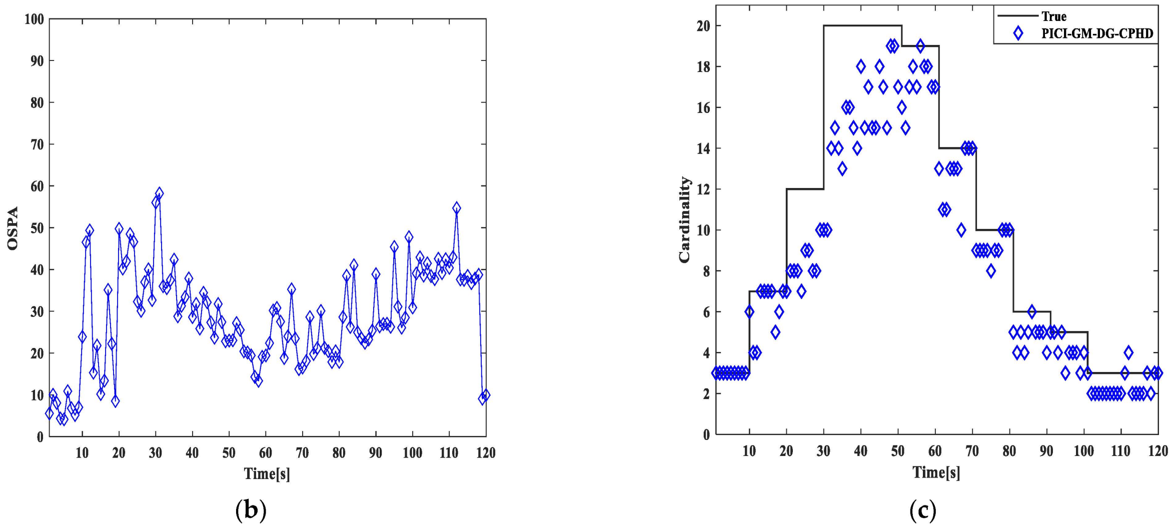
(**c**)

Figure 11. Comparison of GM-DG-CPHD filtering results: (**a**) target tracking path result; (**b**) cardinal distribution results; (**c**) OSPA error results.



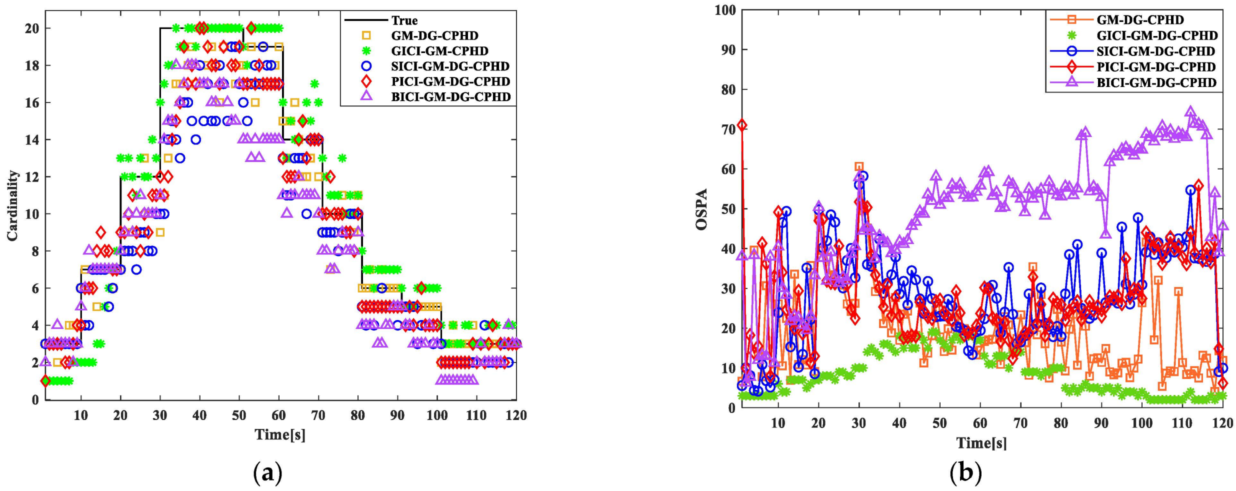
(**a**)

Figure 12. *Cont.*



(**b**) (**c**)

Figure 12. Comparison of PICI-GM-DG-CPHD filtering results: (**a**) target tracking path result; (**b**) cardinal distribution results; (**c**) OSPA error results.



