# An Efficient Data-Driven Particle PHD Filter for Multi-Target Tracking

Yunmei Zheng, *Student Member, IEEE,* Zhiguo Shi, *Member, IEEE,* Rongxing Lu, *Member, IEEE,* Shaohua Hong, *Member, IEEE,* and Xuemin (Sherman) Shen, *Fellow, IEEE.*

***Abstract*—In this paper, we propose an efficient data-driven particle PHD filter for real-time multi-target tracking of nonlinear/non-Gaussian system in dense clutter environment. In specific, the input measurements are first classified into two sets, namely survival measurements and spontaneous birth measurements, after eliminating clutters by using existing historic state data of targets. Since most clutters do not participate in the complex weight computation of particle PHD filter, better real-time performance can be achieved. The tracking performance is also improved because the survival measurements are used for survival targets and the spontaneous birth measurements are used for spontaneous birth targets, resulting in less interference from each other and from clutters. Extensive simulations validate the improvement of both the real-time performance and tracking performance of the proposed data-driven particle PHD filter in comparison with the traditional particle PHD filter.**

***Index Terms*—Data-Driven Mechanism, Particle PHD Filter, Real Time Performance, Tracking Performance**

## I. Introduction

MULTIPLE target tracking (MTT) is a very important technology for many industrial applications, such as automated surveillance [1], wireless sensor networks [2], [3], [4], mobile robots [5], traffic monitoring [6], etc. Recently, the so-called Probability Hypothesis Density (PHD) filter and Cardinalized PHD (CPHD) filter which avoid explicit associations between measurements and targets have been widely studied for MTT problems. The idea of PHD/CPHD filter is to represent the targets and measurements as Random Finite Sets (RFSs) and use finite set statistics (FISST) to solve MTT problems under Bayesian framework.

For the PHD filter [7], it propagates the intensity of the RFS of states in time, the advantage of which is that it operates only on the single-target state space and completely avoids any data association computation. For the CPHD filter [8], [9], it propagates the intensity of the RFS and the entire probability distribution of the target number in time, which relaxes the Poisson distribution assumption on the number of targets in the PHD filter at the cost of much higher computational complexity than that of the PHD filter [10]. From implementation perspective, a full sequential Monte Carlo (SMC) implementation of PHD filter, also called particle PHD filter, was proposed in [11], and closed form solutions to the PHD/CPHD recursions were derived for linear Gaussian multi-target models in [12] and [13], respectively.

In addition to the PHD/CPHD filter, Mahler has proposed the Multi-Target Multi-Bernoulli (MeMBer) recursion as a tractable approximation to the Bayes multi-target recursion under low clutter density scenarios [14]. Unlike the PHD/CPHD recursions, the MeMBer recursion propagates (approximately) the multi-target posterior density, and it allows reliable and inexpensive extraction of state estimates without clustering in the PHD/CPHD filter.

The demands of "real-time" MTT have been increasing [15], [16], [17], [18], [19], [20]. Since the CPHD filter propagates both the intensity of the RFS and the posterior cardinality distribution [13], its real-time characteristic is intrinsically not as good as the PHD filter. Although GM-PHD/CPHD has closed-form solution which makes it easy for real-time implementation, its application scenario is constrained to linear Gaussian system. When comparing the PHD filter with the MeMBer filter, according to the modeling assumptions on the PHD filter [11] and the MeMBer filter [21], the PHD filter is more suitable for denser clutter environments. Thus, we consider particle PHD filter as a good candidate for nonlinear/non-Gaussian MTT problems in dense clutter scenarios with high real-time requirements. However, since the particle PHD filter is a kind of Sequential Monte Carlo approach, its computational complexity is very high. Therefore, we are interested in improving the real-time performance of the particle PHD filter.

Similar to the particle filter [22], [23], [24], the particle PHD filter mainly consists of three steps, namely, Generation of Particles (Prediction), Weight Computation (Update) and Resampling (Resample). Specifically, Weight Computation consists of a mass of complicated mathematical computations and forms a major bottleneck of the traditional particle PHD processing. Since the measurements act as the only input in the particle PHD filter, it is possible to use data-driven approach to accelerate the filtering speed. In this paper, we propose an efficient data-driven particle PHD filter, where the novelty lies in the way of employing the data-driven mechanism in particle PHD filtering, to distinguish the survival measurements, spontaneous birth measurements and clutters for Weight Computation. The main contributions of this paper are summarized as follows.



Y. Zheng and Z. Shi are with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, 310027, China. Emails: {zhengyunmei, shizg}@zju.edu.cn

R. Lu and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. Emails: {rxlu, xshen}@bbcr.uwaterloo.ca

S. Hong is with the Department of Communication Engineering, Xiamen University, Fujian, 361005, China. Email: hongsh@xmu.edu.cn

- First, we present a systematic analysis on the time delay of the traditional particle PHD filter iteration. It is found that the participation of clutters in Weight Computation not only causes significant computation consumption but also incurs tracking performance degradation. In addition, we also analyze the computational complexity of the proposed particle PHD filter.
- Second, we propose an efficient data-driven particle PHD filter where all input measurements are classified into two categories of survival measurements and spontaneous birth measurements respectively after eliminating the participation of clutters. Both the real-time performance and the tracking performance are improved, in comparison with the traditional particle PHD filter.
- Third, extensive simulations show that the data-driven particle PHD filter has much better real-time performance and tracking performance in a progressive way.

The remainder of this paper is organized as follows. Section II presents a systematic time delay analysis of traditional particle PHD filter iteration. In Section III, we present our data-driven particle PHD filter, followed by the performance evaluation in Section IV. We also discuss the real-time performance improvement and tracking performance improvement under different clutter density environments in Section V. Finally, we draw our conclusions in Section VI.

