Progress In Electromagnetics Research, Vol. 136, 369–383, 2013

THRESHOLD-BASED RESAMPLING FOR HIGH-SPEED PARTICLE PHD FILTER

Zhiguo Shi[1], Yunmei Zheng[1], Xiaomeng Bian[2, *], and Zhengde Yu[3]

[1]Department of Information and Electronic Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China

[2]College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China

[3]Naval Unit No. 91656, Shanghai 200439, China

Abstract—In recent years, particle probability hypothesis density (PHD) filtering has become an active research topic for multiple targets tracking in dense clutter scenarios. However, it is highly required to improve the real-time performance of particle PHD filtering because it is a kind of Monte Carlo approach and the computational complexity is very high. One of major difficulties to improve the real-time performance of particle PHD filtering lies in that, resampling, which is usually a sequential process, is crucial to the fully-parallel implementation of particle PHD filter. To overcome this difficulty, this paper presents a novel threshold-based resampling scheme for the particle PHD filter, in which the particle weights are all set below a proper threshold. This specific threshold is determined using a distinguishing feature of the particle PHD filters: The weight sum of all particles in weight update is equal to the total target number in the current iteration. This proposed resampling scheme allows the use of fully-pipelined architecture in the hardware design of particle PHD filter. Theoretical analysis indicates that the particle PHD filter employing the proposed resampling technique can reduce the time complexity by 33% around in a typical multi-target tracking (MTT) scenario compared with that employing the traditional systematic resampling technique, while simulation results show that it can maintain the almost same performance of estimation accuracy.

Received 4 December 2012, Accepted 10 January 2013, Scheduled 20 January 2013

* Corresponding author: Xiaomeng Bian (xiaomeng.bian@ieee.org).

1. INTRODUCTION

In the research area of state estimation and target tracking [1–6], the technique of multiple target tracking (MTT) can estimate the target number as well as each target's state in the scene based on a sequence of uncertain measurements, where the "uncertain" mainly comes from detection uncertainty, association uncertainty and clutters in radar systems [7, 8]. Traditional MTT filters, such as multiple hypothesis tracking (MHT) filters, Markov chain Monte Carlo (MCMC) filters and joint probabilistic data association (JPDA) [9], mostly adopt the strategy of measurement-data association to existing tracks and then track each target with a separate filter based on the assumption that each target moves independently.

Recently a novel approach called probability hypothesis density (PHD) filter/cardinalized PHD (CPHD) filter has been widely used to avoid explicit association between measurements and (target) tracks [10–12]. The idea of PHD/CPHD filter is to represent the targets and the uncertain measurements as random finite sets (RFSs) and use finite set statistics (FISST) to solve MTT problems under the Bayesian framework. Furthermore, similar to the particle filter (PF) [13–19], the particle PHD/CPHD mainly consists of three steps, namely, *Generation of Particles* (*Prediction*), *Weight Computation* (*Update*) and *Resampling*. Based on our analysis in this paper, the resampling is a major bottleneck to increase the real-time performance of the particle PHD filter, where, due to its feature of sequential processing, the main difficulty lies in reducing the processing latency. That is, the traditional resampling schemes, such as systematic resampling (SR) and residual resampling (RR), have to start after all the particle weights are available [20]. Therefore, one of the major challenges for the real-time hardware implementation of particle PHD filters comes from the fact that it is difficult to implement pipeline architecture between *Weight Computation* and *Resampling*. Though CPHD, GM-PHD and MeMBer filter have their own advantages in terms of real-time performance, the particle PHD filter is a more general solution for nonlinear/non-Gaussian MTT problems in dense clutter scenarios. Therefore, we are interested in improving the real-time hardware design of particle PHD filters.

In this paper, to overcome the bottleneck in *Resampling*, we propose a novel threshold-based resampling for fully-pipelined high-speed particle PHD filter. The main contributions are summarized as follows.

- It is disclosed that the *Resampling* occupies a large portion of the processing delay in each iteration of the traditional particle PHD filter.

- We propose an efficient high-speed particle PHD filter, where the *Resampling* is carried out by the proposed threshold-based resampling scheme based on the distinguishing feature of the particle PHD filter that the weight sum of all the particles equals to the target number at each iteration.
- Theoretic analysis indicates that, in comparison with the traditional SR based particle PHD filters, a particle PHD filter employing the proposed resampling can reduce the processing time by 33% around in a typical MTT secnario, while simulation results show that it can maintain the same level of estimation accuracy.