(**a**) (**b**)

Figure 13. Comparison results of several filters: (**a**) cardinal distribution results; (**b**) OSPA error results.

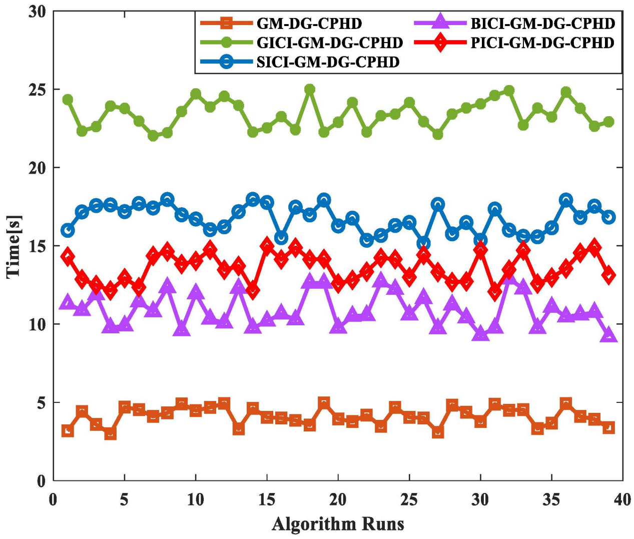


Figure 14. Comparison results of running times of several filters.

4.3. Algorithm Complexity Analysis

Based on the research conclusion that the complexity of GM-DG-CPHD increases significantly with the increase in the number of targets, it can be seen that that of the PICI-GM-DG-CPHD algorithm also increases significantly with the increase in the number of targets, and it will be far greater than the recursive complexity $\mathcal{O}(m^3) \sim \mathcal{O}|Z|^3 n_{\max})$ of CPHD. However, the running time of PICI-GM-DG-CPHD can be between that of SICI-GM-DG-CPHD and BICI-GM-DG-CPHD, which, to some extent, indicates that the computational complexity is between the two mentioned above.

5. Conclusions

This study proposes a GM-DG-CPHD filter based on PICI for the efficient implementation of distributed fusion in sensor networks and verifies the simulation performance through a linear Gaussian measurement model. Comparing PHD, GM-PHD, CPHD, GM-CPHD, DG-CPHD, and GM-DG-CPHD, the simulation results show that the DG-CPHD filtering effect is similar to the CPHD filtering effect, but the calculation time can be significantly shorter than the CPHD filtering running time, making it suitable for the efficient implementation of linear system target tracking. Comparing GM-DG-CPHD (single sensor 1), GICI-GM-DG-CPHD, SICI-GM-DG-CPHD, BICI-GM-DG-CPHD, PICI-GM-DG-CPHD, and PICI-GM-DG-CPHD, the simulation results show that PICI-GM-DG-CPHD can substantially reduce the computing time, and its performance is second only to that of GICI-GM-DG-CPHD, making it more suitable for distributed fusion with a large number of sensors.

Author Contributions: Conceptualization, L.W.; methodology, G.C.; validation, L.W.; data curation, L.W.; writing—original draft preparation, L.W.; writing—review and editing, G.C.; visualization, G.C.; project administration, L.W.; funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: Special Industrial Technology Research Project of Jilin Province, Research on Self-Organizing Network System of Unmanned Platform for Optoelectronic Composite Communication, No. 2022C047-8; Supported by "Thirteenth Five-Year Plan" of Provincial Science and Technology of Education Department of Jilin Province, Research on Large-Scale D2D Access, and Traffic Balancing Technology for Heterogeneous Wireless Networks (JJKH20181130KJ).