## II. Time Delay Analysis of Particle PHD Filter

### *A. Particle PHD Filter* [11]

As an approximate implementation of PHD filter, particle PHD filter is considered as a promising filter for MTT problems.

For any $k \geq 1$, let $L_k$ and $J_k$ denote the number of survival particles and spontaneous birth particles at time $k$, respectively. Let $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{L_k}$ denote a particle approximation of PHD at time $k$. The traditional particle PHD filter procedure can be described as following three steps which are conducted in an iterative way, where details of parameter notation can be referred to [11].

Step 1: Prediction

For $i = 1, ..., L_{k-1}$, sample $\tilde{\mathbf{x}}_k^{(i)} \sim q_k(\cdot|\tilde{\xi}_{k-1}^{(i)}, \mathbf{Z}_k)$ and compute the weights of survival particles,

$$\tilde{w}_{k|k-1}^{(i)} = \frac{\phi_{k|k-1}(\tilde{\mathbf{x}}_k^{(i)}, \mathbf{x}_{k-1}^{(i)})}{q_k(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)} w_{k-1}^{(i)} \tag{1}$$

For $i = L_{k-1}+1, ..., L_{k-1}+J_k$, sample $\tilde{\mathbf{x}}_k^i \sim p_k(\cdot|\mathbf{Z}_k)$ and compute the weights of spontaneous birth particles,

$$\tilde{w}_{k|k-1}^{(i)} = \frac{1}{J_k} \frac{\gamma_k(\tilde{\mathbf{x}}_k^{(i)})}{p_k(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{Z}_k)} \tag{2}$$

Step 2: Update

For each $\mathbf{z} \in \mathbf{Z}_k$, compute

$$C_k(\mathbf{z}) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{k,\mathbf{z}}(\tilde{\mathbf{x}}_k^{(j)}) \tilde{w}_{k|k-1}^{(j)} \tag{3}$$

For $i = 1, ..., L_{k-1}+J_k$, update weights

$$\tilde{w}_k^{(i)} = \left[1 - P_D + \sum_{\mathbf{z}\in\mathbf{Z}_k} \frac{\psi_{k,\mathbf{z}}(\tilde{\mathbf{x}}_k^{(i)})}{\kappa_k(\mathbf{z}) + C_k(\mathbf{z})}\right] \tilde{w}_{k|k-1}^{(i)} \tag{4}$$

Step 3: Resample

Compute the total mass $\tilde{N}_k = \sum_{i=1}^{L_{k-1}+J_k} \tilde{w}_k^{(i)}$ and estimate the number of targets $\hat{N}_k = \text{round}(\tilde{N}_k)$, resample $\{\tilde{\mathbf{x}}_k^{(i)}, \tilde{w}_k^{(i)}/\tilde{N}_k\}_{i=1}^{L_{k-1}+J_k}$ to get $\{\mathbf{x}_k^{(i)}, w_k^{(i)}/\tilde{N}_k\}_{i=1}^{L_k}$, and rescale (multiply) the weights by $\tilde{N}_k$ to get $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{L_k}$.

### *B. Analysis of Time Delay*

Fig. 1 shows the timing of operations for one particle PHD filter iteration. In the figure, $N_{\text{predict}}$, $N_{\text{update}}$ and $N_{\text{resample}}$ are the number of cycles required for Generation of Particles, Weight Computation and Resampling in the particle PHD iteration, respectively. The total cycle time of one particle PHD iteration is then $T_{\text{PHD}} = (N_{\text{predict}} + N_{\text{update}} + N_{\text{resample}})T_{\text{clk}}$, where $T_{\text{clk}}$ is the system clock.

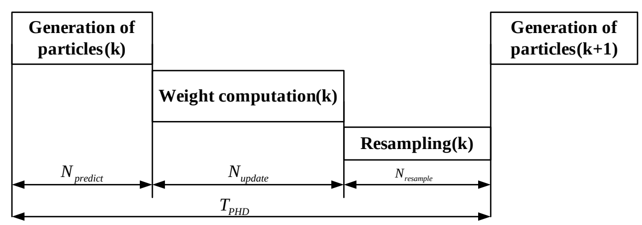


Fig. 1. Timing of operations in the traditional particle PHD filter

From Fig. 1, both the Weight Computation and the Resampling are the bottlenecks in the particle PHD iteration. Since the traditional resampling can not be pipelined with other operations due to its sequential nature, it significantly occupies the processing time. Hence, the development of faster and more efficient resampling algorithms is vital to the implementation of the particle PHD filter in high-speed applications. Since it is a distinguishing feature of all PHD filters that the total weight of particles equals the total number of targets at each Weight Computation, it is possible to develop a threshold-based resampling mechanism to break the bottleneck of the sequential nature of traditional resamplings.

For the bottleneck of the PHD Weight Computation, the complicated computations in it limit the processing. The Weight Computation is separated into three sub-steps. First, the likelihood function between the position of each particle and each input measurement is computed. Second, the product accumulation between the likelihoods and the predicted weights is computed to get the Eq. (3). Third, update the weights of all predicted particles to get the updated weights according to the Eq. (4). In other word, the Weight Computation step consists of a series of complex mathematical computations including multiplication, division and accumulation, and so forth. The computation time of each sub-step depends on both the number of all particles and the number of all input measurements.

Since the particle PHD filter exploits the idea of particle filter as a feasible solution to solve PHD filtering, the tracking performance of the particle PHD filter is proportional to the number of particles that is used to characterize the multi-target posterior probability. However, the real-time performance of the filter is inversely proportional to the number of particles used. Therefore, there is a trade-off between the tracking performance and the processing time by choosing an appropriate numerical value of the number of particles before the particle PHD filter is used to tracking multiple targets, i.e., the proper values of $L_k$ and $J_k$ should be determined. We introduce the criterion "lost tracking ratio" for determining the number of particles, which is exactly defined as the ratio of the number of tracking lost runs and the total number of Monte Carlo (MC) simulation runs. Herein, one tracking lost run means, in a single MC simulation run, the number of targets is wrongly estimated for $6$ consecutive moments. Generally the lost tracking ratio decreases rapidly with the increase of the number of particles. However, it tends to decrease gradually with the increase of the number of particles when the number of particles becomes large enough. In our example, $2048$ of the total number of particles are available, i.e., $L_k = 1024$ and $J_k = 1024$.