The rest of this paper is organized as follows. Section 2 presents a systematic analysis of the time delay in the traditional particle PHD filter iterations, and proposes the threshold-based resampling along with the high-speed particle PHD filter, followed by the performance evaluation in Section 3. In Section 4 we discuss a key parameter a to determine the threshold T in the proposed high-speed particle PHD filter. Finally, conclusions are drawn in Section 5.

2. HIGH-SPEED PARTICLE PHD FILTER

2.1. Analysis of Time Delay in Hardware Implementation of Particle PHD Filter

As an approximate implementation of PHD filter, the particle PHD filter is a promising candidate for MTT problems [10, 21]. The diagram of operations in the implementation of particle PHD filter is depicted in Figure 1, where, $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$ denotes a particle approximation of PHD at time k, and "Data" denotes the input measurements received from the radar sensors. The initial particle samples can be obtained by two ways. One way is to get the initial particle samples by assuming that the targets distribute uniformly in the scene according to the targets' prior distribution, and the other is to get the initial particle samples by determining the target number and targets' states according to the input measurements. Once the initial samples are obtained, the particle PHD filter mainly contains three steps of processings, namely, *Prediction*, *Update* and *Resampling*, iteratively, and output the estimated particles from *Resampling* step to compute the estimated targets' states.

Intuitively, the particle PHD filter can be implemented in a fully-sequential way, and its timing of operations is shown in Figure 2, where $N_{predict}$, N_{update} and $N_{resampling}$ are the number of clock cycles required for *Prediction*, *Update* and *Resampling* in the particle PHD

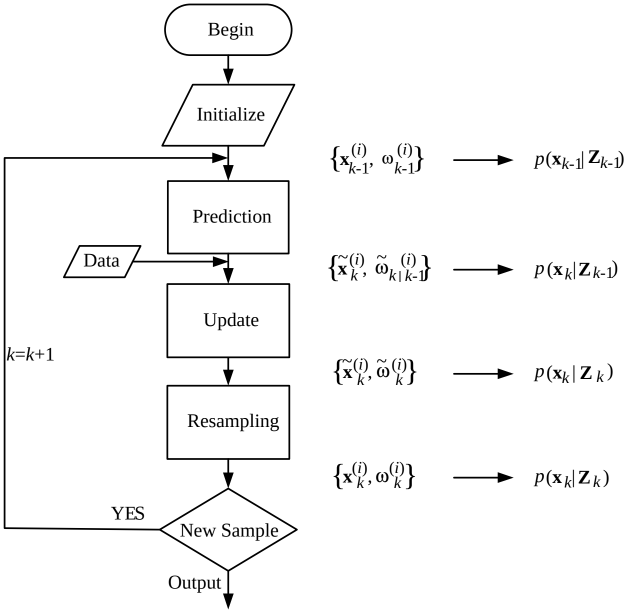

Figure 1. The diagram of operations of particle PHD filter.

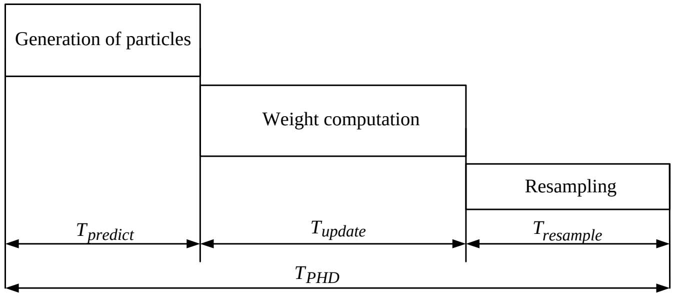

Figure 2. Timing of operations in the fully-sequential implementation of particle PHD filter.

iteration, respectively. Consequently, the overall processing time of one particle PHD iteration (e.g., the k-th iteration) is $T_{FS\text{-}PHD} = (N_{predict} + N_{update} + N_{resampling})T_{clk}$, where T_{clk} is the number of time cycles of system clock.

