A Novel Merging Algorithm in Gaussian Mixture Probability Hypothesis Density Filter for Close Proximity Targets Tracking

Liming Chen, Zhe Chen, Fuliang Yin*

School of Information and Communication Engineering, Dalian University of Technology
Dalian 116024, China

Abstract

This paper proposes a novel merging algorithm in Gaussian mixture probability hypothesis density filter to track close proximity targets. The proposed algorithm is added after GM-PHD recursion, in a condition that more than one target has the same state. The weights of Gaussian components decide whether the components can be utilized to extract states, and the means and covariances of Gaussian components are used to determine the distance of components. Depending on these weights, means and covariances, the proposed algorithm avoids that the components which have higher weights than other components are merged in foresaid condition. Simulation results show that the new algorithm can enhance the precision of estimation for multi-target states when the targets move closely.

Keywords: Probability Hypothesis Density Filter; Multi-target Tracking; Gaussian Mixture; Close Proximity

1 Introduction

The multi-target tracking is to estimate moving targets’ states by signal processing algorithm. As an attractive field, multi-target tracking can be applied in video surveillance, radar, intelligent transportation systems, medical image, etc. [1, 2, 3, 4, 5].

Most traditional multi-target tracking approaches apply data association techniques, such as Global Nearest Neighbor (GNN), Joint Probabilistic Data Association (JPDA), and Multiple Hypothesis Tracking (MHT) [6, 7, 8]. GNN considers all possible associations within track gates under a constraint that an observation can only be associated with the most likely hypothesis. The GNN approach has lower computational complexity, but only works well in the case of sparse targets and few clutter in the track gates [6]. The JPDA method allows a track to be updated by

*This work was supported by the National Nature Science Foundation of China (No. 61172110, 60772161) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 200801410015).

Corresponding author.

Email addresses: chenliming1127@163.com (Liming Chen), zhechen@dlut.edu.cn (Zhe Chen), flyin@dlut.edu.cn (Fuliang Yin).
all observation in its track gate, however, it suffers from a problem that the targets’ number during tracking should be fixed [7]. The MHT approach hypothesizes all possible data associations over multiple scans and uses subsequent data to resolve the uncertainty of associations, but it overly relies on prior information [8]. All these approaches mentioned above have a common drawback that their computational complexity grows rapidly when the number of measures increases.

To remedy the above problems, Mahler proposed Random Finite Set (RFS) theory to track multiple targets [9, 10, 11]. In this theory, the states of targets and measures may be represented as random sets. Because data association is not included, its main advantage is that computational complexity does not grow rapidly in situation that appearing and disappearing time of targets are unknown and the targets’ number is varying. The Probability Hypothesis Density (PHD) filter [12], as a kind of RFS-based filter, is a recursion form of the posterior intensity, which is a first-order statistical moment of the posterior multi-target states. However, the close-form solutions of the PHD filter are difficult to be obtained, since multiple integrals are included in the recursion. Recently, some close-form solutions for PHD filter, i.e. sequential Monte Carlo PHD (SMC-PHD) filter, Gaussian mixture PHD (GM-PHD) filter, were proposed. SMC-PHD filter, using the sequential Monte Carlo (SMC) technique to propagate the posterior intensity in time, has been presented in [14, 15] to track multiple targets whose number is time-varying in nonlinear system. The main drawbacks of SMC-PHD filter is that the large number of particles lead to the high computational complexity [13]. Another implementation of PHD filter is Gaussian mixture PHD filter [13]. In this filter, a weighted mixture of Gaussian density is used to approximate the posterior intensity function, and the weights, means, and covariances of each Gaussian component can be updated from recursions. The computational load of this approach is reduced because of pruning algorithm for Gaussian components. But there is a precondition in GM-PHD filter, i.e., the distance between targets is not lower than a certain threshold. Most improvements about the GM-PHD filter focus on developing the performance on the premise that the precondition is fulfilled [16, 17], whereas the hypothesis makes the GM-PHD filter limited in applications and can not distinguish targets in close proximity, which often happen in multi-target tracking when there are numerous targets in the scenario.

In this paper, we propose a novel merging algorithm in GM-PHD filter to make the filter have generality. Total information of Gaussian components, including weights, means, and covariances, derived from the GM-PHD recursion is considered in our merging algorithm. The weights of Gaussian components decide whether the components can be utilized to extract the states, and the means and covariances of Gaussian components are used to determine the distance of components. Depending on these weights, means and covariances, the proposed algorithm is robust enough even any two targets are very closed. Simulation results show that the new algorithm can give more accurate states estimate.

This paper is organized as follows. Section 2 describes the linear system model of multi-target tracking, and summarizes the GM-PHD filter. Section 3 presents the new merging algorithm in GM-PHD filter. In section 4, the effectiveness of the proposed approach is demonstrated via a simulation. Finally, conclusions are given in Section 5.