On the other hand, several measurements may be available at each time step, and each measurement may be generated by survival targets or spontaneous birth targets or clutters. For the traditional particle PHD filter, obviously the participation of clutters in the Weight Computation will not only lead to a high processing delay, but also inevitably has some negative effects on the estimation results of the survival targets and the spontaneous birth targets. Also, the measurements of spontaneous birth targets in the update of survival targets may dramatically decrease the estimate quality of survival targets, and vice versa. Hence, we present an efficient data-driven mechanism to solve the above problems and speed up the particle PHD filtering.

## III. Data-Driven Particle PHD Filter

In this section, we describe our data-driven particle PHD filter in detail. Concretely, we first give the target tracking model, present our data-driven mechanism, and then apply the data-driven mechanism in the particle PHD filter together with analysis of time delay and computational complexity.

### A. Bearing and Range Tracking Model

For most tracking systems, the target state is modeled in Cartesian coordinates and maintained in a reference frame that is stabilized relatively to the location of the platform. The dynamical equation that is commonly used to represent the motion of the target relative to the platform is given by

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \Gamma \xi_k \tag{5}$$

where

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

is the dynamical constraint for nearly constant velocity motion at time $k$. $\Delta T$ denotes the sample period.

$$\mathbf{\Gamma} = \begin{bmatrix} \Delta T^2/2 & 0 \\ 0 & \Delta T^2/2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix}$$

is the input matrix. $\mathbf{x}_k = [x \quad v_x \quad y \quad v_y]_k^T$ is the target state vector at time $k$. $(x, y)_k$ and $(v_x, v_y)_k$ represent the position and the velocity of the target in Cartesian coordinates at time $k$, respectively. $\xi_k = (\xi_x, \xi_y)_k^T$ is the vector of input white noise with zero mean in Gaussian distribution with $\xi_k \sim N(0, Q_k)$.

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

$$r_k = \left\| \mathbf{H}\mathbf{x}_k - \begin{bmatrix} x_s \\ y_s \end{bmatrix} \right\| + w_{1,k} \tag{6}$$

$$\theta_k = \arctan\left( \frac{\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k + y_s}{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_k + x_s} \right) + w_{2,k} \tag{7}$$

where $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

The sensor is located at $[x_s, y_s]^T$, and the measurement noises $w_{1,k}$ and $w_{2,k}$ are zero-mean Gaussian white noise with mutually independent standard deviations.

### B. Data-Driven Mechanism with Gating Technique

For our tracking task, the measurement at time $k$ is not a single measurement $\mathbf{z}_k$ but a measurement set $\mathbf{Z}_k$. Each measurement may be generated by survival targets or spontaneous birth targets or clutters. In the traditional particle PHD filter, we usually make use of all input measurements into the Weight Computation to update the weights of each particle. However, it will incur heavy computational time delay because of the participation of all input measurements. In addition, when the spontaneous birth measurements are taken into account in the update of survival targets, it may dramatically decrease the estimate quality of survival targets, and vice versa. Therefore, we present a data-driven mechanism to improve the processing time by classifying the input measurements.

To solve the challenge of distinguishing measurements of survival targets from spontaneous birth targets, the validation gating technology is designed. The validation region is used to reduce the number of candidate measurements to a value that can be reasonably associated with the predicted target state. The validation window based on the concept of statistical distance is given by

$$d^2_{statistical} = \tilde{\mathbf{Z}}_k^T S_k^{-1} \tilde{\mathbf{Z}}_k \tag{8}$$

where $\tilde{\mathbf{Z}}_k$ is the measurement residual vector (i.e., the difference between the measurement and the predicted location) at time $k$ and $S_k$ is the residual covariance matrix. The residual covariance matrix is the sum of the covariance matrix of the measurement error and the covariance matrix of the predicted measurement. A measurement is accepted as a valid measurement if

$$d^2_{statistical} \le \lambda \tag{9}$$

The parameter $\lambda$ denotes an appropriate threshold. Each measurement satisfying the expression is assumed to be the reasonable candidate for association with the given track.

As a result, a reasonable measurement associated with the $i$-th survival target's predicted position is defined as follows:

$$\begin{aligned}\tilde{\mathbf{z}}_{k+1}^{i} &= (\mathbf{z}_{k+1,s}: \\ &\min_{s}((\mathbf{z}_{k+1,s} - \mathbf{HF}\mathbf{x}_{k}^{i})^{T} \times \sum\nolimits_{v}^{-1}(\mathbf{z}_{k+1,s} - \mathbf{HF}\mathbf{x}_{k}^{i})))\end{aligned} \tag{10}$$

where $\mathbf{HF}\mathbf{x}_{k}^{i}$ is the predicted location for the particle $\mathbf{x}_{k+1}^{i}$, and $\mathbf{z}_{k+1,s}$ is the $s$-th measurement in the set $\mathbf{Z}_{k+1}$. The measurement set of survival targets is defined as the union of all survival measurements, i.e.,

$$\tilde{\mathbf{Z}}_{k+1} = \bigcup_{i=1}^{\hat{N}_k} \tilde{\mathbf{z}}_{k+1}^{i} \tag{11}$$

Then, the residual measurement set $\overline{\mathbf{Z}}_{k+1}$ for spontaneous birth targets is defined by drawing the survival measurements out of the set $\mathbf{Z}_{k+1}$:

$$\overline{\mathbf{Z}}_{k+1} = \mathbf{Z}_{k+1} - \tilde{\mathbf{Z}}_{k+1} \tag{12}$$

When no spontaneous birth target appears at time $k+1$, the set $\overline{\mathbf{Z}}_{k+1}$ consists of all clutters. On the contrary, when there is a spontaneous birth target in the scene, the corresponding measurement is included in the set $\overline{\mathbf{Z}}_{k+1}$.