However, if we look insight into the operation of Figure 2, the *Prediction* step and the *Update* step can be pipelined naturally, because once a particle is generated from the *Prediction*, its weight can be updated in the next time step. The timing of operation of this implementation is shown in Figure 3. We term this implementation

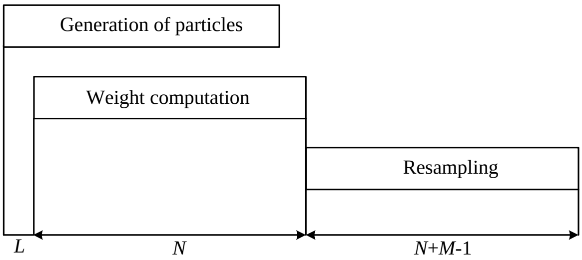

Figure 3. Timing of operations in the partially-pipelined implementation of particle PHD filter.

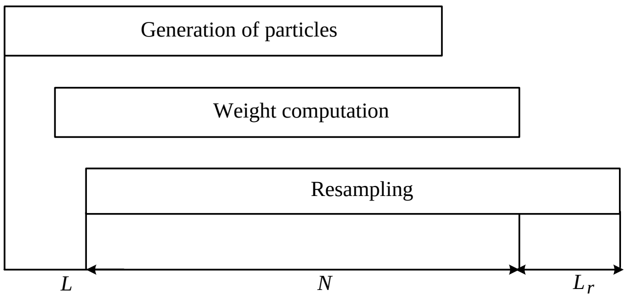

Figure 4. Timing of operations in the fully-pipelined implementation of particle PHD filter.

"partially-pipelined implementation".

In general, it is difficult to pipeline the *Update* and *Resampling* for the traditional PF and particle PHD filter due to their inherent sequential nature. In specific, resampling can not be started until the updated weights of all the particles are ready because the resampling is to keep the relatively more important particles and discard those relatively less important particles, and the relative importance can be decided only after all the particle weights are obtained. As seen in Figure 3, the partially-pipelined implementation with traditional SR has to occupy $N + M - 1$ clock cycles, that is, $T_{resampling} = (N + M - 1)T_{clk}$, where N, M are the particle numbers before resampling and after resampling, respectively. And the overall processing time of one particle PHD iteration is

$$\begin{aligned} T_{PP\text{-}PHD} &= (L + N + (N - M + 1))T_{clk} \\ &= (L + 2N - M + 1)T_{clk} \approx (2N - M)T_{clk} \end{aligned} \tag{1}$$

considering that L is much less than N. This is the case for particle PHD filter with most existing resampling scheme.

However, if a fully-pipelined particle PHD filter is achievable as shown in Figure 4, the overall processing time of one particle PHD iteration is

$$T_{FP\text{-}PHD} = (L + N + L_r)T_{clk} \approx NT_{clk} \quad (2)$$

considering that L and L_r are much smaller than N. Typically, we have $N = 2M$ for many multiple target tracking scenarios when using particle PHD filter, then from Eq. (1) and Eq. (2), we have $T_{PP\text{-}PHD} = 3MT_{clk}$ and $T_{FP\text{-}PHD} = 2MT_{clk}$. That means, the fully-pipelined particle PHD filter can save M clock cycles which is approximate 33% of the total cycle time of the partially-pipelined particle PHD filter. In other words, the fully-pipelined particle PHD filter is of great value in accelerating the processing of particle PHD filtering. In this paper, we will propose a threshold-based resampling scheme which can be adopted in the particle PHD iteration to achieve the fully-pipelined implementation.

Although some threshold-based resampling techniques have been proposed to improve the processing speed in particle filtering, none of them can implement fully-pipelined operations of the three main steps [22, 14, 23]. In particular, Ref. [22] presents a kind of threshold-based resampling to reduce the computation complexity and processing time, which is a partial resampling because only part of all particles are resampled. Ref. [14] proposes a compact threshold-based resampling and the corresponding efficient hardware implementation, where the threshold is set as S/N, with S representing the sum of all the updated weights and N representing the total particle number at each iteration. Although the clock cycles are only N, because the sum of all the updated weights still have to be obtained to determine the threshold for resampling, the bottleneck of the sequential nature of resampling has not been broken.

In this paper, to the best of our knowledge, it is the first time that a threshold-based resampling for the particle PHD filter is proposed, which can implement fully-pipelined particle PHD filter, as shown in Figure 4.