2 Model Description and GM-PHD Filter [13]

Multi-target tracking can be modeled by random finite set framework. Its mathematical formulation is presented first in this section, and followed with description of GM-PHD filter.
Let $x_{k+1}$ and $z_k$ denote the state and the measure respectively, consider the general linear system model

\[ \begin{align*}
x_{k+1} &= F_k x_k + v_k \\
z_k &= H_k x_k + w_k
\end{align*} \tag{1} \]

where $F_k$ is the state transition matrix, $v_k$ is the process noise; $H_k$ is the observation matrix, and $w_k$ is the observation noise.

Based on the linear system model, a linear Gaussian multi-target model is used in GM-PHD filter. Let $f_{k|k-1}(\cdot|\cdot)$ denotes transition density, $g_{k|k-1}(\cdot|\cdot)$ denotes the likelihood function. Each target follows a linear Gaussian model [13], i.e.

\[ \begin{align*}
f_{k|k-1}(x|\zeta) &= N(x; F_{k-1}\zeta, Q_{k-1}) \tag{2} \\
g_{k|k-1}(z|x) &= N(z; H_k x, R_k) \tag{3}
\end{align*} \]

where $x$ is Gaussian random variable with mean $F_{k-1}\zeta$ and process noise covariance $Q_{k-1}$, $z$ is Gaussian random variable with mean $H_k x$ and observation noise covariance $R_k$.

As targets appear and disappear randomly in some applications, the targets at time $k$ may be divided into three kinds [13]. First, the targets exist at time $k - 1$, and continue to exist at time $k$; Second, the targets arise by spontaneous birth; The last kind is the targets arise by spawning from targets at time $k - 1$. According to this classification, the multi-target states are made up of three parts, i.e. the surviving targets, the spontaneous birth targets, and the spawned targets.

Considering some implementation problems, the whole GM-PHD filter includes three parts, i.e., the GM-PHD recursion, the pruning, and the multi-target state extraction.

The GM-PHD recursion has two steps: the prediction step and the update step. In the prediction step, the predicted intensity $v_{k|k-1}$ is given by

\[ v_{k|k-1}(x) = v_{S,k|k-1}(x) + v_{\beta,k|k-1}(x) + \gamma_k(x) \tag{4} \]

where $v_{S,k|k-1}(x)$, $v_{\beta,k|k-1}(x)$, and $\gamma_k(x)$ denote the surviving intensity, spawning intensity, and spontaneous birth intensity respectively. Each of them can be expressed as a Gaussian mixture, i.e.

\[ \begin{align*}
v_{S,k|k-1}(x) &= p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} N(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}) \tag{5} \\
m_{S,k|k-1}^{(j)} &= F_{k-1} m_{k-1}^{(j)} \tag{6} \\
P_{S,k|k-1}^{(j)} &= Q_{k-1} + F_{k-1} P_{k-1} F_{k-1}^T \tag{7} \\
v_{\beta,k|k-1}(x) &= \sum_{j=1}^{J_{k-1}} \sum_{l=1}^{J_{\beta,k}} w_{k-1}^{(j)} w_{\beta,k}^{(l)} N(x; m^{(j,l)}_{\beta,k|k-1}, P^{(j,l)}_{\beta,k|k-1}) \tag{8} \\
m_{\beta,k|k-1}^{(j,l)} &= F_{\beta,k-1} m_{k-1}^{(j)} + d_{\beta,k-1}^{(l)} \tag{9} \\
P_{\beta,k|k-1}^{(j,l)} &= Q_{\beta,k-1} + F_{\beta,k-1} P_{\beta,k-1} F_{\beta,k-1}^T (F_{\beta,k-1})^T \tag{10} \\
\gamma_k(x) &= \sum_{j=1}^{J_{\gamma,k}} w_{\gamma,k}^{(j)} N(x; m_{\gamma,k}^{(j)}, P_{\gamma,k}^{(j)}) \tag{11}
\end{align*} \]
where \( p_{S,k} \) is the probability that a target which exists at time \( k-1 \) still exists at time \( k \); \( J_{k-1}, w^{(j)}_{k-1}, m^{(j)}_{k-1}, P^{(j)}_{k-1}, j = 1, \ldots, J_{k-1} \), are the number, weights, means, and covariances of Gaussian components in the posterior intensity at time \( k-1 \); \( J_{\bar{\beta},k}, w^{(l)}_{\bar{\beta},k}, F^{(l)}_{\bar{\beta},k-1}, P^{(l)}_{\bar{\beta},k-1}, Q^{(l)}_{\bar{\beta},k-1}, j = 1, \ldots, J_{k-1}, l = 1, \ldots, J_{\bar{\beta},k} \), determine the spawning intensity; \( J_{\gamma,k}, w^{(j)}_{\gamma,k}, m^{(j)}_{\gamma,k}, P^{(j)}_{\gamma,k}, j = 1, \ldots, J_{\gamma,k} \), are the parameters of spontaneous birth intensity \( \gamma_k(x) \).