Under the assumption that only one target appears spontaneously at each time step, let $\mathbf{z}_{k+1,j}$ be the measurement which is the nearest one to the position of the mean of the spontaneous birth targets, i.e.,

$$\mathbf{z}_{k+1,j} = (\mathbf{z}_{k+1,s} : \min_{s}(|\mathbf{z}_{k+1,s} - \mathbf{H}\overline{\mathbf{x}}|)) \tag{13}$$

where $|\mathbf{z}_{k+1,s} - \mathbf{H}\overline{\mathbf{x}}|$ is the Euclid distance between $\mathbf{z}_{k+1,s}$ and $\mathbf{H}\overline{\mathbf{x}}$. $\mathbf{z}_{k+1,s}$ is the $s$-th measurement in the set $\overline{\mathbf{Z}}_{k+1}$. The mean of the initial state distribution of spontaneous birth target is $\overline{\mathbf{x}} = (x^{\text{new}}, v_x^{\text{new}}, y^{\text{new}}, v_y^{\text{new}})_k^T$. The superscript "new" denotes spontaneous birth targets.

Note that the above data-driven mechanism can be applied when the two assumptions hold: (1) the target maneuvering is not too abrupt; and (2) the sample period $\Delta T$ is not too large.

### C. Data-Driven Particle PHD Filter

With the above data-driven mechanism, our proposed data-driven particle PHD filter is described as follows.

For a set of arriving measurements in the particle PHD iteration at time $k$, the first step is to derive the survival measurements and the spontaneous birth measurements respectively from all the input measurements.

Step 1: Data-driven Mechanism

a). Obtain the survival measurements $\tilde{\mathbf{Z}}_k$ according to Eq. (10): the survival measurements are extracted by comparing each input measurement with the predicted position of each target. At time $k=1$, no survival target is considered, i.e., only one spontaneous birth measurement $\mathbf{z}_{1,j}$ is obtained at the first time step according to Eq. (13).

b). By substracting the survival measurements from all input measurements, the possible measurements for the spontaneous birth targets are obtained according to Eq. (12).

c). Obtain the spontaneous birth measurement $\mathbf{z}_{k,j}$ according to Eq. (13): the spontaneous birth measurements are extracted by comparing each possible spontaneous birth measurement with the initial position of the spontaneous birth targets, and the remaining measurements are regarded as clutters, which will not participate in the update of the particles. While no spontaneous birth target appears, only the survival measurements $\tilde{\mathbf{Z}}_k$ are obtained.

Step 2: Prediction

For $i = 1, ..., L_{k-1}$, sample $\tilde{\mathbf{x}}_k^{(i)} \sim q_k(\cdot|\tilde{\mathbf{x}}_{k-1}^{(i)}, \mathbf{Z}_k)$ and compute the predicted weights

$$\tilde{w}_{k|k-1}^{(i)} = \frac{\phi_{k|k-1}(\tilde{\mathbf{x}}_k^{(i)}, \mathbf{x}_{k-1}^{(i)})}{q_k(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)} w_{k-1}^{(i)} \tag{14}$$

For $i = L_{k-1}+1, ..., L_{k-1}+J_k$, sample $\tilde{\mathbf{x}}_k^{(i)} \sim p_k(\cdot|\mathbf{Z}_k)$ and compute the predicted weights

$$\tilde{w}_{k|k-1}^{(i)} = \frac{1}{J_k} \frac{\gamma(\tilde{\mathbf{x}}_k^{(i)})}{p_k(\tilde{\mathbf{x}}_k^{(i)}|\mathbf{Z}_k)} \tag{15}$$

Step 3: Update

For survival targets, for each $\mathbf{z} \in \tilde{\mathbf{Z}}_k$, compute

$$C_k(\mathbf{z}) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{k,\mathbf{z}}(\tilde{\mathbf{x}}_k^{(j)}) \tilde{w}_{k|k-1}^{(j)} \tag{16}$$

For $i \in$ survival particles, update weights

$$\tilde{w}_k^{(i)} = [1 - P_D + \sum_{\mathbf{z} \in \tilde{\mathbf{Z}}_k} \frac{\psi_{k,\mathbf{z}}(\tilde{\mathbf{x}}_k^{(i)})}{\kappa_k(\mathbf{z}) + C_k(\mathbf{z})}] \tilde{w}_{k|k-1}^{(i)} \tag{17}$$

At time step $k=1$, this sub-step can be ignored.

For spontaneous birth targets, for $\mathbf{z} = \mathbf{z}_{k+1,j}$, compute

$$C_k(\mathbf{z}) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{k,\mathbf{z}}(\tilde{\mathbf{x}}_k^{(j)}) \tilde{w}_{k|k-1}^{(j)} \tag{18}$$

For $i \in$ spontaneous birth particles, update weights

$$\tilde{w}_k^{(i)} = [1 - P_D + \sum_{\mathbf{z} = \mathbf{z}_{k+1,j}} \frac{\psi_{k,\mathbf{z}}(\tilde{\mathbf{x}}_k^{(i)})}{\kappa_k(\mathbf{z}) + C_k(\mathbf{z})}] \tilde{w}_{k|k-1}^{(i)} \tag{19}$$

When no spontaneous birth target appears, this sub-step can be ignored.

Step 4: Resample

Compute the total mass $\tilde{N}_k = \sum_{i=1}^{L_{k-1}+J_k} \tilde{w}_k^{(i)}$ and estimate the number of targets $\hat{N}_k = \text{round}(\tilde{N}_k)$, resample $\{\tilde{\mathbf{x}}_k^{(i)}, \tilde{w}_k^{(i)}/\tilde{N}_k\}_{i=1}^{L_{k-1}+J_k}$ to get $\{\mathbf{x}_k^{(i)}, w_k^{(i)}/\tilde{N}_k\}_{i=1}^{L_k}$, and rescale (multiply) the weights by $\tilde{N}_k$ to get $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{L_k}$.

### D. Analysis of Time Delay and Computational Complexity

In the traditional particle PHD filter, all the input measurements in the set $\mathbf{Z}_{k+1}$ and all particles have to be considered in the Weight Computation of all predicted particles. In the data-driven particle PHD filter, since targets are modeled into two categories: survival targets and spontaneous birth targets, thus all the measurements are classified into survival measurements and spontaneous birth measurements correspondingly. Hence, the PHD weight computation is carried out on survival measurements and spontaneous birth measurements, respectively. For survival targets, the processing time depends only on the number of survival particles and the number of the survival measurements. At time step $k = 1$, this sub-step is ignored. For spontaneous birth targets, the processing time depends only on the number of spontaneous birth particles and the number of spontaneous birth measurements. When no spontaneous birth target appears, this sub-step is ignored. The further results of the quantitative analysis on the improved computational time are presented in the Discussion and Extension Section.

According to the analysis, the computational complexity of data-driven mechanism step in the data-driven particle PHD filter at time $k+1$ is $O(\hat{N}_k|Y_{k+1}|)$, which is much lower than that of an earlier data-driven mechanism $O((L_k + J_{k+1})|Y_{k+1}|)$ in [25] that is carried out by designing the importance functions and correspondent weight functions for survival and spontaneous birth targets.

## IV. Performance Evaluation

### A. Simulation Setup

To validate the proposed data-driven particle PHD filter, scenarios are generated according to [11], and targets can appear or disappear in the scene at any time. The probability of target survival is $e_{k|k-1}(\cdot) = 0.95$ and no spawning is considered for simplicity.

Each spontaneous birth target has initial state distribution according to a Gaussian distribution with mean and covariance

$$\overline{\mathbf{x}} = [100m \quad 3m/s \quad 100m \quad -3m/s]^T \tag{20}$$

$$Q_x = \text{diag}([10m^2 \quad 1(m/s)^2 \quad 10m^2 \quad 1(m/s)^2]) \tag{21}$$

The number of target births follows a Poisson distribution with an average rate of $0.2$ target per scan.

For comparison, the results of tracking three maneuvering trajectory using the traditional particle PHD filter are presented to evaluate the performance. For simulation parameters, the sampling interval is $\Delta T = 1$; the process noises are $\sigma_{\xi_x} = 0.8, \sigma_{\xi_y} = 0.08$, respectively; the probability of detection is $P_D(\mathbf{x}_k) = 0.98$; the measurement errors are $\sigma_{w_x} = 2.5, \sigma_{w_y} = 0.005$, respectively. Clutters are uniformly distributed over a $300m \times 100m$ rectangle region. The number of clutter points per scan is Poisson distribution with an average rate of $r = 10$.

### B. Simulation Results

The true trajectories of $3$ tracks over $40$ scans are plotted in Fig. 2.

Fig. 2 also shows the positions of the estimated targets superimposed on the tracks over 40 time steps at the meantime. The individual $x$ and $y$ coordinates of the tracks and estimated targets for each time step are shown in Fig. 3. It can be seen that estimated positions based on the traditional particle PHD filter and the data-driven particle PHD filter are similar and they are all close to the true tracks at a single MC trial.

Fig. 4 plots the estimated measurements superimposed on both the input measurements and the true tracks over 40 time steps. It is clearly seen that the estimated measurements are closely related to the true tracks, and most of the clutters are eliminated which will benefit a lot to the tracking performance. From Fig. 4, it can be incurred that the proposed data-driven mechanism is very effective to distinguish the survival measurements and spontaneous birth measurements from clutters.

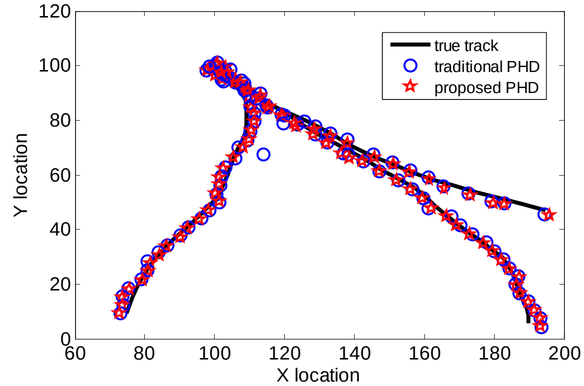


Fig. 2. Estimated trajectories and true trajectories

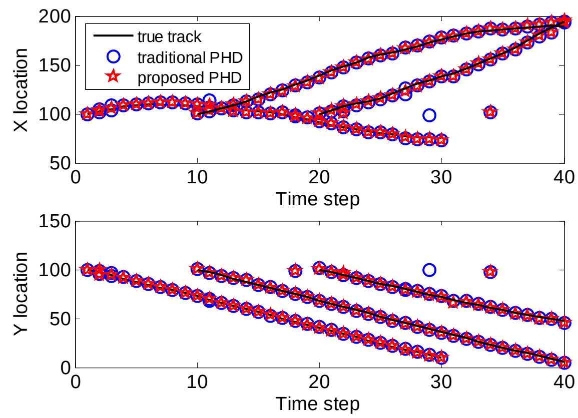


Fig. 3. Estimated positions and true positions in the $x$ and $y$ directions

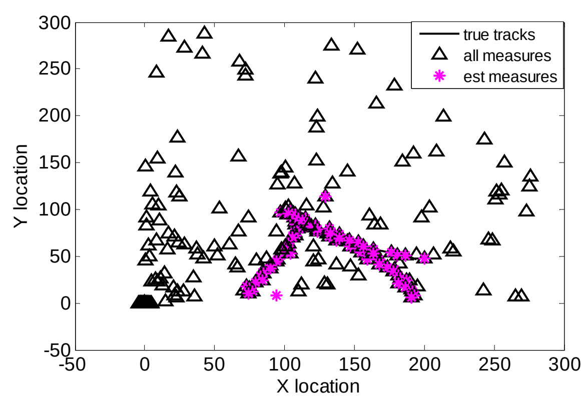


Fig. 4. Estimated measurements superimposed on the tracks and all input measurements

It was proposed in [26] to use the Optimal Sub-Pattern Assignment (OSPA) as a multi-target miss-distance metric, i.e., the error between the estimated and true state. Compared with the former Wasserstein distance metric [27], the OSPA distance jointly captures differences in cardinality and individual elements between two finite sets in a mathematically consistent yet intuitively meaningful way.

The OSPA metric $\overline{d}_p^{(c)}$ is defined as follows. Let $d^{(c)}(x,y) := \min(c, \|x-y\|)$ for $x, y \in \chi$, and $\Pi_k$ denotes the set of permutations on $\{1,2,...,k\}$ for any position integer $k$. Then, for $p \geq 1$, $c > 0$, and $\mathbf{X} = \{x_1,...,x_m\}$ and $\mathbf{Y} = \{y_1,...,y_n\}$ in $\mathcal{F}(\chi)$, if $m \leq n$,

$$\overline{d}_p^{(c)}(\mathbf{X},\mathbf{Y}) := \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m} d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m)\right)\right)^{\frac{1}{p}} \tag{22}$$

and if $m > n$, $\overline{d}_p^{(c)}(\mathbf{X},\mathbf{Y}) := \overline{d}_p^{(c)}(\mathbf{Y},\mathbf{X})$ ; and if $m = n = 0$, $\overline{d}_p^{(c)}(\mathbf{X},\mathbf{Y}) = \overline{d}_p^{(c)}(\mathbf{Y},\mathbf{X}) = 0$ . This distance is interpreted as a $p$-th order per-target error, comprising a $p$-th order per-target cardinality error. The order parameter $p$ determines the sensitivity to outliers, and the cut-off parameter $c$ determines the relative weighting of the penalties assigned to cardinality and localization errors. According to the analysis in [26], we choose the parameters $p = 2$ and $c = 20$ in our simulation.

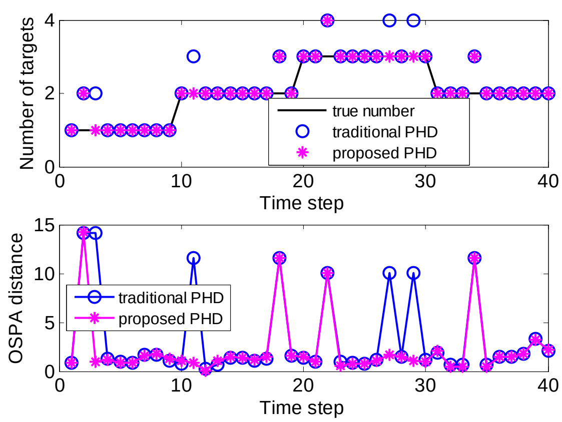


Fig. 5. Estimated number of targets and multi-target miss distance of the data-driven particle PHD filter implemented in MATLAB: (a) Estimated number of targets versus the true number of targets, (b) Multi-target miss distance

Fig. 5 plots the estimated targets against ground truth in terms of target number and OSPA multi-target miss-distance at each time step. It can be seen that the data-driven particle PHD filter has better tracking performance than the traditional particle PHD filter. And multi-target miss-distance exhibits peaks at the instances where the estimated number is incorrect. When the estimated number of targets is correct, the OSPA miss-distance is relatively small.

Fig. 6 shows the 5000 MC average of the OPSA distance for $p = 2$ and $c = 100$, which provides a natural and intuitive interpretation of the OSPA metric in terms of localization and cardinality error. Note that in terms of cardinality error, the OSPA distance demonstrates a small pulse once the number of targets varies. From the figure, it is clear that the data-driven particle PHD filter achieves a lower error than the traditional particle PHD filter.

Table I gives the 5000 MC average OSPA distance and the 5000 MC average computational time in MATLAB R2007b

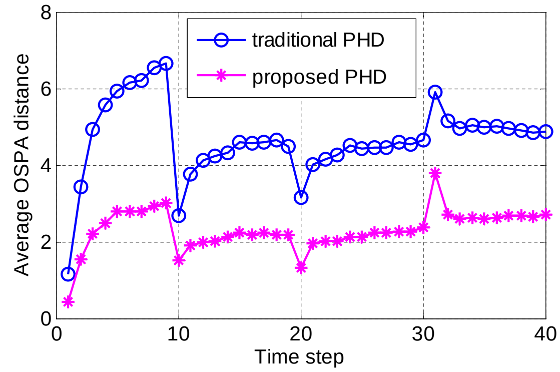


Fig. 6. 5000 MC average OSPA distance versus time

TABLE I
COMPARISON BETWEEN THE TRADITIONAL PARTICLE PHD FILTER AND PROPOSED PARTICLE PHD FILTER

| Filter algorithm | Average OSPA | Comput. time (s) |
|---|---|---|
| Traditional PHD | 4.3052 | 2.8231 |
| Proposed PHD | 2.1709 | 1.5145 |

on a Core 4 Duo 2.40-GHz workstation with 24GB RAM for both filters with the clutter density of $r = 10$.

From Table I, the average OSPA distance of the data-driven particle PHD filter is $2.1709$, and that of the traditional particle PHD filter $4.3052$. It is proved that the tracking performance of the data-driven particle PHD filter is much better than that of the traditional particle PHD filter in terms of localization and cardinality error. From the theory aspect, it is because that all input measurements are explicitly classified into two categories of survival measurements and spontaneous birth measurements after eliminating most of the clutters, so the survival measurements are only used for updating the survival particles and spontaneous birth measurements are only used for updating the spontaneous birth particles in data-driven particle PHD filtering.

On the other hand, for the average computational time, both filters run $5,000$ MC trials and their computational time are recorded. Then the average time is computed. From Table I, it can be found that compared with the traditional particle PHD filter, the data-driven particle PHD filter has faster processing rate, that is $(2.8231 - 1.5145)/2.8231 = 46.4\%$. The main reason is that the data-driven particle PHD filter discards most clutters in the Weight Computation. Moreover, the processing time of data-driven particle PHD filter will gain more improvements when the clutters become denser in the scene, which will be discussed later.

## V. DISCUSSION AND EXTENSION

### A. Real-Time Performance Improvement

To evaluate the real-timeness, we define the Real-Time Performance Improvement (RTPI) between the traditional PHD filter and the data-driven particle PHD filter as:

$$\text{RTPI} = \frac{\bar{t}_{\text{TRA}} - \bar{t}_{\text{DD}}}{\bar{t}_{\text{TRA}}} \times 100\% \tag{23}$$

where $\bar{t}_{\rm TRA}$ and $\bar{t}_{\rm DD}$ are the $5,000$ MC average computational time in the traditional particle PHD filter and the data-driven particle PHD filter, respectively. The RTPI of the data-driven particle PHD filter is plotted in Fig. 7. It can be seen that the proposed data-driven particle PHD filter gains better real-time performance with higher clutter density. The reason is that the Weight Computation is simplified as most of the clutters are eliminated from all the measurements. Even when the clutter density is at an average rate of $r = 0$, the data-driven particle PHD filter can gain $7.3\%$ real-time performance improvement, since all the measurements are treated separately. Because the participation of clutters occupies much processing time in the Weight Computation, the proposed data-driven mechanism can benefit the real-time industrial applications.

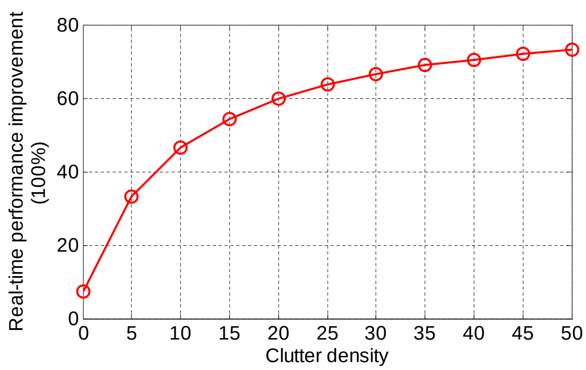


Fig. 7. Real-time performance improvement versus clutter density

### B. Tracking Performance Improvement

In Fig. 8, the 5000 MC average of the OPSA distance (time-averaged over the whole tracks) for the traditional particle PHD filter and the data-driven particle PHD filter are shown versus clutter densities from 0 to 50. As expected, the OSPA distance increases with the increment of clutter density. From the figure, it appears that the proposed data-driven particle PHD filter has much better performance than that of the traditional particle PHD filter. That is because that the data-driven mechanism tries to distinguish the survival measurements and spontaneous birth from all measurements in the scene.

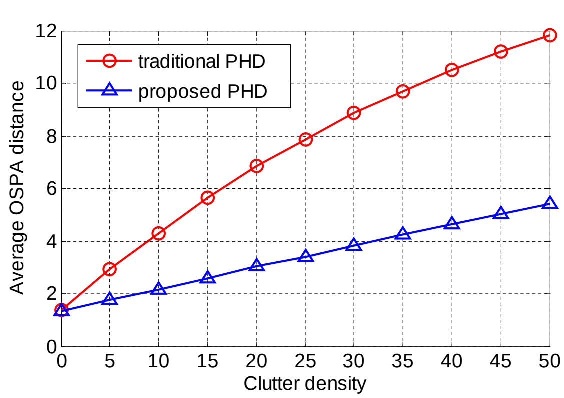


Fig. 8. Time-averaged OSPA distances for varying clutter density

## VI. RELATED WORKS

In recent years, there are some related works on improving the computation efficiency of PHD filter by using gating techniques. In Ref. [25], Wang et al proposed a data-driven mechanism for imaging MTT problem by designing the importance functions and corresponding weight functions where gating technique is used for classify the particles. In Ref. [10], Zhang et al. developed a method of reducing the computational cost of Gaussian mixture CPHD filter by incorporating the elliptical gating technique, where a gating validation region is proposed and evaluated. In Ref. [28], Swain et al. presented a first-moment recursion for a single-group filter and an closed-form solution under linear-Gaussian assumptions, where a gating technique is used to select the parental targets. In Ref. [29], Macagnano et al. studied multitarget tracking with the Gaussian mixture PHD and CPHD filter and proposed a novel weighted gating strategy that is adaptive to environmental settings.

The main idea of gating in these works are similar, i.e., classify either the particles [25] or the measurements [10], [28], [29] for different types of targets based on differences between measurements and state estimates. For our paper, the main differences are described as follows.

