A Novel Computationally Efficient SMC-PHD Filter Using Particle-measurement Partition

Rui Sun¹, Lingling Zhao², Xiaohong Su³*
¹, ², ³ School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
¹ sunr001@163.com, ² zhaoll@hit.edu.cn, ³ sxh@hit.edu.cn

Abstract—The probability hypothesis density (PHD) filter is widely used to solve multi-target tracking (MTT) problems. Although the Sequential Monte Carlo (SMC) implementation provides a tractable solution for PHD filter to handle the highly nonlinear and non-Gaussian MTT scenario, the high computational cost caused by a large number of particles limits the applications that need to be performed in real-time. This paper proposes a computationally efficient SMC-PHD filter using particle-measurement partition and intermediate region strategy. Firstly, the partition strategy provides a way to solve the related PHD calculation in each partition independently. Secondly, based on the rectangular gating technique, the particle intermediate region strategy ensures the estimation accuracy of the proposed method. The simulation results indicate that the partition strategy significantly reduces the computational complexity of the SMC-PHD filter. In addition, the proposed method can maintain comparable accuracy as the standard SMC-PHD filter via the intermediate region strategy.

Keywords—Multi-target tracking; PHD filter; Computationally efficient SMC-PHD filter;

I. INTRODUCTION

Multiple target tracking (MTT) [1] has attracted increasing attention of researchers in different fields owing to its great widely applications [2],[3],[4],[5]. The traditional method for MTT problem is to assign the measurement to the target by the data association techniques. Some widely-used classical association-based MTT formulations include Global Nearest Neighbor (GNN) [6], Multiple Hypothesis Tracking (MHT) [7] and Joint Probabilistic Data Association Filter (JPDAF) [8]. However, the association based formulation is numerically intractable, as the computational complexity grows exponentially with target number. The PHD filter [9], which avoids explicit data associations, is proposed as a tractable alternative for the MTT problem. To alleviate the intractability of the multiple target Bayesian filter, the PHD filter propagates the first-order statistical moment of the random variable which describes the multitarget state [10] instead of the multi-target Bayesian posterior. However, it still requires to deal with the multiple integrals with no close-form solution [1]. To solve the problem, the PHD filter has been implemented by Gaussian mixture (GM-PHD) [11] or via Sequential Monte Carlo (SMC-PHD or Particle PHD) [12]. The SMC-PHD filter has the ability to handle highly nonlinear and non-Gaussian situations where the Gaussian Mixture-based method is not modeling the scenario appropriately. However, the high computational complexity is the main challenge in SMC-PHD filter and limits its application to MTT scenarios that need to be solved in real-time. In recent years, various computationally efficient SMC-PHD algorithms have been presented. S. Hong et al. [13] proposed a simplified particle PHD filter, which can reduce the calculation of the update step by a threshold. However, setting the threshold for different tracking scenarios is the main challenge. Z. Shi et al. [14] proposed a threshold based resampling scheme suitable for the fully-pipelined architecture hardware. As an increasing number of the relatively less important particles are discarded, it may lead to the particle degradation. Zheng et al. [15] proposed a data-driven Particle PHD Filter, which shields the clutters via existing historic state of targets and achieves a better performance than the standard SMC-PHD filter. However, the data-driven mechanism of the proposed method is limited by the two assumptions that the target maneuvering is not too abrupt and the sampling period is not too long. The high-speed sigma-gating SMC-PHD filter [16] uses a rectangular sigma gating to reduce the updating calculation. However, in some cases, the lack of samples diversity will occur, because the randomness of the noise may lead more and more useful particles distribute outside the gating. In order to meet the high-speed requirement for SMC-PHD, we focus on the reduction of the problem size by a partition strategy. Based on the partition strategy, our proposed method is also suitable for a parallel computing implementation.