The predicted intensity \( v_{k|k-1}(x) \) has the following Gaussian mixture form

\[
v_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} w^{(i)}_{k|k-1} \mathcal{N}(x; m^{(i)}_{k|k-1}, P^{(i)}_{k|k-1})
\]

In the update step, the posterior intensity \( v_k(x) \) at time \( k \) is given by a Gaussian mixture, i.e.

\[
v_k(x) = (1 - p_{D,k})v_{k|k-1} + \sum_{z \in Z_k} v_{D,k}(x; z)
\]

\[
v_{D,k}(x; z) = \sum_{j=1}^{J_{k|k-1}} w^{(j)}_{k}(z) \mathcal{N}(x; m^{(j)}_{k|k}(z), P^{(j)}_{k|k})
\]

\[
w^{(j)}_{k}(z) = \frac{p_{D,k}w^{(j)}_{k|k-1}q^{(j)}_{k}(z)}{\kappa_k(z) + p_{D,k}\sum_{i=1}^{J_{k|k-1}} w^{(i)}_{k|k-1}q^{(i)}_{k}(z)}
\]

\[
q^{(j)}_{k}(z) = \mathcal{N}(z; H_k m^{(j)}_{k|k-1}, R_k + H_k F^{(j)}_{k|k-1} H_k^T)
\]

\[
m^{(j)}_{k|k}(z) = m^{(j)}_{k|k-1} + K^{(j)}_{k}(z - H_k m^{(j)}_{k|k-1})
\]

\[
F^{(j)}_{k|k} = [I - K^{(j)}_{k} H_k] P^{(j)}_{k|k-1}
\]

\[
K^{(j)}_{k} = P^{(j)}_{k|k-1} H_k^T (H_k F^{(j)}_{k|k-1} H_k^T + R_k)^{-1}
\]

where \( p_{D,k} \) is the probability of detection at time \( k \), \( \kappa_k(z) \) is the intensity of clutter random finite set at time \( k \), and \( Z_k \) is the measures.

Let \( J_k \) denotes the number of Gaussian components in posterior intensity at time \( k \). From (13) and (14) we obtain

\[
v_k(x) = \sum_{i=1}^{J_k} w^{(i)}_{k} \mathcal{N}(x; m^{(i)}_{k}, P^{(i)}_{k})
\]

The second part of GM-PHD filter is pruning step. In implementation, the GM-PHD filter suffers from computation problem, since the number of Gaussian components in the posterior intensity increases without bound [13]. The pruning step is used to decrease the number of the components. Components whose weights are lower than the threshold set previously are discarded. Then the components whose means are close together are merged into one component.

The last part of GM-PHD filter is multi-target state extraction. Since higher peaks of \( v_k \) is more probable for presenting targets, the states can be extracted from the means of components corresponding to these peaks. And because higher peaks of \( v_k \) imply components with greater weights, the means of the components whose weights are greater than a threshold are used to extract targets’ states.
As before, GM-PHD filter, as a closed-form solution of the PHD recursion, can track targets while the targets’ number is unknown under lots of noises in multi-target tracking problem. However, as mentioned in the introduction, how to improve the estimate precision for close proximity targets is still a problem to be resolved in the application of GM-PHD filter.

3 A Novel Merging Algorithm in GM-PHD Filter

As mentioned in previous section, in order to resolve computation problem, there is a pruning part in GM-PHD filter to truncate and merge the Gaussian components. The merging algorithm in pruning part considers only the condition that the components’ means are close together, whereas another condition whether the components represent the different state estimates or not is ignored unfortunately [13]. Therefore, the disadvantage of the pruning part is that it has a limitation that the GM-PHD filter can not distinguish the targets when the distance between them is lower than a threshold [10]. To remedy this problem, focused on the pruning step, we propose a novel merging algorithm in GM-PHD filter.

As we known, a Gaussian component has three parameters, i.e., weight, mean, and covariance. In pruning step of GM-PHD filter, the decision criterion which is used to judge whether components should be merged only utilizes means and covariance, whereas the weights of components are not used. Thus in proposed algorithm, we merge the components depending on a new condition which utilizes all three parameters of the components.

Let $G$ denotes the Gaussian components \{${w}_k^{(i)}, m_k^{(i)}, P_k^{(i)}$\}_{j=1}^{J_k}$ derived from GM-PHD recursion at time $k$, the new algorithm is presented as follows.

**Step 1** Truncation. The components which have the weights below the threshold $T_{\text{truncating}}$ will be discarded. The truncating result presented by $G_{\text{truncating}}$ is kept for further use.

**Step 2** First merging. The components which are close together are merged into one by condition

$$\begin{align*}
(m_k^{(i)} - m_k^{(j)})^T(P_k^{(i)})^{-1}(m_k^{(i)} - m_k^{(j)}) &\leq T_{1st}
\end{align*}
$$

where $T_{1st}$ is a threshold. Although this step decreases the precision when distance between the targets is lower than a certain threshold, the merging part is still necessary in the proposed approach as the number of components may increase rapidly without it. However, there still exist a problem that the targets can not be distinguished in the situation that the targets are so close together that their corresponding components are satisfied with the merging condition (21).