- To the best of our knowledge, it's the first time that we proposed how to use data-driven mechanism with gating technique to improve the real-time performance of particle PHD filter for multi-target tracking nonlinear/non-Gaussian system in dense clutter environment.
- The process of selecting measurements for survival and spontaneous birth targets is different: first, we select the measurements for the survival measurements; then, from the remaining measurements, we select the proper measurements for the spontaneous birth targets; meanwhile, the clutters are also eliminated. Another difference between the proposed method and existing ones is: because we select the measurements that are nearest to the expected positions, and consequently no specific gating level is required.
- In existing gating related techniques for multiple target tracking, the reduction of the computational complexity depends on the volume of the validating gate. In the proposed method, the reduction of the computational complexity is independent on the gating threshold and the clutter density. Thus, the real-time performance can be significantly improved, especially in case of dense clutter environment.
- In the proposed data-driven particle PHD filter, because of the weight update scheme proposed, the tracking performance are also improved, especially in dense clutter environment. This is different from other gating based techniques which will generally degrade the tracking performance to some extent.

## VII. CONCLUSION

In this paper, we have proposed an efficient data-driven particle PHD filter for real-time multi-target tracking applications. The novelty lies in the data-driven mechanism, by which the input measurements are classified in the Weight Computation. Since the participation of clutters is eliminated, the data-driven particle PHD filter gains better real-time performance with higher clutter density. The tracking performance can be improved as either survival measurements are used for updating the weights of survival particles, or the spontaneous birth measurements for updating the weights of spontaneous birth particles. The data-driven particle PHD filter has been demonstrated to be feasibly adapted to eliminate clutters generated by common targets detectors and dense clutter environment. In addition, it is possible to extend the proposed measurement-driven mechanism to CPHD filtering and MeMBer filtering, which will be discussed in the future work.

## ACKNOWLEDGMENT

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

## REFERENCES

[1] D. Bruckner, C. Picus, R. Velik, W. Herzner, and G. Zucker, "Hierarchical semantic processing architecture for smart sensors in surveillance networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 291–301, 2012.

[2] S. Shin, T. Kwon, G.-Y. Jo, Y. Park, and H. Rhy, "An experimental study of hierarchical intusion detection for wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 6, pp. 744–757, 2010.

[3] F. Zhang, J. Chen, H. Li, Y. Sun, and X. Shen, "Distributed active sensor scheduling scheme for target tracking in wireless sensor networks," *ACM Mobile Networks and Applications (MONET)*, to appear.

[4] L. Wu and D. W. C. Ho, "Fuzzy filter design for itô stochastic systems with application to sensor fault detection," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 233–242, 2009.

[5] S. Livatino, F. Banno, and G. Muscato, "3-d integration of robot vision and laser data with semiautomatic calibration in augmented reality stereoscopic visual interface," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 69–77, Feb 2012.

[6] M.-S. Lee and Y.-H. Kim, "An efficient multitarget tracking algorithm for car applications," *IEEE Transactions on Industrial Eletronics*, vol. 50, no. 2, pp. 397–399, 2003.

[7] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152 – 1178, 2003.

[8] ——, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 1523–1543, 2007.

[9] B. Ristic, 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, pp. 1656–1668, 2012.

[10] H. Zhang, Z. Jing, and S. Hu, "Gaussian mixture cphd filter with gating technique," *Signal Processing*, vol. 29, pp. 1521 – 1530, 2009.

[11] B. N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multi-target filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[12] B. N. Vo and W. K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[13] B. T. Vo, B. N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3553–3567, 2007.

[14] R. Mahler, "Statistical multisource-multitarget information fusion," Artech House, 2007.

[15] G. Buttazzo, E. Bini, and Y. Wu, "Partitioning real-time applications over multicore reservations," *IEEE Transactions on Industrial Electronics*, vol. 7, no. 2, pp. 302–315, May 2011.

[16] Z. Wang and D. Gu, "Cooperative target tracking control of multiple robots," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3232–3240, 2012.

[17] P. T. A. Quang and D.-S. Kim, "Enhancing real-time delivery of gradient routing for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 61–68, 2012.

[18] C. Tran and M. Trivedi, "3-d posture and gesture recognition for interactivity in smart spaces," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 178–187, 2012.

[19] Z. Liang, S. Feng, D. Zhao, and X. Shen, "Delay performance analysis for supporting real-time traffic in a cognitive radio sensor network," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 325–335, 2011.

[20] L. Wu and D. W. C. Ho, "Reduced-order l2-l1filtering of switched nonlinear stochastic systems," *IET Control Theory and Applications*, vol. 3, no. 5, pp. 493–508, 2009.

[21] B. T. Vo, B. N. Vo, and A. Cantoni, "The cardinality balanced multi-bernoulli filter and its implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.

[22] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[23] J. Lim and D. Hong, "Cost reference particle filtering approach to high-bandwidth tilt estimation," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3830–3839, 2010.

[24] S. Hong, Z. Shi, and K. Chen, "Easy-hardware-implementation mmpf for maneuvering target tracking: Algorithm and architecture," *Journal of Signal Processing Systems*, vol. 61, no. 3, pp. 259–269, 2010.

[25] Y. Wang, J. Wu, A. Kassim, and W. Huang, "Data-driven particle phd filter for visual tracking," *IEEE Transactions on Circuit and Systems For Video Technology*, vol. 18, no. 8, pp. 1085 – 1095, 2008.

[26] D. Schuhmacher, B. T. Vo, and B. N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3447–3457, Aug. 2008.

[27] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol. 34, no. 3, pp. 327–336, 2004.

[28] A. Swain and D. Clark, "The single-group phd filter: An analytic solution," in *Proceedings of the 14th International Conference on Information Fusion,*, July 2011, pp. 1–8.

[29] D. Macagnano and G. de Abreu, "Gating for multitarget tracking with the gaussian mixture phd and cphd filters," in *8th Workshop on Positioning Navigation and Communication*. IEEE, 2011, pp. 149–154.