In this paper, the main contributions are as follows. First, we propose a partition strategy to divide the measurement and particle set into some subsets and provide a way to solve the related PHD calculation in each subset independently. Second, in order to ensure the computational accuracy, we propose the intermediate region strategy based on the rectangular gating technique. Finally, based upon the partition strategy and the intermediate region strategy, we propose a computationally efficient SMC-PHD filter algorithm, which speeds up the execution and makes parallelism of update step possible. The remainder of this paper is organized as follows. The PHD filter formulation is described in Section II. In section III, the proposed computationally efficient SMC-PHD filter is presented. The simulation results are given in Section IV, following by the conclusion in Section V.

II. PROBLEM FORMULATION

A. The PHD filter

In the RFS formulation [1], the multiple target states and measurements at the time step $k$ are naturally represented as random finite sets $X_k = \{x_{k,1}, \ldots, x_{k,M(k)} \in \mathcal{F}(\chi)\}$ and $Z_k = \{z_{k,1}, \ldots, z_{k,N(k)} \in \mathcal{F}(\zeta)\}$, where $\mathcal{F}(\chi)$ and $\mathcal{F}(\zeta)$ are...
power sets of $\chi$ and $\zeta$, $X_k$ is composed of survived targets $S_k(\{k\}) (x_{k-1})$, spawned targets $B_{k|k-1} (x_{k-1})$ and new born targets $\Gamma_k$. $Z_k$ is composed of clutter $K_k$ and target measurements $\theta_k(X_k)$. Based upon the RFS formulation and the Bayesian filter, Mahler proposed the PHD Filter as a tractable solution for the MTT problem. The PHD filter propagates the first-order statistical moment of posterior density [12], known as PHD (or intensity function) denoted by $D_k|k (x|z_{1:k})$ [1], to alleviate the intractability of the Bayesian filter. The integral of the PHD over any region $S$ is equal to the expected number of targets in $S$. The local maxima of $D_k|k$ generate the target state estimation. The PHD prediction step is defined as follows.

$$D_k|k-1 = \gamma_k(x) + \int \Phi_k|k-1 (x, x_{k-1}) D_{k-1|k-1} dx_{k-1} \quad (1)$$

Where $\gamma_k(\cdot)$ denotes the intensity function of the new born target RFS $\Gamma_k$.

$$\Phi_k|k-1 (x, x_{k-1}) = P_s(x_{k-1}) f_{k|k-1} (x|x_{k-1}) + b_{k|k-1} (x|x_{k-1}) \quad (2)$$

$P_S(\zeta)$ denotes the survival probability, $f_{k|k-1}(\cdot|\cdot)$ denotes the state transition probability density, and $b_{k|k-1}(\cdot|\cdot)$ denotes the spawned intensity function. The PHD update step is defined as follows.

$$D_k|k = \left[ \nu(x) + \sum_{\nu \in \mathcal{Z}_{k|k}\{D_k|k-1 w_{k|k}(x)\} \right] D_{k|k-1} \quad (3)$$

Where $\mathcal{Z}_{k|k}(\cdot)$ denotes the intensity function of the clutter RFS $K_k$, $K_k(z) = \lambda_k c_k(z)$. $\lambda_k$ is the expected number of clutter points per scan, clutter point obeys the probability distribution $c_k(\cdot)$. $P_D(\zeta)$ denotes the detection probability of target, $\nu(x) = 1 - P_D(x)$ denotes the lost detection probability. $\psi_{k|k}(x) = P_D(x) g(z|x), g(\cdot|\cdot)$ is the multi-target likelihood.

B. The SMC-PHD filter

In order to solve the multiple integral problem, the numerical integration method of which may suffer from the curse of dimensionality, the particle filter (or SMC) implementation for the PHD filter is proposed [12]. The SMC-PHD filter algorithm is presented as follows.