2.2. High-speed Particle PHD Filter

From the viewpoint of processing speed, most of the existing resampling approaches can start only when the weights of all the involved particles are ready, giving rise to the bottleneck in increasing the processing speed. To combat this, we make use of the distinguishing feature of the particle PHD filter that the weight sum of all the particles equals to the target number at each iteration, and propose a threshold-based resampling for the particle PHD filter [10].

In particular, the target number from previous weight update is used as an approximation of the weight sum in the current iteration, which is formulated as follows.

$$O_{k+1} \approx \sum_{i=1}^{L_k+J_k} \tilde{w}_k^{(i)} \tag{3}$$

where L_k and J_k denote the particle number for the survival targets and spontaneous birth targets, respectively.

Note that the above equation can be applied with the assumption that the target number does not change too abruptly, which is a reasonable assumption in most cases.

Therefore, the proposed threshold-based PHD resampling is described as in Algorithm 1,

Algorithm 1 The threshold-based resampling for particle PHD filter.

Initialize threshold:
  Set $T = S_{k-1}/(a \times N)$, $a > 1$;
Begin:
  $n = 0$;
  for $j = 1, \ldots, N$ **do**
    if $w_k^{(j)} > T$ **then**
      $n \leftarrow n + 1$;
      $\hat{\mathbf{x}}_k^{(n)} \leftarrow \mathbf{x}_k^{(j)}$;
    end if
  end for
  $r = 1$;
  for $j = 1, ..., M$ **do**
    Assign sample: $\tilde{\mathbf{x}}_k^{(j)} \leftarrow \hat{\mathbf{x}}_k^{(r)}$
    Assign weight: $\tilde{w}_k^{(j)} \leftarrow S_k/M$
    $r \leftarrow \mathbf{mod}(r, n) + 1$;
  end for

where $\mathbf{x}_k^{(j)}$ and $\tilde{\mathbf{x}}_k^{(j)}$ represent the particles before resampling and after resampling, respectively; $\hat{\mathbf{x}}_k^{(j)}$ represents the substantial particles; S_k represents the sum of all weights after weight update at time k; N and M represent the particle number before resampling and after resampling, respectively; a is the parameter in the threshold T, which satisfies $a > 1$.

The threshold should be smaller than S_k/N in theory. Otherwise, all particles will be discarded when they have equal weights, i.e., $w_k^{(j)} = S_k/N$ for $j = 1, \ldots, N$. Obviously, there is a trade-off

Algorithm 2 High-speed particle PHD filter.

Step 1: Prediction
for $i = 1, \ldots, L_{k-1}$ **do**
  sample $\mathbf{x}_k^i \sim q_k(\cdot|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  compute the predicted weights of survival particles:
$$w_{k|k-1}^i \leftarrow \frac{\phi_{k|k-1}(\mathbf{x}_k^i, \mathbf{x}_{k-1}^i)}{q_k(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} w_{k-1}^i$$
end for
for $i = L_{k-1} + 1, ..., L_{k-1} + J_k$ **do**
  sample $\mathbf{x}_k^i \sim p_k(\cdot|\mathbf{z}_k)$
  compute the predicted weights of spontaneous birth weights:
$$w_{k|k-1}^i \leftarrow \frac{1}{J_k} \frac{\gamma_k(\mathbf{x}_k^i)}{p_k(\mathbf{x}_k^i|\mathbf{z}_k)}$$
end for
Step 2: Update
for each $\mathbf{z} \in \mathbf{Z}_k$ **do**
  compute:
$$C_k(\mathbf{z}) \leftarrow \sum_{i=1}^{L_{k-1}+J_k} \psi_{k,\mathbf{z}}(\mathbf{x}_k^i) w_{k|k-1}^i$$
end for
for $i = 1, ..., L_{k-1} + J_k$ **do**
  update weights: $w_k^i \leftarrow \left[1 - P_D(\mathbf{x}_k^i) + \sum_{\mathbf{z} \in \mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}}(\mathbf{x}_k^i)}{\kappa_k(\mathbf{z}) + C_k(\mathbf{z})}\right] w_{k|k-1}^i$
end for
compute the sum of all weights: $S \leftarrow \sum w_k^j$
Step 3: Resampling
if $k = 1$ **then**
  Set $T = 1/(a \times N)$, $a > 1$;
else
  Set $T = S_p/(a \times N)$, $a > 1$;
end if
$n = 0$;
for $j = 1, \ldots, N$ **do**
  if $w_k^{(j)} > T$ **then**
    $n \leftarrow n + 1$;
    $\hat{\mathbf{x}}_k^{(n)} \leftarrow \mathbf{x}_k^{(j)}$;
  end if
end for
$r = 1$;
for $j = 1, ..., M$ **do**
  Assign sample: $\tilde{\mathbf{x}}_k^{(j)} \leftarrow \hat{\mathbf{x}}_k^{(r)}$
  Assign weight: $\tilde{w}_k^{(j)} \leftarrow S/M$
  $r \leftarrow \mathbf{mod}(r, n) + 1$;
end for
$S_p = S$;
Step 4: Target estimation
estimate the target number: $N_k \leftarrow \mathbf{round}(N_{k|k})$;
use clustering algorithm to determine the N_k peaks of the posterior, which are the estimates of target states.

