A High-speed Algorithm for Particle CBMeMBer Filter

X.Y. He, G.X. Liu*

Abstract—**The particle cardinality-balanced multi-target multi-Bernoulli (CBMeMBer) filter developed recently is an effective multi-target tracking (MTT) algorithm for nonlinear tracking models. However, the main drawback of this filter is that a significant amount of time is required to compute measurement-updated tracks in the update step, and discard particles with low weights and reproduce particles with high weights in the resampling step. To overcome such a drawback, a high-speed algorithm for particle CBMeMBer filter is proposed in this paper, which modifies the particle CBMeMBer recursion equations by taking the predicted state estimations and measurement likelihoods into consideration. The performance of the presented algorithm has been verified by numerical simulation experiments.**

Index Terms—**Multi-Bernoulli filter, multi-target tracking, random finite set, particle filter**

I. Introduction

THE purpose of multi-target tracking (MTT) is to jointly obtain the target number estimations and target state estimations from a sequence of measurement sets in the presence of clutter [1]. Most traditional MTT approaches require the association between measurements and targets [2-6]. However, these data association methods are computationally intensive in general. Recently, more and more researchers have used the random finite set (RFS) theory [7] in their publications to tackle with different MTT problems. The probability hypothesis density (PHD) multi-target filter [8] is an effective approach for tracking multiple targets based on the RFS theory, it can simultaneously estimate the number and the state of targets without the measurement-to-track association used in the traditional MTT approaches. The PHD filter needs to calculate multiple integrals and the integrals might be also intractable in many cases of interest. In order to overcome the inherent intractability of the PHD filter, two major implementations for the PHD filter have been developed. One is known as the SMC-PHD filter or particle PHD filter [9, 10] and the other is known as the GM-PHD filter [11, 12]. The particle PHD filter uses a large number of particles to approximate the PHD distribution, whereas the GM-PHD filter estimates the PHD distribution as a mixture of Gaussian densities. Convergence results for the particle PHD filter and GM-PHD filter have been given in [13] and [14], respectively. The resulting PHD filter subsequently became a very popular multi-target tracking method with applications in sonar image tracking [15], video target tracking [16, 17], vehicle cooperative localization [18], etc.

The multi-target multi-Bernoulli (MeMBer) recursion [1], which propagates the multi-target posterior density approximately, is another approximation to the multi-target Bayes filter using multi-Bernoulli RFS. However, it has been analyzed that the MeMBer filter overestimates the number of targets. A satisfactory solution named the cardinality-balanced MeMBer (CBMeMBer) filter has been proposed to reduce the posterior cardinality bias by modifying the measurement-updated tracks parameters [19]. Similar to the PHD filter, there are two major implementations of the CBMeMBer filter known as the particle CBMeMBer filter and the Gaussian mixture (GM-CBMeMBer) filter [19]. Moreover, the convergence results for the particle CBMeMBer filter have been given in [20]. Afterwards, the work in [21] proposes an improved MTT algorithm based on the CBMeMBer filter and variational Bayesian approximation to track multiple targets for the linear Gaussian models with unknown measurement noise variances. To track multiple manoeuvring targets, two different extensions based on the CBMeMBer filter and the multi-model method have been proposed in [22] and [23], respectively. Following the CBMeMBer filter in MTT scenarios, a forward-backward CBMeMBer smoothing algorithm aimed at improving the performance of the CBMeMBer-based filtering algorithms was proposed in [24].

With the particle implementation, the CBMeMBer filter is advantageous for the reason that it does not require the additional clustering technique for estimating target states at each time step. Therefore, in the particle CBMeMBer filter extracting the state estimations is reliable and inexpensive. However, for the particle CBMeMBer filter, a significant amount of time is required to compute measurement-updated tracks in the update step, and eliminate particles with low weights and multiply particles with high weights in the resampling step. The computational complexity of computing the measurement-updated tracks is linear in the cardinality of measurement set. Since the measurement set received at each

Manuscript received June 20, 2016; revised October 10, 2016. This work was supported by the Fund for Preliminary Research Field of Equipment (Grant No. 6140001010216DZ01001), the 13th Five-Year Preliminary Research Project of Equipment (Grant No. 31511070403), and the Preliminary Research Foundation of Weapon Equipment (Grant No. 90406150007).

Xiangyu He is with the School of Mechano-electronic Engineering, Xidian University, Xi'an 710071, China. Xiangyu He is also with the School of Physics and Electronic Information, Luoyang Normal University, Luoyang 471934, China (e-mail: hexyming@163.com).

Guixi Liu is with the School of Mechano-electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: gxliu@xidian.edu.cn).

* Corresponding author.

time step includes a lot of clutter measurements, if all the measurements are used to compute the posterior multi-target density in the update step, more time will be consumed. In addition, at each time step, since one target can only generate one measurement and one measurement can only be assigned to one target, if each target generated measurement is used to update all the predicted multi-target densities, the tracks updated by the target generated measurements will include a large number of particles with very low weights, which will consume a lot of time to find particles with high weights in the resampling step.