1) Prediction step: At time step $k \geq 1$, get particle set $\{w_k^{(i)}(i), x_{k-1}^{(i)}\}_{i=1}^{L_k}$, where $w_k^{(i)}$ denotes the particle weight and $x_{k-1}^{(i)}$ denotes the particle state. Sample $\tilde{x}_k^{(i)} \sim q_k(\cdot| x_{k-1}, Z_k), i = 1, ..., L_k$ and predict the weights of $\{x_k^{(i)}\}_{i=1}^{L_k}$. 

$$\tilde{w}_{k|k-1}^{(i)} = \frac{\nu_k(x_k^{(i)})}{f_k p_k(x_k^{(i)}|z_k)}, i = L_k-1 + 1, ..., L_k-1 + J_k \quad (5)$$

2) Update step: Update the particle weight,

$$w_k^{(i)} = \left[ \tilde{w}_k^{(i)} + \sum_{\nu \in \mathcal{Z}_{k|k}} \psi_{k|z}(\tilde{x}_k^{(i)}) \right] \tilde{w}_{k|k-1}^{(i)}, i = 1, ..., L_k-1 + J_k \quad (6)$$

Where $C_k(z) = \sum_{j=1}^{L_k} P(z_k|\nu_j) \psi_{k|z}(\tilde{x}_k^{(j)}) \tilde{w}_{k|k-1}^{(j)}$, and $z \in Z_k$.

3) Particle resampling: Compute the sum of the total particle weight $\bar{N}_{k|k} = \sum_{j=1}^{L_k} w_k^{(j)}$, $T_k = \text{round}(\bar{N}_{k|k})$ gives the estimated target number. Set $L_k = T_k \times \rho$ after resampling, $\rho$ denotes the particle number per target. Using the resampling algorithm to resample the particles $\{w_k^{(i)}(i)/\bar{N}_{k|k}, x_k^{(i)}\}_{i=1}^{L_k}$, we get the resampled particle set $\{w_k^{(i)}(i), x_k^{(i)}\}_{i=1}^{L_k}$ with normalized weight $\tilde{w}_{k|k} = L_k^{-1}$.

From the recursion of the SMC-PHD filter, it can be drawn that the computational complexity grows linearly with the measurement number and the particle number.

III. THE COMPUTATIONALLY EFFICIENT SMC-PHD FILTER USING PARTICLE-MEASUREMENT PARTITION

A. Partition of measurement and particle-measurement set

We propose a novel computationally efficient SMC-PHD filter based on the partition of measurement and particle set. The partition strategy is motivated by the fact that the far the metric distance between a measurement and a particle is, the less likelihood contribution the particle makes. During the likelihood calculation, the likelihood of those particles far from the measurement is close to zero. Therefore, we discard the likelihood calculation of those particles whose projections onto the measurement space far from the measurement to reduce the computational cost of SMC-PHD filter. The grid partition strategy is presented in Table 1.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 1: Grid partition algorithm</td>
</tr>
<tr>
<td>Input: The measurement set $Z_k = {z_{ik}}, n = 1, ..., N(k)$, the particle set projected onto the measurement space $P_k = {p_{ik}}, m = 1, ..., L_k$, the measurement region of the sensor $[M_{min}, M_{max}] \times [M_{ymin}, M_{ymax}]$, the number of the partition $N$ and the threshold vector $T_k = {t_{ik}}, i = 1, ..., 2M$ trained from the measurement set, where $M$ denotes the number of the partition points.</td>
</tr>
<tr>
<td>Output: ${x_{ik}}, i = 1, ..., N, {p_{ik}}, i = 1, ..., N$.</td>
</tr>
<tr>
<td>1: $A_k = {a_{ik}}, i = 1, ..., 2M + 4$. Set $a_{ik} = M_{ymin}, a_{2k} = M_{ymin}, a_{2k+2} = M_{ymax}, a_{2k+4} = M_{ymin}, a_{2k+6} = M_{ymax}$, let $a_{jk} = 3, j = 3, ..., 2M + 2$.</td>
</tr>
<tr>
<td>2: Set $X_{\text{boundary}} = 0, Y_{\text{boundary}} = 0, X_{\text{num}} = 0, Y_{\text{num}} = 0, Z_{ik} = 0, P_{ik} = 0, i = 1, ..., N$.</td>
</tr>
<tr>
<td>3: for $m = 0$ to $M + 1$ do</td>
</tr>
<tr>
<td>4: if $a_{2m+1} \neq$ is not NULL and not in $X_{\text{boundary}}$ then if $a_{2m+1} \neq$ is not NULL and not in $X_{\text{boundary}}$ then put $a_{2m+1} + k$ in $X_{\text{boundary}}, X_{\text{num}} = X_{\text{num}} + 1$.</td>
</tr>
<tr>
<td>5: if $a_{2m+2} \neq$ is not NULL and not in $Y_{\text{boundary}}$ then if $a_{2m+2} \neq$ is not NULL and not in $Y_{\text{boundary}}$ then put $a_{2m+2} + k$ in $Y_{\text{boundary}}, Y_{\text{num}} = Y_{\text{num}} + 1$.</td>
</tr>
<tr>
<td>6: end for</td>
</tr>
<tr>
<td>7: Sort the elements of $X_{\text{boundary}} {x_{ik}}, i = 1, ..., X_{\text{num}}$ in ascending order.</td>
</tr>
</tbody>
</table>
order, and sort the elements of $Y_{\text{boundary}} \{y_{i,\text{bound}}, i = 1, ..., Y_{\text{num}} \}$ in ascending order, where $(X_{\text{num}} - 1) \ast (Y_{\text{num}} - 1) = N$.