**Step 3** State extraction. The components which have the weights higher than a threshold $T_{\text{state}}$ are used to extract the multi-target states. Rounded value $\text{round}(w_k^{(i)})$ of the satisfied weights is the number of targets at the position, which can be extracted from the mean $m_k^{(i)}$ of corresponding Gaussian component. When $\text{round}(w_k^{(i)}) > 1$, more than one target with same state can be extracted from $m_k^{(i)}$. Suppose that the output of GM-PHD filter has no clutter, compared with true state, the estimated targets which have same state can be divided into two kinds. The first is that they are close together; The second is that they are at same position indeed. Here, we consider how to resolve the first kind situation.
Step 4 Second merging. This step is used to reduce the number of merged components, which represent different targets. It is necessary if more than one target at the same position in the result of step 3. The input of step 4 is $G_{\text{truncating}}$, which has been kept in step 1. Different from the merging algorithm in step 2, the components which are satisfied with the merging condition (21) are merged selectively. The relationship between $w_k^{(i)}$ and $T_{\text{state}}$ is applied to judge whether the satisfied components represent targets or not. And depending on it, the satisfied components can be divided into two parts: one part of the components which match the condition $w_k^{(i)} > T_{\text{state}}$ can be used to extract the targets’ states, while another part which can not match the condition are discarded as they have lower probability for representing targets. So the second merging step is that if the components $G_{\text{truncating}}$ are satisfied with two conditions simultaneously, they could be merged: one is the condition (21), another is that the weights are lower than a threshold $T_{2nd}$. New weight after merging is the sum of weights of satisfied components.

Step 5 State extraction. The purpose of this step is to extract the multi-target states from the result of the second merging. The means of the components that have weights greater than a threshold $T_{\text{state}}$ are selected. Then the position information is obtained according to the selected means.

As merging and state extraction operation of steps 4 and 5 also appears in steps 2 and 3, it seems if we made use of the second merging after step 1, the following steps 2 and 3 of the proposed pruning and state extraction algorithm can be omitted. However, because merging condition of step 4 is stricter than step 2, the number of merged components may become less. It induces that some new components representing the estimated states after merging do not have greater weights than $T_{\text{state}}$. Although the way that components with weights greater than threshold $T_{\text{state}}$ are used to extract targets’ states is a better alternative in GM-PHD filter, it does not indicate that the components with weak weights do not represent estimated states. As a result, if steps 2 and 3 are omitted, some states represented by those components might be missed at the step of state extraction. Therefore, the proposed algorithm uses the second merging step only when more than one target has the same state after step 3.

Let \( \setminus \) denotes the difference of two sets, according to the improvement mentioned before, improved GM-PHD filter is summarized as follows.

1. GM-PHD Recursion

Let \( \{w_{\gamma,k}^{(i)}, m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}\}_{i=1}^{J_{\gamma,k}} \) and \( \{w_{\beta,k}^{(i)}, m_{\beta,k}^{(i)}, P_{\beta,k}^{(i)}\}_{i=1}^{J_{\beta,k}} \) denote the Gaussian components of spontaneous birth intensity \( \gamma_k(x) \) and spawning intensity \( v_{\beta,k|k-1}(x) \) respectively.

input: measurement set \( Z_k \) and the Gaussian components \( \{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k} \) at time \( k \).

(a) According to (4)-(11), using \( \{w_{\gamma,k}^{(i)}, m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}\}_{i=1}^{J_{\gamma,k}}, \{w_{\beta,k}^{(i)}, m_{\beta,k}^{(i)}, P_{\beta,k}^{(i)}\}_{i=1}^{J_{\beta,k}}, \{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k} \) of predicted intensity \( v_{\beta,k|k-1}(x) \) in (12).

(b) According to (13)-(19), using \( \{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k} \) of posterior intensity \( v_k(x) \) in (20).

output: \( G = \{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k} \)
(2) Truncation of Gaussian Components
for \( i = 1, \ldots, J_k \)
if \( w_k^{(i)} < T_{\text{truncating}} \)
\[
G_{\text{truncating}} = G \setminus \{ w_k^{(i)}, m_k^{(i)}, P_k^{(i)} \}.
\]
end
output: \( G_{\text{truncating}} = \{ w_k^{(i)}, m_k^{(i)}, P_k^{(i)} \}_{i=1}^{J_k} \).

(3) First merging
input: \( G_{\text{truncating}} = \{ w_k^{(i)}, m_k^{(i)}, P_k^{(i)} \}_{i=1}^{J_k} \).
Let \( l = 0, I = \{ 1, \ldots, J_{k,\text{truncating}} \} \), \( J_{\max} \) is a maximum number of Gaussian components.
while \( I \neq \emptyset \)
\[
l = l + 1, \\
j = \arg \max_{i \in I} w_k^{(i)}, \\
I_{1st} = \{ i \in I | (m_k^{(i)} - m_k^{(j)})^T (P_k^{(j)})^{-1} (m_k^{(i)} - m_k^{(j)}) \leq T_{1st} \}, \\
w_k^{(l)} = \sum_{i \in I_{1st}} w_k^{(i)}, \\
m_k^{(l)} = \frac{1}{w_k^{(l)}} \sum_{i \in I_{1st}} w_k^{(i)} m_k^{(i)}, \\
P_k^{(l)} = \frac{1}{w_k^{(l)}} \sum_{i \in I_{1st}} w_k^{(i)} (P_k^{(i)} + (m_k^{(l)} - m_k^{(i)})(m_k^{(l)} - m_k^{(i)})^T), \\
I = I \setminus I_{1st}.
\]
end
output:
\[
G_{1st} = \{ w_{k,1st}^{(i)}, m_{k,1st}^{(i)}, P_{k,1st}^{(i)} \}_{i=1}^{J_{k,1st}}, \\
J_{k,1st} = \min \{ l, J_{\max} \}.
\]