in determining the threshold T. Too large a value would discard some of the substantial particles while too small a value would select some of the negligible particles. In practice, if the threshold is determined according to the sum of all weights S_k at the time k, then the resampling can not be started until all the particle weights are updated. Therefore, conditioned on that the target number does not vary abruptly with time in subsequent iteration and the particle PHD filter has the distinguishing feature that the weight sum of all the particles equals to the target number at each iteration, the weight sum at the previous iteration S_{k-1} can be used to approximate S_k, that is, $T < S_{k-1}/N$ or $T = S_{k-1}/(a \times N)$. The selection of parameter a will be discussed in Section 5.

With the above threshold-based resampling scheme, the proposed high-speed particle PHD filter is described in Algorithm 2.

3. TRACKING PERFORMANCE EVALUATION

3.1. Target Tracking Model

To evaluate the overall processing efficiency of the proposed high-speed particle PHD filter, the following multi-target scenario is used for illustrating the tracking performance of the proposed algorithm. The dynamics equation of the targets is the following:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \Delta T^2/2 & 0 \\ 0 & \Delta T^2/2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} \mathbf{w}_k \tag{4}$$

where $\mathbf{x}_k = [x \;\; v_x \;\; y \;\; v_y]_k^T$ is target state vector at time $k\Delta T$ (k is the time index, and ΔT is the sampling interval), $\mathbf{w}_k = [w_{x,k} \;\; w_{y,k}]^T$ is the vector of independent zero-mean Gaussian white noise with $\mathbf{w}_k \sim N(0, Q(k))$.

The measurements originate from either targets or clutters. The target-originated measurement equations are

$$r_k = \left\| \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} x_s \\ y_s \end{bmatrix} \right\| + \nu_{1,k} \tag{5}$$

$$\theta_k = \arctan\left(\frac{[\; 0 \;\; 0 \;\; 1 \;\; 0 \;]\; \mathbf{x}_k - y_s}{[\; 1 \;\; 0 \;\; 0 \;\; 0 \;]\; \mathbf{x}_k - x_s} \right) + \nu_{2,k} \tag{6}$$

where $\nu_k = (\nu_{1,k}, \, \nu_{2,k})$ is the vector of independent zero-mean Gaussian measurement noises with standard deviations.

The probability of target survival is $e_{k|k-1}(\cdot) = 0.95$ and no spawning is considered here. Each spontaneous birth target has an

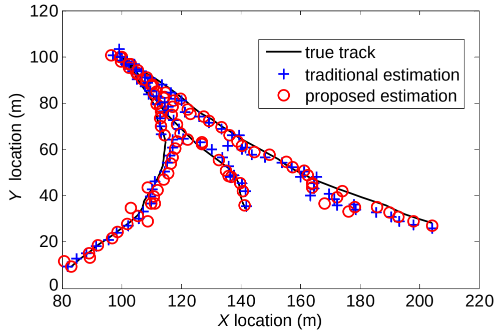

Figure 5. Estimated trajectories and true trajectories.