8: for $i = 1$ to $Y_{\text{num}} - 1$ do
9: for $j = 1$ to $X_{\text{num}} - 1$ do
10: Assign the identity $I_{\text{par}} = (X_{\text{num}} - 1) \ast (i - 1) + j$ for the partition space $[y_{\text{bound}}, y_{1,\text{bound}}] \times [x_{\text{bound}}, x_{1,\text{bound}}]$.
11: for each $\{x_{\text{par}}, y_{\text{par}}, x_{\text{bound}}\}$ do
12: if $x_{\text{par}}$ in the partition $I_{\text{par}}$ then put $x_{\text{par}}$ in $Z_{\text{par}, k}$.
13: end for
14: for each $\{x_{\text{par}}, y_{\text{par}}, x_{\text{bound}}\}$ do
15: if $x_{\text{par}}$ in the partition $I_{\text{par}}$ then put $x_{\text{par}}$ in $P_{\text{par}, k}$.
16: end for
17: end for
18: end for

To illustrate the grid partition strategy, a four-partition example with our partition method is given in Fig. 1. Where the vector $A_k = \{M_{x_{\text{min}}}, M_{y_{\text{min}}}, t_{1, k}, t_{2, k}, M_{x_{\text{max}}}, M_{y_{\text{max}}}\}$, $t_{1, k}$ and $t_{2, k}$ are the mean values of the measurement set in different dimensions at time step $k$. Based on the grid partition strategy, we can update the weight of particles only using the measurements in the same partition pace, and the calculation of each partition is independent of each other.

### B. The intermediate region

The partition strategy reduces the likelihood calculation cost, it also raises a problem that the particles close to the measurements may be divided into different partitions, especially when some particles are located near the partition boundary (encircled by the blue dashed ellipse in Fig. 1). Consequently, the discarding of the nearby particles outside of the partition will give rise to the loss of the approximation accuracy. To alleviate the aforementioned problem, we propose the intermediate region strategy. We define the intermediate region by a difference set operator. Consider a partition region $R_{\text{part}} = [x_{\text{min}}, x_{\text{max}}] \times [y_{\text{min}}, y_{\text{max}}]$, the intermediate region of the partition is $R_{\text{inter}}$, and the region contain the partition region and its intermediate region is $R_{\text{union}}$.