Improving the computing speed of signal processing algorithm is an important research issue and has gained great attention [25-27], and it is this problem that we address in this paper. To accelerate the computation, a straight but efficient algorithm for the particle CBMeMBer filter is proposed in this paper. By using the predicted state estimations obtained in prediction step and measurement likelihoods computed in the update step, the proposed algorithm modifies the prediction and update equations. As shown in the numerical simulation, the proposed algorithm obtains significant improved computing speed and satisfactory estimation accuracy.

The rest of this paper is organized as follows. In Section Ⅱ a brief introduction to the kinematic model is provided. In addition, the particle CBMeMBer filter is presented and the existing drawback of the particle CBMeMBer filter is illustrated. The proposed algorithm is elaborated in Section Ⅲ. In Section Ⅳ, the simulated results are given and discussed. Finally, some meaningful conclusions are drawn in Section Ⅴ.

II. BACKGROUNDS

A. Kinematic model

The state equation and the measurement equation of a single target in two-dimensional plane are described by

$$x_k = F_{k-1}x_{k-1} + G_{k-1}q_{k-1} \tag{1}$$

$$z_k = H_k x_k + v_k \tag{2}$$

where x_k denotes the target state vector at time step k, F_{k-1} and G_{k-1} are the state transition matrix and the noise input matrix, respectively. z_k and H_k are the measurement vector and the observation matrix; q_{k-1} and v_k are the state noise and the observation noise, respectively.

B. Particle CBMeMBer filter

The particle CBMeMBer filter, which can accommodate nonlinear dynamic and measurement models, involves a prediction step, an update step and a resampling step that propagate the multi-target posterior probability density recursively in time [19]. The particle CBMeMBer filter is briefly described below.

Suppose that the parameters of the multi-Bernoulli RFS for birth targets at time step k are given by

$$\pi_{\Gamma,k} = \{(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}} \tag{3}$$

where $M_{\Gamma,k}$ denotes the number of birth tracks. $r_{\Gamma,k}^{(i)}$ and $p_{\Gamma,k}^{(i)}$ denote the existence probability and the probability density of the ith birth track, respectively.

Assume that the posterior multi-target density at time step $k-1$ is a multi-Bernoulli of the form

$$\pi_{k-1} = \{(r_{k-1}^{(i)}, p_{k-1}^{(i)})\}_{i=1}^{M_{k-1}} \tag{4}$$

where $r_{k-1}^{(i)}$ denotes the existence probability of the ith hypothesized track, M_{k-1} denotes the number of hypothesized tracks. $p_{k-1}^{(i)}$, which denotes the probability density of the ith hypothesized track, is comprised of a set of weighted particles, that is

$$p_{k-1}^{(i)}(x) = \sum_{j=1}^{L_{k-1}^{(i)}} w_{k-1}^{(i,j)} \delta(x - x_{k-1}^{(i,j)}) \tag{5}$$

where $L_{k-1}^{(i)}$ denotes the number of particles for the ith hypothesized track at time step $k-1$.

Then, the predicted multi-target density for targets to time step k is also a multi-Bernoulli and is given by

$$\pi_{k|k-1} = \{(r_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}} \cup \{(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}} \tag{6}$$

Assume that the predicted multi-target density at time step k is a multi-Bernoulli of the form

$$\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}} \tag{7}$$

where $p_{k|k-1}^{(i)}(x) = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \delta(x - x_{k|k-1}^{(i,j)})$.

Then, the updated multi-target density at time step k is also a multi-Bernoulli and is given by

$$\pi_k = \{(r_{L,k}^{(i)}, p_{L,k}^{(i)})\}_{i=1}^{M_{k|k-1}} \cup \{(r_{U,k}(z), p_{U,k}(\cdot;z))\}_{z\in Z_k} \tag{8}$$

where

$$r_{L,k}^{(i)} = r_{k|k-1}^{(i)} \frac{1-p_D}{1-p_D r_{k|k-1}^{(i)}} \tag{9}$$

$$p_{L,k}^{(i)}(x) = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \delta(x - x_{k|k-1}^{(i,j)}) \tag{10}$$

$$r_{U,k}(z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)}(1-r_{k|k-1}^{(i)})G_{k|k-1}^{(i)}(z)}{(1-p_D r_{k|k-1}^{(i)})^2}}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(i)} G_{k|k-1}^{(i)}(z)}{1-p_D r_{k|k-1}^{(i)}}} \tag{11}$$

$$G_{k|k-1}^{(i)}(z) = \sum_{j=1}^{L_{k|k-1}^{(i)}} p_D w_{k|k-1}^{(i,j)} g_k(z \mid x_{k|k-1}^{(i,j)}) \tag{12}$$

$$p_{U,k}(x;z) = \sum_{i=1}^{M_{k|k-1}} \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{U,k}^{(i,j)}(z) \delta(x - x_{k|k-1}^{(i,j)}) \tag{13}$$

$$w_{U,k}^{(i,j)}(z) = \frac{w_{k|k-1}^{(i,j)} \frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} g_k(z \mid x_{k|k-1}^{(i,j)})}{\sum_{i=1}^{M_{k|k-1}} \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} g_k(z \mid x_{k|k-1}^{(i,j)})} \tag{14}$$

where p_D is the detection probability, Z_k is the measurement set, $g_k(\cdot \mid x)$ is the single-target measurement likelihood, $\kappa_k(z)$ is the intensity of Poisson clutter.

C. Drawback of the particle CBMeMBer filter

Suppose that at time step k, the received measurement set is $Z_k = \{z_k^1, z_k^2, \cdots, z_k^{m_k}\}$, some measurements in Z_k may be clutter measurements, clutter measurements are spurious measurements that do not belong to any target. From (8), it can be seen that the computational complexity of computing the measurement-updated tracks is $O(|Z_k|)$, which can be very time-consuming for the update step when the clutter rate is high.

In addition, from (13), one can notice that the number of particles is $\sum_{i=1}^{M_{k|k-1}} L_{k|k-1}^{(i)}$ for each measurement-updated track. Note the particle set corresponding to the predicted posterior density $p_{k|k-1}^{(i)}(x)$ as $\{x_{k|k-1}^{(i,j)}\}_{j=1}^{L_{k|k-1}^{(i)}}$, since one target can only generate one measurement, if z is not the measurement originating from the ith target whose predicted state estimation is $m_{k|k-1}^{(i)}$, for $j = 1, \cdots, L_{k|k-1}^{(i)}$, the value of $g_k(z \mid x_{k|k-1}^{(i,j)})$ is very small, almost zero. According (14), the value of $w_{U,k}^{(i,j)}(z)$ is also almost zero. This can be viewed as that the measurements only contribute to the nearby particles. Therefore, for each $z \in Z_k$, if each target generated measurement is used to update all the predicted tracks, the updated posterior density $p_{U,k}(x;z)$ will involve a lot of particles with very low weights. This phenomenon can lead to a high time consumption on the particles with very low weights which are of insignificance to the filtering results in the resampling step.

III. THE PROPOSED ALGORITHM

To increase the computing speed of the particle CBMeMBer filter, we modify the CBMeMBer recursion equations by considering the predicted state estimations and measurement likelihoods. This is based on the fact that the measurements with high likelihoods are more likely to match with the particles and thus contribute significantly to the update of the posterior multi-target density, whereas the measurements with negligible likelihoods contribute quite less. Only the measurements with likelihoods greater than or equal to a threshold are used to update predicted multi-target density in the update step, hence, the proposed algorithm reduces unnecessary computation and accelerates the computing process. In the following, a high-speed algorithm for particle CBMeMBer is presented.

A. Prediction

Suppose that at time step $k-1$, the posterior multi-target density has the form of (4). Note the predicted state estimation for the ith surviving target as $m_{P,k|k-1}^{(i)}$, let $m_{\Gamma,k}^{(i)}$ denote the state of the ith birth target. Then, the modified prediction equation can be rewritten as

$$\begin{aligned}\pi_{k|k-1} = &\{(r_{P,k|k-1}^{(i)}, m_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}} \\ &\cup \{(r_{\Gamma,k}^{(i)}, m_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}}\end{aligned} \tag{15}$$

where

$$r_{P,k|k-1}^{(i)} = p_S r_{k-1}^{(i)} \tag{16}$$

$$p_{P,k|k-1}^{(i)}(x) = \sum_{j=1}^{L_{k-1}^{(i)}} w_{k-1}^{(i,j)} \delta(x - x_{P,k|k-1}^{(i,j)}) \tag{17}$$

$$x_{P,k|k-1}^{(i,j)} = F_{k-1} x_{k-1}^{(i,j)} + G_{k-1} q_{k-1} \tag{18}$$

$$p_{\Gamma,k}^{(i)}(x) = \sum_{j=1}^{L_{\Gamma,k}^{(i)}} \left(L_{\Gamma,k}^{(i)}\right)^{-1} \delta(x - x_{\Gamma,k}^{(i,j)}) \tag{19}$$

$$x_{\Gamma,k}^{(i,j)} \sim p_{\Gamma,k}^{(i)}(x), j = 1, \cdots L_{\Gamma,k}^{(i)} \tag{20}$$

where p_S is the survival probability, $L_{\Gamma,k}^{(i)}$ is the number of particles for the ith birth track.

B. Update

Given the predicted posterior multi-target density $\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, m_{k|k-1}^{(i)}, p_{k|k-1}^{(i)})\}_{i=1}^{M_{k|k-1}}$ and the received measurement set Z_k at time step k, then the update equation can be written as

$$\begin{aligned}\pi_k = &\{(r_{L,k}^{(i)}, p_{L,k}^{(i)})\}_{i=1}^{M_{k|k-1}} \cup \\ &\{(r_{U,k}^*(z), p_{U,k}^*(\cdot;z))\}_{z \in Z_k}\end{aligned} \tag{21}$$

where the legacy tracks are computed via (9-10).

For each $z \in Z_k$, one can compute the measurement likelihoods

$$C_k^{(i)}(z) = g(z \mid m_{k|k-1}^{(i)}), i = 1, \cdots, M_{k|k-1} \tag{22}$$

The measurements with likelihoods below a threshold contribute insignificantly to the particles, and thus they will not be used to compute the measurement-updated tracks in the proposed algorithm. The threshold used in the following equations is denoted as η. The threshold η should be empirically set according to the actual situation, with a larger value for a low clutter rate and a smaller value for a high clutter rate. If $\sum_{i=1}^{M_{k|k-1}} C_k^{(i)}(z) < \eta$, this phenomenon means that the measurement z whose projection onto the state space is far from all the predicted state estimations and thus contribute insignificantly to the update of the posterior multi-target density, then we can set $r_{U,k}^*(z) = 0$ and $p_{U,k}^*(\cdot;z) = []$ directly to accelerate the running speed of the update step, where [] is the Matlab notation for the null matrix.

If $\sum_{i=1}^{M_{k|k-1}} C_k^{(i)}(z) \geq \eta$, the parameters $r_{U,k}^*(z)$ and $p_{U,k}^*(\cdot;z)$ can be estimated as follows

$$r_{U,k}^*(z) = \frac{\displaystyle\sum_{i \in I = \{i | C_k^{(i)}(z) \geq \eta\}} \frac{r_{k|k-1}^{(i)}(1 - r_{k|k-1}^{(i)}) G_{k|k-1}^{(i)}(z)}{(1 - p_D r_{k|k-1}^{(i)})^2}}{\displaystyle\kappa_k(z) + \sum_{i \in I = \{i | C_k^{(i)}(z) \geq \eta\}} \frac{r_{k|k-1}^{(i)} G_{k|k-1}^{(i)}(z)}{1 - p_D r_{k|k-1}^{(i)}}} \tag{23}$$

$$p_{U,k}^*(x;z) = \sum_{i \in I = \{i | C_k^{(i)}(z) \geq \eta\}} \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{U,k}^{(i,j)} \delta(x - x_{k|k-1}^{(i,j)}) \tag{24}$$

$$\tilde{w}_{U,k}^{(i,j)}(z) = p_D w_{k|k-1}^{(i,j)} \frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} g_k(z \mid x_{k|k-1}^{(i,j)}), \quad i \in I = \{i \mid C_k^{(i)}(z) \geq \eta\} \tag{25}$$

$$w_{U,k}^{(i,j)}(z) = \frac{\tilde{w}_{U,k}^{(i,j)}(z)}{\sum_{i \in I=\{i \mid C_k^{(i)}(z) \geq \eta\}} \sum_{j=1}^{L_{k|k-1}^{(i)}} p_D w_{k|k-1}^{(i,j)} \frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} g_k(z \mid x_{k|k-1}^{(i,j)})} \tag{26}$$

From (24) and (25), one can observe that a lot of particles with very low weights can be discarded in the update step. Thus, a lot of unnecessary computation will be avoided in the resampling step.

C. Output result

Let $\hat{N}_k$ denote the estimated target number at time step k, then, $\hat{N}_k$ can be estimated as follows

$$\hat{N}_k = \mathrm{round}\left(\sum_{i=1}^{M_{k|k-1}} r_{L,k}^{(i)} + \sum_{z \in Z_k} r_{U,k}^{*}(z) \right) \tag{27}$$

where $\mathrm{round}(\cdot)$ denotes rounding operation.

Target state estimates can be obtained by computing $\hat{N}_k$ means of posterior densities with the highest existence probabilities.

In the next section, we will analyze the performance of the proposed algorithm compared with the particle CBMeMBer filter using the Monte Carlo (MC) simulations.

IV. Simulation Results

To verify the effectiveness of the proposed algorithm, we consider a two-dimensional scenario with an unknown and time varying number of targets observed in clutter. The target state vector $x_k = [\tilde{x}_k^T, \omega_k]^T$ is comprised of the planar position and velocity $\tilde{x}_k^T = [p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}]^T$ as well as the turn rate ω_k. The simulation environment was as follows: AMD A8-6600K APU with Radeon HD(tm) Graphics 3.9 GHz, 4 GB DDR3 1600 Memory, Windows 7, and MATLAB R2012a.

The target dynamic equations are described as

$$\tilde{x}_k = F(\omega_{k-1})\tilde{x}_{k-1} + G q_{k-1} \tag{28}$$

$$\omega_k = \omega_{k-1} + \Delta u_{k-1} \tag{29}$$

where

$$F(\omega) = \begin{bmatrix} 1 & \frac{\sin\omega\Delta}{\omega} & 0 & -\frac{1-\cos\omega\Delta}{\omega} \\ 0 & \cos\omega\Delta & 0 & -\sin\omega\Delta \\ 0 & \frac{1-\cos\omega\Delta}{\omega} & 1 & \frac{\sin\omega\Delta}{\omega} \\ 0 & \sin\omega\Delta & 0 & \cos\omega\Delta \end{bmatrix} \tag{30}$$

$$G = \begin{bmatrix} \Delta^2/2 & 0 \\ \Delta & 0 \\ 0 & \Delta^2/2 \\ 0 & \Delta \end{bmatrix} \tag{31}$$

$$q_{k-1} \sim N\left(0, \begin{bmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{bmatrix}\right) \tag{32}$$

$$u_{k-1} \sim N(0, 0.01^2) \tag{33}$$

For simplicity, we assume that the measurement is a noisy version of the position, and the measurement equation is described as

$$z_k = H_k \tilde{x}_k + v_k \tag{34}$$

where

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{35}$$

$$v_k \sim N\left(0, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix}\right) \tag{36}$$

The birth process is a multi-Bernoulli RFS with density $\pi_\Gamma = \{(r_\Gamma, p_\Gamma^{(i)})\}_{i=1}^{3}$, where $r_\Gamma = 0.03$, and

$$p_\Gamma^{(i)} = N\left(x; m_\gamma^{(i)}, P_\gamma\right), \quad i = 1,2,3 \tag{37}$$

where $m_\gamma^{(1)} = [220, 2.5, 120, -1]^T$, $m_\gamma^{(2)} = [180, 4, 160, 2]^T$, $m_\gamma^{(3)} = [120, 3.5, 220, 3]^T$, $P_\gamma = \mathrm{diag}([2,1,2,1])$, and $\mathrm{diag}(\cdot)$ denotes the diagonal matrix.

The survival probability is $p_S = 0.99$. The detection probability is $p_D = 0.98$. $\Delta = 1$ s is the sampling period. The clutter is modelled as a Poisson RFS with the mean $r = 6$ over the region [0, 300] × [0, 300] m^2. At each time step, a maximum of $L_{\max} = 1000$ and minimum of $L_{\min} = 300$ particles are imposed for each hypothesized track. In addition, measurement-updated tracks are pruned by discarding those with existence probabilities less than 10^{-3}, The threshold in the proposed algorithm is $\eta = 10^{-10}$. Multinomial resampling is employed for the simulation studies.

The filtering performance of the proposed algorithm is evaluated by using the optimal subpattern assignment (OSPA) distance [28] defined as

$$d_{OSPA}(X, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}\left(x_i, y_{\pi_i}\right)^p + c^p (n-m) \right) \right)^{\frac{1}{p}} \tag{38}$$

where the parameters are set to *p*=2 and *c*=50 m in our simulation.

In addition, to evaluate the real-time performance of our proposed algorithm, the real-time performance improvement (RTPI) [29] is used as follows

$$\mathrm{RTPI} = \frac{\bar{t} - \bar{t}_p}{\bar{t}} \times 100\% \tag{39}$$

where $\bar{t}$ and $\bar{t}_p$ denote the average computing time of one MC trial for the particle CBMeMBer filter and the proposed algorithm, respectively.

Fig.1 plots the individual x and y coordinates of the true target tracks and the measurements at each time step in the presence of the clutter, where the solid lines denote the true target tracks, and the plus signs denote the measurements.

Fig.2 shows the target position estimations superimposed on the true target tracks over 50 time steps. It can be seen that the proposed algorithm provides satisfactory filtering performance. Like the particle CBMeMBer filter, the proposed algorithm not only correctly tracks the individual target motions but also identifies the various target births and deaths throughout.

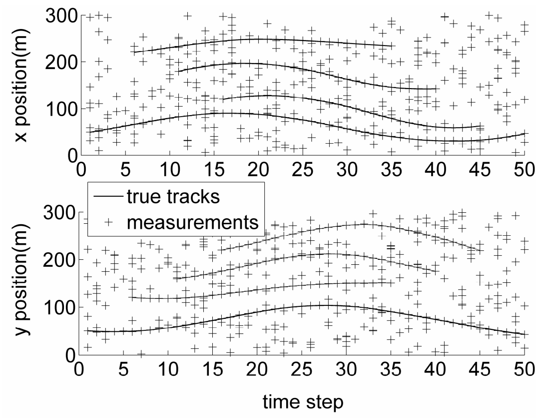

Fig. 1. True target tracks and measurements

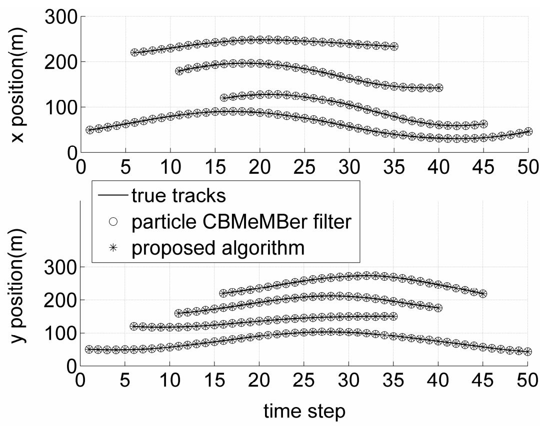

Fig. 2. True target tracks and position estimations

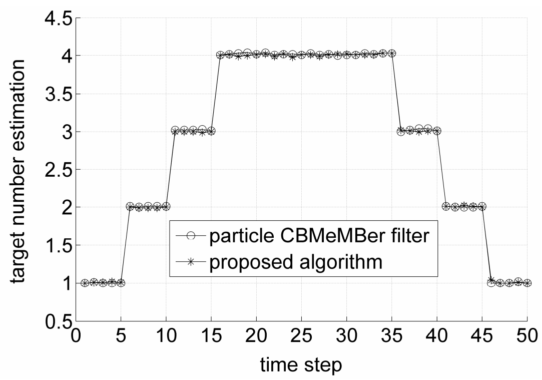

Fig. 3. Average target number estimations for different algorithms

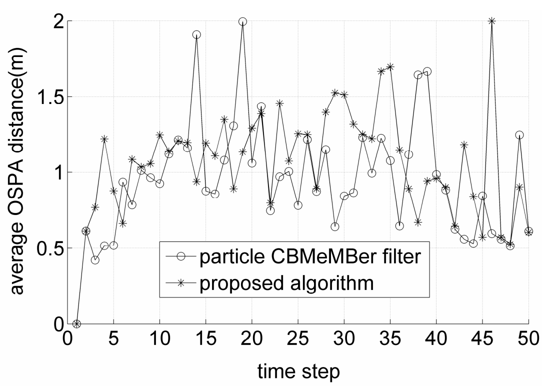

Fig. 4. Average OSPA distances for different algorithms

To obtain reliable results, 100 MC trials are performed for each algorithm on the same target tracks but with independently generated measurements. The filtering performances of different algorithms are compared in terms of the target number estimations and OSPA distances. In Fig. 3, the true target number at each time step is shown along with the average target number estimations for the particle CBMeMBer filter and the proposed algorithm. The results show that the target number estimation of the proposed algorithm is very similar to that of the particle CBMeMBer filter, and that both algorithms can provide reliable number estimations. For each time step, the average OSPA distances for the particle CBMeMBer filter and the proposed algorithm are shown in Fig. 4. The plots demonstrate that the estimation accuracy of the proposed algorithm is satisfactory, it also demonstrate that the filtering performance of the proposed algorithm does not degrade compared with particle CBMeMBer filter.

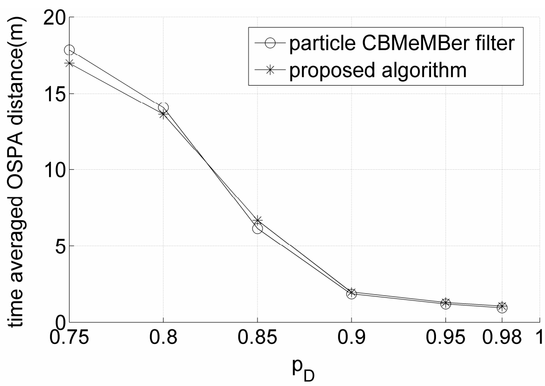

Fig. 5. Time averaged OSPA distances for different algorithms versus detection probability

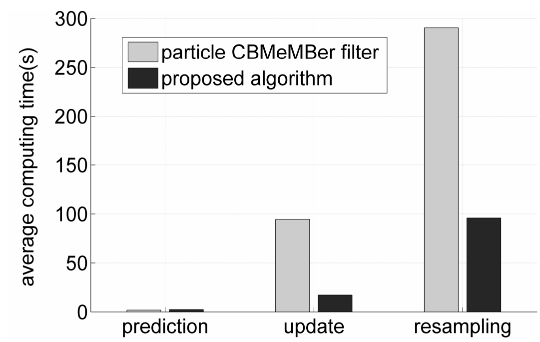

Fig. 6. Average computing time of prediction, update, and resampling for different algorithms

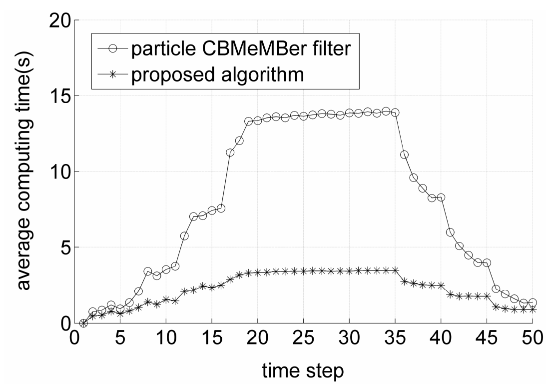

Fig. 7. Average computing time for different algorithms versus time step

The time averaged OSPA distances for various values of detection probability p_D are given in Fig. 5. As seen from

Fig.5, the time averaged OSPA distances for both filtering algorithms decrease as p_D increases. However, the gap between the particle CBMeMBer filter and the proposed algorithm is not noticeable.

One MC trial of both algorithms consists of three steps: prediction, update, and resampling. Fig. 6 shows the average computing time of the prediction, update, and resampling for the particle CBMeMBer filter and the proposed algorithm. As seen from Fig. 6, the update and resampling steps both consume much more time than the prediction step. Compared with the update and resampling steps, the average computing time of the prediction step is almost negligible. A comparison shows that the proposed algorithm outperforms the particle CBMeMBer filter in the aspect of the average computing time of the update and the resampling steps. Furthermore, the average computing time of each time step is shown in Fig. 7 for the particle CBMeMBer filter and the proposed algorithm. As seen, when clutter rate is constant, the average computing time of particle CBMeMBer filter grows linearly with the number of targets. It can also be seen that the proposed algorithm requires less time than the particle CBMeMBer filter to complete the calculation of each time step. Thus, the proposed algorithm can save a lot of time to complete one MC simulation.

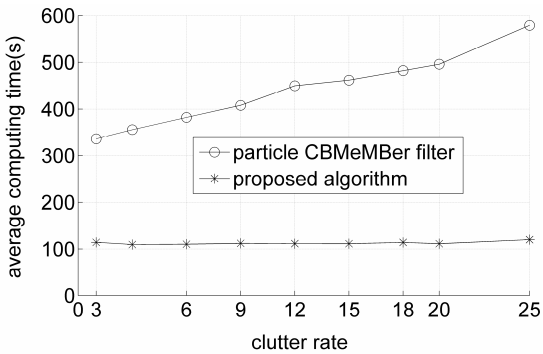

Fig. 8. Average computing time for different algorithms versus clutter rate

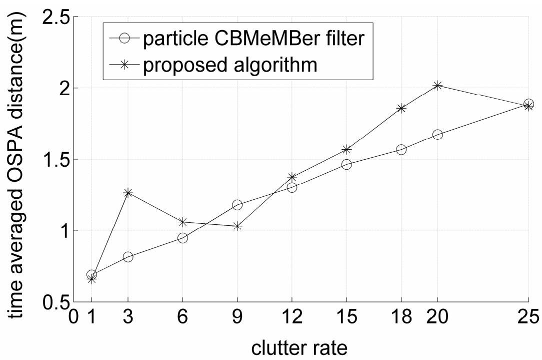

Fig. 9. Time averaged OSPA distances for different algorithms versus clutter rate

Also, 100 MC trials are performed for both algorithms over varying clutter rates to compare average performances in terms of the average computing time and time averaged OSPA distances, with the results shown in Figs. 8 and 9, respectively. As seen from Fig. 8, the average computing time of the proposed algorithm is much smaller than that of the particle CBMeMBer filter. In particular, the higher the clutter rate is the more the computing time can be saved. Moreover, it is observed from Fig. 9 that the estimation accuracy of the proposed algorithm is only slightly affected compared with the particle CBMeMBer filter. Therefore, we can conclude that the proposed algorithm can achieve similar tracking performance but with a much smaller computational cost comparing with the particle CBMeMBer filter.

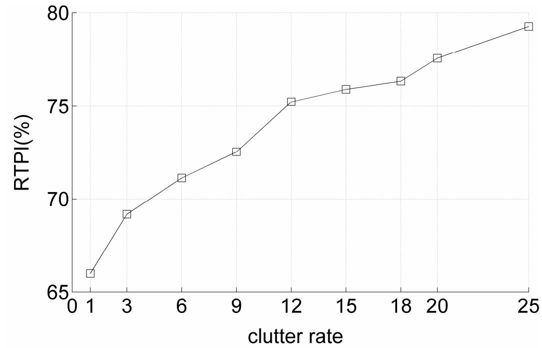

Fig. 10. RTPI against different clutter rates

In addition, the RTPI against clutter rates range from $r=1$ to 25 is shown in Fig. 10. As seen from Fig. 10, the proposed algorithm obtains more real-time performance improvement when clutter rate increases.

V. Conclusion

Based on the analysis of the drawback present in the particle CBMeMBer filter, a high-speed algorithm for particle CBMeMBer filter is proposed in this paper, which can improve the computing speed significantly by some modifications of particle CBMeMBer recursion equations. The idea behind the modification of recursion equations is to reduce unnecessary computation by only using the measurements with likelihoods greater than or equal to a threshold. Simulation results show that the proposed algorithm can achieve a much faster computing speed with satisfactory filtering performance as compared with the particle CBMeMBer filter.

References

[1] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
[2] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*. San Diego, CA: Academic Press, 1988.
[3] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.
[4] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451-460, Sep. 1975.
[5] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 19, no. 1, pp. 5-18, Jan. 2004.
[6] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843-854. Dec. 1979.
[7] I. Goodman, R. Mahler, and H. Nguyen, *Mathematics of Data Fusion*. Norwell, MA: Kluwer Academic Publishers, 1997.
[8] R.P.S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152-1178, Oct. 2003.
[9] B.N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. 6th Int. Conf. Information Fusion*, Cairns, 2003, pp. 792-799.
[10] B.N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224-1245, Oct. 2005.

[11] B.N. Vo and W.K. Ma, “A closed form solution for the probability hypothesis density filter,” in *Proc. 8th Int. Conf. Information Fusion*, Philadelphia, 2005, pp. 856-863.
[12] B.N. Vo and W.K. Ma, “The Gaussian mixture probability hypothesis density filter,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091-4104, Nov. 2006.
[13] D.E. Clark and J. Bell, “Convergence results for the particle PHD filter,” *IEEE Trans. Signal Process.*, vol. 54, no. 7, pp. 2652-2661, Jul. 2006.
[14] D. Clark and B.N. Vo, “Convergence Analysis of the Gaussian Mixture PHD Filter,” *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1204-1212. Apr. 2007.
[15] D. Clark, I.T. Ruiz, Y. Petillot, and J. Bell, “Particle PHD filter multiple target tracking in sonar image,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 409-416, Jan. 2007.
[16] E. Maggio, M. Taj, and A. Cavallaro, “Efficient multitarget visual tracking using random finite sets,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1016-1027, Aug. 2008.
[17] X. Zhou, H. Yu, H. Liu, and Y. Li, “Tracking multiple video targets with an improved GM-PHD tracker,” *Sensors*, vol. 15, pp. 30240-30260, Dec. 2015.
[18] F. Zhang, C. Buckl, and A. Knoll, “Multiple vehicle cooperative localization with spatial registration based on a probability hypothesis density filter,” *Sensors*, vol. 14, pp. 995-1009. Jan. 2014.
[19] B.T. Vo, B.N. Vo, and A. Cantoni, “The cardinality balanced multi-target multi-bernoulli filter and its implementations,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409-423, Feb. 2009.
[20] F. Lian, C. Li, and C. Han, “Convergence analysis for the SMC-MeMBer and SMC-CBMeMBer filters,” *J. App. Math.*, vol. 2012, pp. 1-25, 2012.
[21] J. Yang and H. Ge, “An improved multi-target tracking algorithm based on CBMeMBer filter and variational Bayesian approximation,” *Signal Process.*, vol. 93, no. 9, pp. 2510-2515, Sep. 2013.
[22] D. Dunne and T. Kirubarajan, “Multiple model multi-Bernoulli filters for manoeuvering targets,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2679-2692, Oct. 2013.
[23] J. Yang, H. Ji, and H. Ge, “Multi-model particle cardinality-balanced multi-target multi-Bernoulli algorithm for multiple manoeuvring target tracking,” *IET Radar Sonar Navig.*, vol. 7, no. 2, pp. 101-112, Feb. 2013.
[24] D. Li, C. Hou, and D. Yi, “Multi-Bernoulli smoother for multi-target tracking,” *Aerosp. Sci. Tech.*, vol. 48, pp. 234-245, Jan. 2016.
[25] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri, “Real-Time Intrusion Detection System Using Multi-agent System,” *IAENG International Journal of Computer Science*, vol. 43, no.1, pp80-90, 2016.
[26] T. Li, S. Sun, and T. Sattar, “High-speed Sigma-gating SMC-PHD filter,” *Signal Process*., vol. 93, no.9, pp. 2586-2593, Sep. 2013.
[27] Yanying Li, Youlong Yang, Xiaofeng Zhu, and Xiaoxia Dou, “Towards Fast and Efficient Algorithm for Learning Markov Blankets,” *IAENG International Journal of Computer Science*, vol. 42, no.1, pp17-22, 2015.
[28] D. Schuhmacher, B.T. Vo, and B.N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447-3457, Aug. 2008.
[29] Y. Zheng, Z. Shi, R. Lu, and X. Shen, “An efficient data-driven particle PHD filter for multitarget tracking,” *IEEE Trans. Industrial Informatics*, vol. 9, no. 4, pp. 2318-2326, Nov. 2013.