initial Gaussian distribution with mean and covariance

$$\bar{\mathbf{x}} = [100\,\mathrm{m} \quad 3\,\mathrm{m/s} \quad 100\,\mathrm{m} \quad -3\,\mathrm{m/s}]^T \tag{7}$$

$$Q_x = \mathrm{diag}\left(\left[10\,\mathrm{m}^2 \quad 1\,(\mathrm{m/s})^2 \quad 10\,\mathrm{m}^2 \quad 1\,(\mathrm{m/s})^2\right]\right) \tag{8}$$

The number of spontaneous birth targets has a Possion distribution with an average rate of 0.2 target per scan. For simulation parameters, the sampling interval is $\Delta T = 1$; the process noises are $\sigma_{w_{x,k}} = 0.8$ and $\sigma_{w_{y,k}} = 0.08$, respectively; the probability of detection is $P_D(\mathbf{x}_k) = 0.98$; the measurement noises are $\sigma_{\nu_{1,k}} = 2.5$ and $\sigma_{\nu_{2,k}} = 0.005$. Clutters are uniformly distributed over a $300\,\mathrm{m} \times 100\,\mathrm{m}$ rectangle region. The number of clutter points per scan has a Poisson distribution with an average rate of $r = 10$. The number of all particles are 2048, wherein $L_k = 1024$ particles are for survival targets and the other $J_k = 1024$ particles are for spontaneous birth targets. The parameter a for the proposed threshold-based resampling is set to 2.

To evaluate the error between the estimated and the true state, the Optimal Sub-Pattern Assignment (OSPA) is adopted as a multi-target miss-distance metric, which is described in detail in [24]. Compared with the former Wasserstein distance metric [25], the OSPA distance can jointly capture the differences in cardinality and individual elements between two finite sets in a mathematically consistent and intuitively meaningful way. According to Ref. [24], the parameters $p = 2$ and $c = 100$ are used in our simulation.

3.2. Simulation Results

The simulated true tracks of three targets over 40 scans, as well as the estimated trajectories, are plotted in Figure 5. Correspondingly, Figure 6 depicts the tendency of the x- and the y-components of the

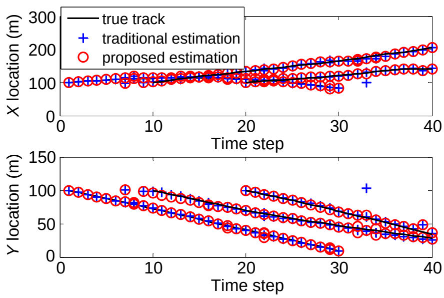

Figure 6. Estimated positions and true positions in the x and y directions.

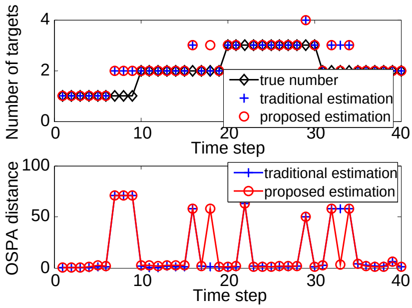

Figure 7. Matlab simulation of the proposed particle PHD filter. (a) Estimated target number versus the true target number. (b) Multitarget miss distance.

tracks vs time, respectively. For comparison, the tracking results using the traditional particle PHD filter are also presented. It is seen that the estimated positions based on both the traditional particle PHD filter and the proposed particle PHD filter are similar and they are all close to the true tracks at a single Monte Carlo (MC) trial.

Figure 7 plots the estimated target number against the ground truth and the OSPA multi-target miss distance at each time instant. It can be seen that the proposed high-speed particle PHD filter has the similar tracking performance as the traditional particle PHD filter. Also it is noticed that for both the propose PHD filter and the traditional one, the multi-target miss distance exhibits peaks at the instances where the estimated target number is incorrect. When the estimated number is correct, the OSPA miss distance is relatively small.

Figure 8 presents a set of OSPA distances averaged over 5000 MC runs for $p = 2$ and $c = 100$, which provides a natural and intuitive interpretation of the OSPA metric in terms of localization and cardinality errors. From this figure, it is clear that the high-speed particle PHD filter achieves the similar performance to the traditional particle PHD filter. Note that in terms of cardinality error, the OSPA distance demonstrates a small pulse once the target number changes.

Intuitively, the tracking performance will be degraded due to the threshold difference between the approximation and the original one when the target number changes in two successive time index. However, note that although the idea of proposed resampling is to keep

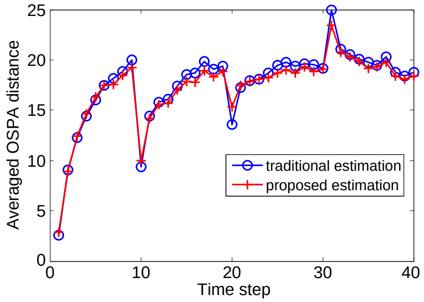

Figure 8. 5000 MC averaged OSPA distance versus time.

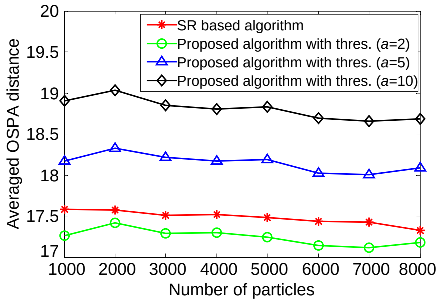

Figure 9. 5000 MC averaged OSPA distance versus number of particles.

the relatively more important particles and discard those relatively less important particles according to the threshold, the particles and the corresponding weights other than the threshold is directly related to the tracking performance. This makes the simulation results above reasonable. Besides, the occasions that the target number changes do not happen very frequently.

4. DISCUSSIONS

In this section, we further discuss how to determine the threshold T, or more specifically, the parameter a. To compare the proposed resampling with the traditional SR for the particle PHD filter, a simulation-based experiment on the averaged OSPA distance based 5000 MC trial is carried out. The results are shown in Figure 9, where tracks with different initial conditions are simulated, and we can have the following inferences.

- For a certain particle number, e.g., $N = 1000$, it is found that the particle PHD filters with a larger threshold corresponding to a smaller parameter a has a smaller averaged OSPA distance, and thus a better tracking accuracy.
- In the traditional particle PHD filter with SR as shown in Figure 9, the particle number characterizing the multi-target posterior probability density is determined from a trade-off between the tracking accuracy and the computation cost. That is, a larger particle number leads to higher estimation accuracy, which requires more computation cost at the same time.
- However, for the proposed threshold-based resampling based

particle PHD filter, the meaning of the threshold $T = S_{k-1}/(a \times N)$ are two-folds. On one hand, for a fixed value of parameter a, a larger particle number N means a lower threshold, which will degrade the resampling accuracy, and consequently affect the overall filtering accuracy. On the other hand, for a fixed value of threshold T, a larger particle number leads to higher estimation accuracy. So the expected tracking accuracy of the proposed particle PHD filter is determined by the combination of a and N.

Hence, to obtain better tracking accuracy by increasing the particle number, we can adopt a strategy of keeping the threshold constant for a reasonable value. For example, for one case that $a = 10$ and $N = 1000$, the averaged OSPA distance is 18.9059. If we want to obtain better tracking accuracy by doubling the particle number with $N = 2000$, the value of parameter a can be halved at the same time, and the consequent averaged OSPA distance is 18.3245. Obviously, the latter case outperforms the former case in terms of the tracking accuracy.

5. CONCLUSION

In this paper, we have proposed a novel threshold-based resampling scheme for high-speed particle PHD filtering, where the threshold is determined according to the target number estimated in the previous iteration. It has been shown that the particle PHD filter with the proposed resampling is able to fully pipeline the *Prediction*, *Update* and *Resampling*, and thus significantly reduce the processing latencies of the particle PHD filter. Meanwhile, the particle PHD filter with the proposed resampling scheme has similar tracking accuracy as the traditional particle PHD filter.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61171149), Zhejiang Province Commonweal Technique Research Project (No. 2010C31069) and the Research Foundation of State Key Laboratory of Industrial Control Technology (No. ICT1119).

REFERENCES

1. Liu, H. Q. and H. C. So, "Target tracking with line-of-sight identification in sensor networks under unknown measurement

noises," *Progress In Electromagnetics Research*, Vol. 97, 373–389, 2009.

2. Chang, Y., C. Chiang, and K. Chen, "SAR image simulation with application to target recognition," *Progress In Electromagnetics Research*, Vol. 119, 35–57, 2011.
3. Fan, L., X. Zhang, and L. Wei, "Tbd algorithm based on improved randomized hough transform for dim target detection," *Progress In Electromagnetics Research C*, Vol. 31, 271–285, 2012.
4. Lee, J., S. Cho, S. Park, and K. Kim, "Performance analysis of radar target recognition using natural frequency: Frequency domain approach," *Progress In Electromagnetics Research*, Vol. 132, 315–345, 2012.
5. Diao, W., X. Mao, H. Zheng, Y. Xue, and V. Gui, "Image sequence measures for automatic target tracking," *Progress In Electromagnetics Research*, Vol. 130, 447–472, 2012.
6. Tugac, S. and M. Efe, "Radar target detection using hidden Markov models," *Progress In Electromagnetics Research B*, Vol. 44, 241–259, 2012.
7. Zhang, Z. and J. Zhou, "A novel LPI method of radar's energy control," *Progress In Electromagnetics Research C*, Vol. 33, 81–94, 2012.
8. Fouda, A. and F. Teixeira, "Imaging and tracking of targets in clutter using differential time-reversal techniques," *Waves in Random and Complex Media*, Vol. 22, No. 1, 66–108, 2012.
9. Bar-Shalom, Y., T. Kirubarajan, and X. R. Li, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software,* Wiley, New York, 2001.
10. Mahler, R., "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39, No. 4, 1152–1178, 2003.
11. Ristic, B., D. Clark, B. N. Vo, and B. T. Vo, "Adaptive target birth intensity for phd and cphd filters," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 2, 1656–1668, 2012.
12. Hong, S. H., L. Wang, Z. G. Shi, and K. S. Chen, "Simplified particle PHD filter for multiple target tracking: Algorithm and architecture," *Progress In Electromagnetics Research*, Vol. 120, 481–498, 2011.
13. Wang, X. F., J. F. Chen, Z. G. Shi, and K. S. Chen, "Fuzzy-control based particle filter for maneuvering target tracking," *Progress In Electromagnetics Research*, Vol. 118, 1–15, 2011.

14. Hong, S. H., Z. G. Shi, J. M. Chen, and K. S. Chen, "A low-power memory-efficient resampling architecture for particle filters," *Circuits, Systems and Signal Processing*, Vol. 29, No. 1, 155–167, 2010.
15. Li, Y., Y. J. Gu, Z. G. Shi, and K. S. Chen, "Robust adaptive beamforming based on particle filter with noise unknown," *Progress In Electromagnetics Research*, Vol. 90, 151–169, 2009.
16. Chen, J. F., Z. G. Shi, S. H. Hong, and K. S. Chen, "Grey prediction based particle filter for maneuvering target tracking," *Progress In Electromagnetics Research*, Vol. 93, 237–254, 2009.
17. Zheng, N., Y. Pan, X. Yan, and R. Huan, "Local weight mean comparison scheme and architecture for high-speed particle filters," *Electronic Letters*, Vol. 47, No. 2, 2011.
18. Wang, Q., J. Li, M. Zhang, and C. Yang, "H-infinity filter based particle filter for maneuvering target tracking," *Progress In Electromagnetics Research B*, Vol. 30, 103–116, 2011.
19. Miao, L., J. Zhang, C. Chakrabarti, and A. Papandreou-Suppappola, "Algorithm and parallel implementation of particle filtering and its use in waveform-agile sensing," *Journal of Signal Processing Systems*, Vol. 65, No. 2, 211–227, 2011.
20. Bolic, M., P. M. Djuric, and S. Hong, "New resampling algorithms for particle filters," *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP'03)*, Vol. 2, 589–592, 2003.
21. Fan, J., Y. Zhu, S. Fan, H. Fan, and Q. Fu, "Feature aided switching model set approach for maneuvering target tracking," *Progress In Electromagnetics Research B*, Vol. 45, 251–268, 2012.
22. Bolic, M., P. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *Eurasip J. Appl. Signal Process.*, Vol. 15, 2267–2277, 2004.
23. Wang, J., H. Wang, Y. Qin, and Z. Zhuang, "Efficient adaptive detection threshold optimization for tracking maneuvering targets in clutter," *Progress In Electromagnetics Research B*, Vol. 41, 357–375, 2012.
24. Schuhmacher, D., B. T. Vo, and B. N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, Vol. 56, 3447–3457, 2008.
25. Hoffman, J. and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 34, No. 3, 327–336, 2004.