\[
R_{\text{inter}} = R_{\text{union}} - R_{\text{part}}
\]

\[
R_{\text{union}} = [r_{\text{min}}, r_{\text{max}}] \times [r_{\text{min}}, r_{\text{max}}] \times [r_{\text{min}}, r_{\text{max}}]
\]

\[
r_{\text{min}} = \max(x_{\text{min}} \ast g_x, M_{x_{\text{min}}})
\]

Where $g_x$ and $g_y$ denote the gating size of different dimensions. The particles of the intermediate region $R_{\text{inter}}$ are within a gating size distance of the partition boundary. It is noted that the gating size selection is verified through adequate experimental in [16]. Typically, $P_{\text{Diss}3\sigma} = 99.73\%$ and $P_{\text{Diss}5\sigma} = 99.999994\%$ are the representative values. where $\text{Dis}$ denotes the distance between the measurement and the estimated value and $\sigma$ is the standard deviation of the measurement noise in each dimension. In our method, $3\sigma$ value is selected as the distance threshold. The grid partition and intermediate region strategy is presented in Table 2.

**Table 2**

**Algorithm 2: Grid partition and intermediate region selection algorithm**

<table>
<thead>
<tr>
<th>Input and step 1-7 are the same as Algorithm 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>8: $R_{\text{par}} = \emptyset, i = 1, ..., N, R_{\text{par}} = \emptyset, i = 1, ..., N$.</td>
</tr>
<tr>
<td>9: for $i = 1$ to $Y_{\text{num}} - 1$ do</td>
</tr>
<tr>
<td>10: for $j = 1$ to $X_{\text{num}} - 1$ do</td>
</tr>
<tr>
<td>11: Assign identity $I_{\text{par}} = (X_{\text{num}} - 1) \ast (i - 1) + j$ for the partition space $R_{\text{par}, I_{\text{par}}} = [y_{\text{bound}}, y_{1,\text{bound}}] \times [x_{\text{bound}}, x_{1,\text{bound}}]$ and the region with intermediate region $R_{\text{union}, I_{\text{par}}}$.</td>
</tr>
<tr>
<td>12: for each ${x_{\text{par}}, y_{\text{par}}, x_{\text{bound}}}$ do</td>
</tr>
<tr>
<td>13: if $x_{\text{par}}$ in $R_{\text{par}, I_{\text{par}}}$ then put $x_{\text{par}}$ in $Z_{\text{par}, k}$.</td>
</tr>
<tr>
<td>14: end for</td>
</tr>
<tr>
<td>15: for each ${x_{\text{par}}, y_{\text{par}}, x_{\text{bound}}}$ do</td>
</tr>
<tr>
<td>16: if $x_{\text{par}}$ in $R_{\text{par}, I_{\text{par}}}$ then put $x_{\text{par}}$ in $P_{\text{par}, k}$.</td>
</tr>
<tr>
<td>17: end for</td>
</tr>
<tr>
<td>18: end for</td>
</tr>
<tr>
<td>19: end for</td>
</tr>
<tr>
<td>20: end for</td>
</tr>
<tr>
<td>Output: ${x_{\text{par}}, y_{\text{par}}}, i = 1, ..., N, R_{\text{par}}, i = 1, ..., N, R_{\text{par}}, i = 1, ..., N$.</td>
</tr>
</tbody>
</table>

A four-partition situation with intermediate regions is illustrated in Fig. 2. The particles in the same partition satisfy the condition that the distance of the particle-measurement and measurement is within $n \ast \sigma$, and the particles belong to any other partitions are more far away from the measurements of its own partition. Therefore, the partition and intermediate region strategy ensures the calculation accuracy of the improved SMC-PHD filter.
C. The computationally efficient SMC-PHD filter algorithm

Based upon the partition and intermediate mechanism, the update can be independently calculated by the measurement set \( \{ Z_{i,k} \} \) and the particle set \( \{ P_{i,k} \cup R_{i,k} \} \). The computationally efficient SMC-PHD filter is proposed as follows.