(4) State extraction
Let multi-target state estimate \( \hat{X}_k = \emptyset, m = 1 \) denotes that more than one target has the same state.
Initialize \( m = 0 \).
input: \( G_{1st} = \{ w_{k,1st}^{(i)}, m_{k,1st}^{(i)}, P_{k,1st}^{(i)} \}_{i=1}^{J_{k,1st}} \).
for \( i = 1, \ldots, J_{k,1st} \)
if \( w_{k,1st}^{(i)} > T_{\text{state}} \) && \( \text{round}(w_{k,1st}^{(i)}) > 1 \)
\[
m = 1, \\
b \text{break}
\]
end
if \( w_{k,1st}^{(i)} > T_{\text{state}} \) && \( \text{round}(w_{k,1st}^{(i)}) == 1 \)
\[
\hat{X}_k = \lbrack \hat{X}_k, m_{k,1st}^{(i)} \rbrack.
\]
(5) Second merging

input: \( G_{\text{truncating}} = \{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k,\text{truncating}} \).

if \( m = 1 \)

\[ I = \{1, \ldots, J_k,\text{truncating}\}, \quad l = 0. \]

while \( I \neq \emptyset \)

\[ l = l + 1, \]
\[ j = \arg \max_{i \in I} w_k^{(i)}, \]
\[ I_{2nd} = \{i \in I | ((m_k^{(i)} - m_k^{(j)})^T(P_k^{(i)})^{-1}(m_k^{(i)} - m_k^{(j)}) \leq T_{1st}) \& \&
              (w_k^{(i)} \leq T_{2nd}) \& \& (w_k^{(j)} \leq T_{2nd})\}, \]
\[ w_k^{(i)}_{2nd} = \sum_{i \in I_{2nd}} w_k^{(i)}, \]
\[ m_k^{(i)}_{2nd} = \frac{1}{w_k^{(i)}_{2nd}} \sum_{i \in I_{2nd}} w_k^{(i)} m_k^{(i)}, \]
\[ P_k^{(i)}_{2nd} = \frac{1}{w_k^{(i)}_{2nd}} \sum_{i \in I_{2nd}} w_k^{(i)} (P_k^{(i)} + (m_k^{(i)} - m_k^{(i)})(m_k^{(i)} - m_k^{(i)})^T), \]
\[ I = I \setminus I_{2nd}. \]

end

end

output:

\[ G_{2nd} = \{w_k^{(i)}_{2nd}, m_k^{(i)}_{2nd}, P_k^{(i)}_{2nd}\}_{i=1}^{J_k,2nd}, \]
\[ J_k,2nd = \min\{l, J_{max}\}. \]

(6) State extraction

Let multi-target state estimate \( \hat{X}_k = \emptyset. \)

input: \( G_{2nd} = \{w_k^{(i)}_{2nd}, m_k^{(i)}_{2nd}, P_k^{(i)}_{2nd}\}_{i=1}^{J_k,2nd} \).

for \( i = 1, \ldots, J_k,2nd \)

if \( w_k^{(i)}_{2nd} > T_{\text{state}} \)

\[ \hat{X}_k = [\hat{X}_k, m_k^{(i)}_{2nd}]. \]

end

end

(7) Update the input of GM-PHD recursion at time \( k + 1. \)

if \( m == 1 \)

\[ J_k = J_k,2nd, \]
\[ \{w_k^{(i)}_{2nd}, m_k^{(i)}_{2nd}, P_k^{(i)}_{2nd}\}_{i=1}^{J_k,2nd} = \{w_k^{(i)}_{2nd}, m_k^{(i)}_{2nd}, P_k^{(i)}_{2nd}\}_{i=1}^{J_k,2nd}. \]
else

\begin{align*}
J_k &= J_{k,1st}, \\
\{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k} &= \{w_{k,1st}^{(i)}, m_{k,1st}^{(i)}, P_{k,1st}^{(i)}\}_{i=1}^{J_{k,1st}}.
\end{align*}

end

output: multi-target state estimate \( \hat{X}_k \) at time \( k \), and the Gaussian components \( \{w_k^{(i)}, m_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k} \) at time \( k \).