Institutional Review Board Statement: The studies did not involve humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank all of the authors cited in this article and the anonymous reviewers for their helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, B.; Li, S.; Battistelli, G.; Chisci, L.; Yi, W. Multi-Agent Fusion with Different Limited Fields-of-View. *IEEE Trans. Signal Process.* **2022**, *70*, 1560–1575. [CrossRef]
2. Marsh, B.; Sadka, A.H.; Bahai, H. A Critical Review of Deep Learning-Based Multi-Sensor Fusion Techniques. *Sensors* **2022**, *22*, 9364. [CrossRef] [PubMed]
3. McAfee, M.; Kariminejad, M.; Weinert, A.; Huq, S.; Stigter, J.D.; Tormey, D. State Estimators in Soft Sensing and Sensor Fusion for Sustainable Manufacturing. *Sustainability* **2022**, *14*, 3635. [CrossRef]
4. Jin, X.-B.; Yu, X.-H.; Su, T.-L.; Yang, D.-N.; Bai, Y.-T.; Kong, J.-L.; Wang, L. Distributed Deep Fusion Predictor for a Multi-Sensor System Based on Causality Entropy. *Entropy* **2021**, *23*, 219. [CrossRef] [PubMed]
5. Fantacci, C.; Vo, B.-T.; Battistelli, G.; Chisci, L. Robust Fusion for Multisensor Multiobject Tracking. *IEEE Signal Process. Lett.* **2018**, *25*, 640–644. [CrossRef]
6. Wang, P.; Ji, H.; Liu, L. Consistent fusion method with uncertainty elimination for distributed multi-sensor systems. *Inf. Sci. Int. J.* **2022**, *595*, 378–394. [CrossRef]

7. Chai, L.; Yi, W.; Kong, L. Joint sensor registration and multi-object tracking with PHD filter in distributed multi-sensor networks. *Signal Process.* **2023**, *206*, 108909. [CrossRef]
8. Yu, H.; Dai, K.; Li, Q.; Li, H.; Zhang, H. Optimal Distributed Finite-Time Fusion Method for Multi-Sensor Networks under Dynamic Communication Weight. *Sensors* **2023**, *23*, 7397. [CrossRef]
9. Senel, N.; Kefferpütz, K.; Doycheva, K.; Elger, G. Multi-Sensor Data Fusion for Real-Time Multi-Object Tracking. *Processes* **2023**, *11*, 501. [CrossRef]
10. Liu, P.; Zhou, S.; Zhang, P.; Li, M. Distributed State Fusion Estimation of Multi-Source Localization Nonlinear Systems. *Sensors* **2023**, *23*, 698. [CrossRef]
11. Su, L.; Yu, J.; Hua, Y.; Li, Q.; Dong, X.; Ren, Z. Multi-target Distributed Tracking with Probability Hypothesis Density in Large-scale Sensor Networks. *IEEE Sens. J.* **2023**, *24*, 31061–31071. [CrossRef]
12. Pal, Y.; Mohan, R.; Rao, G.R.K.; Rastogi, R.; Shah, M.A. Underwater Wireless Sensor Networks Using Distributed Fusion of Optimally Quantized Estimations from Local Trackers. In Proceedings of the 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 19–20 June 2023; pp. 916–921. [CrossRef]
13. Li, C.; Yang, Q.; Cao, Y.; Rao, H.; Xu, Y. Sequential Covariance Intersection Based on the Upper Bound of Correlation Coefficient. In Proceedings of the 2022 13th Asian Control Conference (ASCC), Jeju, Republic of Korea, 4–7 May 2022; pp. 1704–1709. [CrossRef]
14. Noack, B.; Sijs, J.; Hanebeck, U.D. Inverse Covariance Intersection: New Insights and Properties. In Proceedings of the 2017 20th International Conference on Information Fusion (Fusion), Xi'an, China, 10–13 July 2017.
15. Chen, H.; Wang, J.; Wang, C.; Shan, J.; Xin, M. Distributed diffusion unscented Kalman filtering based on covariance intersection with intermittent measurements. *Automatica* **2021**, *132*, 109769. [CrossRef]
16. Kim, S.Y.; Kang, C.H.; Song, J.W. 1-point RANSAC UKF with Inverse Covariance Intersection for Fault Tolerance. *Sensors* **2020**, *20*, 353. [CrossRef] [PubMed]
17. Park, W.J.; Park, C.G. Multi-target Tracking Based on Gaussian Mixture Labeled Multi-Bernoulli Filter with Adaptive Gating. In Proceedings of the 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 16–18 January 2019; pp. 226–229. [CrossRef]
18. Li, Y.; Wang, B. Multi-Extended Target Tracking Algorithm Based on VBEM-CPHD. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2250026. [CrossRef]
19. Sun, T.; Xin, M. Inverse-Covariance-Intersection-Based Distributed Estimation and Application in Wireless Sensor Network. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10079–10090. [CrossRef]
20. Liu, Q.; Shang, T.; Chen, L.; Yu, K.; Gao, Y.; Huo, Y.; Doul, Y. Sequential Inverse Covariance Intersection Fusion Kalman Filter for Multi-sensor Systems with Packet Dropouts. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 3543–3548. [CrossRef]
21. Sun, T.; Xin, M. Distributed Estimation with Iterative Inverse Covariance Intersection. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 1645–1649. [CrossRef]
22. Hu, Z.; Chen, B.; Zhang, A.W.; Yu, L. Enhanced Hierarchical and Sequential Covariance Intersection Fusion. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7888–7893. [CrossRef]
23. Wang, B.; Yi, W.; Hoseinnezhad, R.; Li, S.; Kong, L.; Yang, X. Distributed Fusion with Multi-Bernoulli Filter Based on Generalized Covariance Intersection. *IEEE Trans. Signal Process.* **2017**, *65*, 242–255. [CrossRef]
24. Wang, L.; Chen, G.; Chen, G. Gaussian Mixture Cardinalized Probability Hypothesis Density (GM-CPHD): A Distributed Filter Based on the Intersection of Parallel Inverse Covariances. *Sensors* **2023**, *23*, 2921. [CrossRef]
25. Park, W.J.; Park, C.G. Distributed GM-CPHD Filter Based on Generalized Inverse Covariance Intersection. *IEEE Access* **2021**, *9*, 94078–94086. [CrossRef]
26. Li, G.; Yi, W.; Kong, L. A real-time fusion algorithm for asynchronous sensor system with PHD filter. In Proceedings of the 2017 International Conference on Control, Automation and Information Sciences (ICCAIS), Chiang Mai, Thailand, 31 October–1 November 2017; pp. 134–139. [CrossRef]
27. Li, G.; Yi, W.; Kong, L. A fast implementation of distributed fusion with CPHD filter. In Proceedings of the 2017 International Conference on Control, Automation and Information Sciences (ICCAIS), Chiang Mai, Thailand, 31 October–1 November 2017; pp. 123–127. [CrossRef]
28. Li, G.; Battistelli, G.; Chisci, L.; Gao, L. Distributed Multi-Agent Joint Time Alignment and State Estimation. In Proceedings of the 2020 European Control Conference (ECC), St. Petersburg, Russia, 12–15 May 2020; pp. 923–928. [CrossRef]
29. Wei, S.; Zhang, B.; Yi, W. Trajectory PHD and CPHD Filters with Unknown Detection Profile. *IEEE Trans. Veh. Technol.* **2022**, *71*, 8042–8058. [CrossRef]
30. Li, G.; Li, G.; He, Y. Distributed GGIW-CPHD-Based Extended Target Tracking Over a Sensor Network. *IEEE Signal Process. Lett.* **2022**, *29*, 842–846. [CrossRef]
31. Yu, B.; Li, T.; Ge, S.; Gu, H. Robust CPHD Fusion for Distributed Multitarget Tracking Using Asynchronous Sensors. *IEEE Sens. J.* **2022**, *22*, 1030–1040. [CrossRef]
32. De Melo, F.E.; Maskell, S. A CPHD Approximation Based on a Discrete-Gamma Cardinality Model. *IEEE Trans. Signal Process.* **2019**, *67*, 336–350. [CrossRef]

33. Campbell, M.A.; Clark, D.E.; de Melo, F. An Algorithm for Large-Scale Multitarget Tracking and Parameter Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2053–2066. [CrossRef]
34. Balthasar, M.R.; Zoubir, A.M. Multi-target tracking in distributed sensor networks using particle PHD filters. *Signal Process* **2019**, *159*, 130–146. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.