Step 0: Initialization. At time step \( k = 0 \), initialize particle set \( \{ W_0^{(i)}, x_0^{(i)} \}_{i=1}^{\rho_0} \), \( W_0 \) denotes the weight of the particle \( x_0^{(i)} \), and \( \rho_0 \) is the number of the particles.

Step 1: Prediction. The same as the prediction step of the standard SMC-PHD filter introduced in the section II.

Step 2: Partition and intermediate region selection. Train threshold \( T_k \) from the measurement set, execute process of algorithm 2 to divide the particle-measurement set and measurement set into \( N \) partitions \( \{ P_{i,k} \}, \{ Z_{i,k} \} \) and \( \{ R_{i,k} \} \), \( i = 1, \ldots, N \). The number of the particle set \( \{ P_{i,k} \cup R_{i,k} \} \) is \( \{ L_{i,k} \} \), \( i = 1, \ldots, N \).

For \( \text{num} = 1, \ldots, N \),

\[
\tilde{\omega}_{k,\text{num}}^{(i)} = \left[ \psi(x_k^{(i)}, s_{k,\text{num}}) + \sum_{x \in Z_{k,\text{num}}} \psi_{k,x}(x_{k,\text{num}}) \right] \tilde{\omega}_{k|k-1}^{(i)}
\]  
(13)

\[
C_{k,\text{num}}(z) = \sum_{j=1}^{L_{k,\text{num}}} \psi_{k,x}(x_{k,\text{num}}) \tilde{\omega}_{k|k-1}^{(j)}, \quad z \in Z_{k,\text{num}}.
\]

Step 4: Target estimation. Extract \( \hat{T}_{k,\text{num}} \) PHD peaks as the target state estimations from the particle set \( \{ W_{k,\text{num}}^{(i)}, x_{k,\text{num}}^{(i)} \}_{i=1}^{\rho_{k,\text{num}}} \). Compute the sum of the particle weight

\[
\tilde{R}_{k|k,\text{num}} = \sum_{j=1}^{\rho_{k,\text{num}}} \tilde{\omega}_{k|k}^{(j)} , \quad \hat{T}_{k,\text{num}} = \text{round} \left( \tilde{R}_{k|k,\text{num}} \right) \]
gives the estimated target number. Set \( \rho_{k,\text{num}} = \rho \) after resampling, with \( \rho \) denoting particle number per target.

Step 5: Particle resampling. Using the systematic resampling algorithm, we get the resampled particle set \( \{ W_{k,\text{num}}^{(i)}, x_{k,\text{num}}^{(i)} \}_{i=1}^{\rho_{k,\text{num}}} \) with normalized weight \( \tilde{R}_{k|k,\text{num}}/\rho_{k,\text{num}} \).

\[
\text{num} = \text{num} + 1,
\]

Step 6: Data consolidation. Compute the sum of all \( \hat{T}_{k,\text{num}} \) as the target number estimate and combine the PHD peak set of each partition into the state estimation, the union set of the particle sets \( \{ N_{k|k,\text{num}}/\rho_{k}^{(i)}, x_{k,\text{num}}^{(i)} \}_{i=1}^{\rho_{k,\text{num}}} \) is the particle set for the next time step. Set \( k = k + 1 \).

IV. SIMULATIONS

A. Simulation Setup

In order to verify the performance of the proposed SMC-PHD filter, we exploit the Bearing and Range tracking model. The dynamic model, measurements model, new born model and parameter setting for the tracking scenario are detailed in [12]. In the initialization step, set \( L_0 = 1000 \), the number of new born particles \( I_k = 200 \) at each time step and \( \rho \) particles for each expected target. For the partition, we use the mean value of the measurement set as the partition threshold, and the intermediate region size is \( 3\sigma \). For the state estimation, the STPHD algorithm [17] is used. We choose the optimal sub-pattern assignment (OSPA) [18] as the metric for the performance evaluation of the tracking accuracy. In the simulations, we set \( c = 100 \) and \( p = 2 \). To evaluate the average performance, we run 100 Monte Carlo trials for each filter with the same original initialization data and independently generated measurements. Simulations are implemented in MATLAB and conducted with a desktop having 4.00 GB RAM and an Intel i3-4130 3.40GHz processor.