As before, the improved GM-PHD filter has more complex pruning and state extraction algorithm, which causes its computational load increases correspondingly. Table 1 shows the computational complexity of GM-PHD filter and improved GM-PHD filter for a recursion from time \( k-1 \) to \( k \), when the second merging algorithm is needed. For the convenience of comparison, suppose that Gaussian components’ numbers of two filters at time \( k-1 \) are also the same, named \( J_{k-1} \). Let \( n_m \) denotes the dimension of a target’s state vector, \( n_z \) denotes the dimension of measurement vector, \(|Z_k|\) denotes the number of measurement at time \( k \), and \( J_{k,2nd} \) denotes the number of Gaussian components \( G_{2nd} \) in the second merging step. As the first merging step is the same as the merging algorithm of pruning step of GM-PHD filter, the Gaussian components’ numbers of both outputs are identical, named \( J_{k,1st} \). Further, there is a relationship \( J_{k,1st} \leq J_{k,2nd} \), as the merging condition of the second merging is stricter than the first merging algorithm. And because the computational complexity of set \( I_{1st} \) and \( I_{2nd} \) in the first and second merging steps changes when the Gaussian components are different, the computational complexity of both two filters has maximum and minimum results, which are given in Table 1. Compared the maximum and minimum computational complexity of two filters respectively, it can be seen from Table 1 that improved GM-PHD filter has higher computational load than GM-PHD filter, however, the additional computational complexity only takes place when more than one target has the same estimated position.

| Table 1: Computational complexity of GM-PHD filter and improved GM-PHD filter |
|-------------------------------------------------|-------------------------------------------------|
| GM-PHD filter                                    | Improved GM-PHD filter                          |
| Maximum                                          | Maximum                                          |
| \( O(n_m^2J_{k,1st}^2 + n_m^2J_{k,1st}J_{k-1}|Z_k|) \) | \( O(n_m^2J_{k,1st}J_{k-1}|Z_k| + n_m^2J_{k,2nd}J_{k-1}|Z_k|) \) |
| Minimum                                          | Minimum                                          |
| \( O(n_m^2J_{k,1st}^2 + (n_m^2 + n_z^2)J_{k,1st}J_{k-1}|Z_k|) \) | \( O(n_m^2J_{k,1st}J_{k-1}|Z_k| + n_m^2J_{k,2nd}^2) \) |

4 Simulation Results and Discussion

In simulation experiment, two simulation examples are presented to compare performance of the improved GM-PHD filter and the GM-PHD filter in a two-dimensional surveillance scenario.

Example 1: The surveillance region is 300 meters long and 300 meters wide, involving four targets. This simulation does not include any spawning targets from existing ones. In order to describe the targets conveniently, we name all the targets from 1 to 4. A couple of the targets, namely target 3 and target 4, are close, and the distance between them is 21.21 meters. The initial position and survival time of the targets are given in Table 2. The detected measurements also have clutter which is modeled as a Poisson distribution with intensity \( \kappa_k = \lambda_c V u(z) \), where
deviation of the measurement noise.

From Fig. 1, it is clear that the GM-PHD filter has trouble to estimate targets' position where

\[ \sigma \]

and 0

\[ \square \]

V = 9 \times 10^4 m^2 is the area of surveillance region, \( u(\cdot) \) is the uniform probability density over surveillance region, and \( \lambda_c = 2 \times 10^{-5} m^{-2} \) is the average clutter intensity.

The target state vector \( x_k \) at time \( k \) comprises the position \( [p_{x,k}, p_{y,k}]^T \) and the velocity \( [v_{x,k}, v_{y,k}]^T \), i.e. \( x_k = [p_{x,k}, p_{y,k}, v_{x,k}, v_{y,k}]^T \). And the linear system model (1) contains the transition matrix \( F_k \) and the observation matrix \( H_k \), i.e.

\[
F_k = \begin{bmatrix} I_2 & TI_2 \\ 0_2 & I_2 \end{bmatrix}, \quad H_k = \begin{bmatrix} I_2 & 0_2 \end{bmatrix} \tag{22}
\]

where \( T \) is sampling period of 1 s, \( I_2 \) and 02 denote the 2x2 identity and zero matrices respectively. The process noise \( v_k \) and the observation noise \( w_k \) are both zero mean Gaussian noise with covariance \( Q_k \) and \( R_k \) respectively, i.e.

\[
Q_k = \begin{bmatrix} 0.25T^4I_2 & 0.5T^3I_2 \\ 0.5T^3I_2 & 0.25T^4I_2 \end{bmatrix} \sigma_v^2, \quad R_k = I_2 \sigma_w^2 \tag{23}
\]

where \( \sigma_v = 5m/s^2 \) is the standard deviation of the process noise, and \( \sigma_w = 10m \) is the standard deviation of the measurement noise.

Targets 1 and 2 appear at two different position when \( k = 1s \). The initial Gaussian components \( \{w_1^{(i)}, m_0^{(i)}, P_0^{(i)}\}_{i=1}^{2} \) are given as follows: \( J_0 = 2, w_0^{(1)} = w_0^{(2)} = 0.1, m_0^{(1)} = \begin{bmatrix} 200, 300, 0, 0 \end{bmatrix}^T, m_0^{(2)} = \begin{bmatrix} 5, 5, 0, 0 \end{bmatrix}^T, P_0^{(1)} = P_0^{(2)} = \text{diag}(\begin{bmatrix} 100, 100, 25, 25 \end{bmatrix}^T) \). The spontaneous birth intensity

\[
\gamma_k(x) = \sum_{i=1}^{2} \omega_r^{(i)} N(x, m_r^{(i)}, P_r^{(i)}) \tag{24}
\]

where \( \omega_r^{(1)} = \omega_r^{(2)} = 0.1, m_r^{(1)} = \begin{bmatrix} 50, 255, 0, 0 \end{bmatrix}^T, m_r^{(2)} = \begin{bmatrix} 35, 270, 0, 0 \end{bmatrix}^T, P_r^{(1)} = P_r^{(2)} = \text{diag}(\begin{bmatrix} 100, 100, 25, 25 \end{bmatrix}^T) \). The survival probability of an existing target is \( p_{S,k} = 0.99 \), and the detection probability is \( p_{D,k} = 0.98 \). In the truncation and the first merging step, the parameter used to truncate components \( T_{truncating} = 10^{-5} \), the threshold in the first merging \( T_{1st} = 4 \), and the maximum number of components \( J_{\max} = 100 \). In the state extraction step, the threshold \( T_{state} = 0.5 \). In the second merging step, since \( T_{2nd} \) is used to select Gaussian components representing multi-target state estimates from truncation step’s result \( G_{truncating} \), it has the same selection standard as \( T_{state} \). Based on this point, \( T_{2nd} \) is chosen as 0.5 according to the selection of threshold \( T_{state} \).

Figs. 1 and 2 show the position estimates for measurements using GM-PHD filter and improved GM-PHD filter respectively. Because of high clutter, both of the filters have some errors in the outputs. From Fig. 1, it is clear that the GM-PHD filter has trouble to estimate targets’ position

<table>
<thead>
<tr>
<th>Initial Position(m)</th>
<th>Start Time(s)</th>
<th>End Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>(200, 300)</td>
<td>( k = 1 )</td>
</tr>
<tr>
<td>Target 2</td>
<td>(5, 5)</td>
<td>( k = 1 )</td>
</tr>
<tr>
<td>Target 3</td>
<td>(50, 255)</td>
<td>( k = 60 )</td>
</tr>
<tr>
<td>Target 4</td>
<td>(35, 270)</td>
<td>( k = 60 )</td>
</tr>
</tbody>
</table>
Fig. 1: GM-PHD filter estimates and true tracks of x and y coordinates against time

Fig. 2: Improved GM-PHD filter estimates and true tracks of x and y coordinates against time

when the targets are close. From $k = 62s$ to $k = 100s$, the estimated position of targets 3 and 4 are identical at the same time. They are difficult to be distinguished, since the Gaussian components which have high weights and are satisfied with the condition (21) are merged into one component. In comparison, the improved GM-PHD filter gets more exact estimated position than GM-PHD filter from $k = 62s$ to $k = 100s$, which can be seen from Fig. 2. Also note that the estimated position of targets 3 and 4 at $k = 60s$ and $k = 61s$ are distinguished in Fig. 1. A likely
explanation is that the GM-PHD filter’s estimate results are close to the true targets’ locations at these two seconds since the initial position is accurate.

Two criteria, Wasserstein distance (WD) [18] and circular position accuracy probability (CPAP) [19], are used for performance evaluation. Let $X = \{x_1, \cdots, x_{|X|}\}$ denote the true target state, and $\hat{X} = \{\hat{x}_1, \cdots, \hat{x}_{|\hat{X}|}\}$ denotes the estimated target state, then the WD is defined as

$$d(X, \hat{X}) = \min_C \left( \sum_{i=1}^{|X|} \sum_{j=1}^{|\hat{X}|} C_{i,j} \| \hat{x}_i - x_j \|_2^2 \right)^{\frac{1}{2}}$$

where the minimum is taken over all transportation matrices $C = \{C_{i,j}\}$, and each entry of $C$ satisfies $C_{i,j} > 0$, $\sum_{j=1}^{|\hat{X}|} C_{i,j} = 1/|\hat{X}|$ and $\sum_{i=1}^{|X|} C_{i,j} = 1/|X|$. Let $N_{MC}$ denotes the number of runs in a Monte Carlo (MC) simulation and $X_m$ denotes the true target state, then the CPAP is defined as

$$\text{CPAP}(r) = \frac{1}{N_{MC}} \sum_{m=1}^{N_{MC}} \frac{1}{|X_m|} \sum_{x \in X_m} P_m(r)$$

for some position accuracy radius $r$, where

$$P_m(r) = \begin{cases} 1, & \|H\hat{x} - Hx\|_2 < r, \ \exists \hat{x} \in \hat{X} \\ 0, & \|H\hat{x} - Hx\|_2 \geq r, \ \forall \hat{x} \in \hat{X} \end{cases}$$

where $H = [I_2, 0_2]$ and $\| \cdot \|_2$ denotes the 2-norm.

Fig. 3 (a) shows the WD of GM-PHD filter and improved GM-PHD filter. There are some spikes in the plot of the WD for both GM-PHD filter and our method. This is because the WD penalizes the difference in cardinalities between two finite sets. When the number of actual targets is different from the number of estimated targets at time $k$, a peak appears in the plot of the WD. The WD of the two methods are almost the same before $k = 62s$. This observation can be expected since the improvement of proposed algorithm appears in a condition that more than one object has the same state, and the condition could be easily satisfied from $k = 60s$ to $k = 100s$ in the scenario where two targets move closely. As a consequence, the WD of two filters are nearly identical with each other before $k = 62s$, whereas the WD of new algorithm GM-PHD filter is lower than the WD of GM-PHD filter after $k = 62s$, when the value of WD is not a peak.

Fig. 3 (b) shows the CPAP of the two filters with radius fixed at $r = 3m$, and 100 Monte Carlo simulations are performed. Observe that the CPAP decreases obviously after $k = 60s$, whereas the CPAP of improved GM-PHD filter is higher than the CPAP of GM-PHD filter, where close proximity targets appear. According to (26) and (27), it is clear that the performance of filter is better while its CPAP is higher. As a result, Fig. 3 (b) suggests that the improved GM-PHD filter enhances the estimate precision compared with GM-PHD filter.

In order to compare the robustness of the improved GM-PHD filter and GM-PHD filter, 100 Monte Carlo runs are performed when average clutter intensity and detection probability vary respectively. Fig. 4 shows the tracking performance for varying average clutter intensity $\lambda_c$ with detection probability fixed at $p_{D,D,k} = 0.98$. Observe that WD of improved GM-PHD filter is lower than GM-PHD filter and CPAP of improved GM-PHD filter is higher regardless of value of average clutter intensity. This indicates that improved GM-PHD filter has better tracking performance and this result confirms the analysis before that merging algorithm with stricter
merging condition makes the number of merged components less and estimated targets’ states more exact. Furthermore, note that both the gaps of WD and CPAP between the two filters increase as $\lambda_c$ increases, which means that superiority of improved GM-PHD filter becomes more obvious while the average clutter intensity becomes greater. Fig. 5 shows the performance for varying detection probability $p_{D,k}$ with average clutter intensity fixed at $\lambda_c = 2 \times 10^{-5} m^{-2}$. It can be seen that the improved GM-PHD filter has better performance than GM-PHD filter, although the performance of the two filters enhances as $p_{D,k}$ increases.

Example 2: In this example, the performance of improved GM-PHD filter and GM-PHD filter is
discussed when the position relationship of close proximity targets varies. The surveillance region is 300 meters long and 350 meters wide, and the area of surveillance region $V = 1.05 \times 10^5 \text{m}^2$. For simplicity, compared with example 1, only the initial position of target 4 is changed. Let $d$ denotes the distance between targets 3 and 4, and $\theta$ denotes the angle relation of them. Fig. 6 shows the sketch map of targets’ initial position. According to the map, the parameter $m_r(2)$ of the birth RFS’s intensity is $m_r(2) = [m_r(1) + d \cos \theta, m_r(1) + d \sin \theta, 0, 0]^T$.

For comparison, 100 Monte Carlo runs are performed for both two filters when the distance $d$ and the angle $\theta$ vary respectively. Firstly, the distance $d$ is varied from 5$m$ to 35$m$ when the angle $\theta$ fixed at $\theta = 0.75\pi \text{ rad}$. Fig. 7 shows the time averaged WD and time averaged CPAP versus the distance $d$. It can be seen from Fig. 7 (a) that since the peaks in WD make the variety of time averaged WD subtle, the gap of time averaged WD between the two filters does not vary obviously. However, the gap between the two filters in Fig. 7 (b) shows that performance of improved GM-PHD filter becomes much better than the GM-PHD filter when $d < 24m$. Secondly, the angle $\theta$ is varied from 0 rad to $2\pi \text{ rad}$ when the distance $d$ fixed at $d = 21m$. Fig. 8 shows the performance of two filters versus the angle $\theta$. Note that the improved GM-PHD filter has better performance than GM-PHD filter no matter how the angle $\theta$ changes. As a consequence, the
Fig. 7: WD and CPAP of GM-PHD and improved GM-PHD filters for varying $d$ ($\theta = 0.75\pi$ rad is fixed; CPAP radius $r = 3m$)

Fig. 8: WD and CPAP of GM-PHD and improved GM-PHD filters for varying $\theta$ ($d = 21$ m is fixed; CPAP radius $r = 3m$)

performance of improved GM-PHD filter relates to the distance between close proximity targets and is not relevant to the angle between targets when the distance fixed.

5 Conclusion

This paper presents a novel merging algorithm in GM-PHD filter to improve the performance of multi-target state estimation in close proximity targets environment. The proposed algorithm avoids that the components which have higher weights than other components are merged, when more than one target has the same state. The weights of Gaussian components are used to decide whether the components can be utilized to extract states. The means and covariances of components are used to determine the distance of components. Depending on these weights, means, and covariances, the proposed algorithm not only merges the Gaussian components to reduce the computational load in the GM-PHD recursion, but also decreases the probability that different targets have the same estimated state. Simulation results show that the proposed
algorithm is robust enough even any two targets are very close. We can conclude that the improved GM-PHD filter is recommended in such scenarios, and it has good robustness to track close targets.

References