B. Simulation Results

In this section, we present the simulation results with different clutter rate, to compare the performance of the proposed PPSMC-PHD filter with the conventional SMC-PHD filter. There are totally 20 targets in the simulation with respective birth and death time. The position ground truth of the 20 tracks over 40 time steps are displayed in the first subfigure of Fig. 3.

Fig. 3, Fig. 4, Fig. 5 and Fig. 6 show the performance comparisons of three methods by versus time with clutter rate \( r = 0 \) and \( r = 10 \). Where SMC-PHD denotes the standard SMC-PHD method, the PPSMC-PHD2 denotes the proposed method with 2 partitions, and the PPSMC-PHD4 denotes the proposed method with 4 partitions. Compared with the standard SMC-PHD filter by the OSPA value, the results confirm the expectations and further indicate that our proposed method achieve nearly the same estimation accuracy as the SMC-PHD filter or even more accurate performance in some time steps for different clutter rate observation scenes.

The average update time of the three methods with different clutter rates is given in Fig. 4 and Fig. 6. It obviously demonstrates that the proposed method highly increases the computation speed and that the speedup ratio grows linearly with the number of the partitions. However, the computational time of each time step depends on the partition strategy and the measurement set data. Particularly, in our simulation scenario at the beginning time steps the targets gather near the location of newborn position and the number of particles in the intermediate region are very large. Hence, the speed up ratio is close to 1. With the movement of the targets, the target and the particles within the scene gradually spread out.

The simulation results indicate that the proposed PPSMC-PHD filter can achieve comparative accuracy with the standard SMC-PHD filter, and the method is less affected by the clutter rate. In addition, the speedup ratio of the standard SMC-PHD filter and PPSMC-PHD filter depends on the partition strategy and the particle distribution.

The high-speed sigma-gating SMC-PHD filter [16] is a computationally efficient method, but the method has its
inherent shortcoming. The randomness of the noise may lead more and more useful particles distribute outside the gating when the gating size is not enough. The true target measurement without corresponding particles may be neglected during the likelihood calculation, as a result the target which generated the neglected measurement was tracked lost. The tracking failure example of sigma-gating method with clutter \( r = 10 \) is shown in the bottom subfigure of Fig. 6. Where the rectangular blue dashed box denotes the sigma-gating, the red ellipse marks the missing measurement of true target.

![Positions of tracks over time steps](image)

**Fig. 3.** Simulation results with \( r = 0 \). Positions of tracks over 40 time steps (top). Comparisons on target number estimation (bottom).

![Comparisons on OSPA distance with \( r = 0 \)](image)

**Fig. 4.** Simulation results with \( r = 0 \). Comparisons on OSPA distance (top). Comparisons on running time (bottom).

![Comparisons on target number estimation with \( r = 10 \)](image)

**Fig. 5.** Simulation results with \( r = 10 \). Comparisons on target number estimation (top). Comparisons on OSPA distance (bottom).

![Comparisons on running time with \( r = 10 \)](image)

**Fig. 6.** Simulation results with \( r = 10 \). Comparisons on running time (top). Failure situation of the sigma-gating method (bottom).

In the above simulation scenario, the targets gather near the location of newborn position at the beginning time steps. To evaluate the dispersed observations scenario, we use the output data of the SMC-PHD filter \( k = 10 \) as the input data for the proposed method and the sigma-gating method. We evaluate the performance of different from the time step \( k = 11 \) to \( k = 40 \). With the dispersed simulation results, we do a statistical analysis of different methods per 100 runs in the clutter rate \( r = 0 \). The average performance comparisons of the filters are shown in the Table 3. The “Loss number” means the loss tracking number of the true targets. From Table 3, the
simulation results indicate that the sigma-gating method achieves better computational performance than our method, and our method can achieve a better tracking stability than the sigma-gating method. Especially, no loss tracking occurs in the PPSMC-PHD filter under the clutter rate $r = 0$ with different gating size.

In summary, under the premise of ensuring the computational accuracy, the overall computational cost of the update step will be highly reduced in the PPSMC-PHD filter. In addition, the speedup ratio grows linearly with the number of partitions, which is based on the assumption that the measurements and particles are evenly dispersed distributed in the measurement space.

Table 3 Average performance comparisons of the filters with the SMC-PHD data

<table>
<thead>
<tr>
<th>Gate size</th>
<th>Evaluation</th>
<th>SMC-PHD</th>
<th>Sigma-gating</th>
<th>PPSMC-PHD</th>
<th>PPSMC-PHD</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPA</td>
<td>3.5882</td>
<td>5.0545</td>
<td>3.6537</td>
<td>3.6637</td>
<td></td>
</tr>
<tr>
<td>$3\sigma$</td>
<td>Loss number 0</td>
<td>0.047</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Update time</td>
<td>0.11974</td>
<td>0.064184</td>
<td>0.098772</td>
<td>0.068303</td>
<td></td>
</tr>
<tr>
<td>Speedup ratio</td>
<td>1</td>
<td>1.8656</td>
<td>1.2123</td>
<td>1.7531</td>
<td></td>
</tr>
<tr>
<td>OSPA</td>
<td>3.5972</td>
<td>3.6209</td>
<td>3.6398</td>
<td>3.6464</td>
<td></td>
</tr>
<tr>
<td>$4\sigma$</td>
<td>Loss number 0</td>
<td>0.000333</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Update time</td>
<td>0.12044</td>
<td>0.068052</td>
<td>0.10031</td>
<td>0.072161</td>
<td></td>
</tr>
<tr>
<td>Speedup ratio</td>
<td>1</td>
<td>1.7698</td>
<td>1.2007</td>
<td>1.669</td>
<td></td>
</tr>
<tr>
<td>OSPA</td>
<td>3.5957</td>
<td>3.6412</td>
<td>3.6421</td>
<td>3.6501</td>
<td></td>
</tr>
<tr>
<td>$5\sigma$</td>
<td>Loss number 0</td>
<td>0.002</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Update time</td>
<td>0.11972</td>
<td>0.071976</td>
<td>0.10115</td>
<td>0.074175</td>
<td></td>
</tr>
<tr>
<td>Speedup ratio</td>
<td>1</td>
<td>1.6634</td>
<td>1.1837</td>
<td>1.6141</td>
<td></td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we proposed a computationally efficient SMC-PHD filter called PPSMC-PHD filter. The main contributions of this work include: (1) A grid partition strategy is proposed to divide the measurement and particle set into some subsets and solve the related PHD calculation in each partition independently; (2) The gating-based intermediate region strategy is proposed to ensure the computational accuracy, and (3) a computationally efficient SMC-PHD filter algorithm is proposed to speed up the execution and make it possible to parallel the updating step.

In our proposed method, the overall computational cost of the update step is reduced with a little precision loss because only likelihoods for the measurements and particles that are close with each other are calculated. In addition, the partition and intermediate region strategy prevents the intra-partition update from influence of the irrelevant measurements in other partition. Simulations with different clutter rates demonstrate that our approach can achieve a much faster processing rate and nearly the same tracking accuracy compared with the standard SMC-PHD filter. The PPSMC-PHD filter can be efficiently applied to distributed MTT environment or parallel hardware architecture. Our future work will focus on the adaptive partition strategy of the measurements.

ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities (HIT.NSRIF.2014071), Doctoral Program of Higher Education of China (20132302120044), National Natural Science Foundation of China (61175027) and National Natural Science Foundation of China (61305013). The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES