World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:17, No:6, 2023

Gaussian Particle Flow Bernoulli Filter for Single Target Tracking

Hyeongbok Kim, Lingling Zhao, Xiaohong Su, Junjie Wang

Abstract—The Bernoulli filter is a precise Bayesian filter for single target tracking based on the random finite set theory. The standard Bernoulli filter often underestimates the number of the targets. This study proposes a Gaussian particle flow (GPF) Bernoulli filter employing particle flow to migrate particles from prior to posterior positions to improve the performance of the standard Bernoulli filter. By employing the particle flow filter, the computational speed of the Bernoulli filters is significantly improved. In addition, the GPF Bernoulli filter provides more accurate estimation compared with that of the standard Bernoulli filter. Simulation results confirm the improved tracking performance and computational speed in two- and three-dimensional scenarios compared with other algorithms.

Keywords—Bernoulli filter, particle filter, particle flow filter, random finite sets, target tracking.

I. Introduction

THE aim of target tracking is to estimate the number of targets and their states from time-varying measurements. In complex scenarios, there is a significant amount of clutter in the measurements obtained by the sensor. Estimation of the target state from the uncertainty measurements has become a popular research field. In recent years, several tracking methods have been developed for this situation. The multi-target tracking algorithms can be divided into data association methods [1]–[3] and random finite sets (RFS) [4]–[6]. The classical multi-target tracking algorithms based on data association include the joint probabilistic data association (JPDA) [7], [8] and multiple hypotheses tracking (MHT) [9]–[12]. Compared with the classical multi-target tracking algorithms, the RFS methods model the targets and measurements as RFS, thereby avoiding the data association problem. Representative methods based on RFS have been proposed in recent decades, such as the probability hypothesis density (PHD) filter [13], [14], cardinality PHD (CPHD) filter, Bernoulli filter, cardinality balanced multi-Bernoulli filter (CBMeMber) [15], [16], and labelled Bernoulli filter (LMB) [17], [18]. These filters have been successfully employed in numerous applications including radar/sonar tracking [19]–[21], computer vision [22], [23], sensor management [24], [25], and distributed target tracking [26]–[28]. The PHD filter propagates the first-order moment of an RFS to estimate the number of targets and their states. Mahler derived the CPHD filter, which propagates the cardinality distribution alongside its first moment. The CPHD filter can obtain more precision estimation results compared with the PHD filter, but at a higher computational cost. A significant disadvantage of the CPHD filter is the spooky effect [29]. Therefore, other RFS filters such as Bernoulli, CBMember, and LMB have been proposed.

The Bernoulli filter is the precise Bayes filter that propagates the parameters of a Bernoulli RFS for a single dynamic system, which can randomly switch on and off [15], [30], [31]. The study [32] used the Bernoulli filter to track a small UAV and provided the sensible estimates results over time. The authors developed an extension of the Bernoulli filtering on a moving platform in [33]. The Bernoulli filter can also be used in target detection and state estimation in low SNR cases [34] and moving target scenarios [35]. Gning et al. [36] presented a new Bernoulli filter based on the box particle filter, where each sample is represented by the box particle instead of the particle filter. The authors proposed an improved Bernoulli particle filter to track an underestimated number of targets [37]. However, the proposed improved Bernoulli particle filter still needs a large number of particles and the running time of this method is even longer than that of the standard Bernoulli filter.

Considering the implementation method, the Bernoulli filter has been implemented in two distinct fashions, i.e., as the Gaussian mixture Bernoulli (GM Bernoulli) filter and sequential Monte Carlo Bernoulli (SMC Bernoulli) filter. In the GM Bernoulli filter implementation, the Bernoulli distribution is assumed to be a GM, whereas in the SMC Bernoulli filter implementation, it is approximated by a set of weighted particles. The GM Bernoulli filter needs less computational resources; however, it is constrained in linear and mildly nonlinear systems. In contrast, the SMC Bernoulli filter is more suitable for the non-linear non-Gaussian scenario and it often requires a large number of particles, particularly in the high-dimensional state spaces.

To avoid the severe degeneracy induced by sampling in a high dimensional space, particle flow algorithms were recently proposed [38]–[42]. The basic idea of a particle flow filter is migrating the particles from prior to posterior positions over a pseudo-time variable. The particle flow approach can lead to improved filter performance, particularly for high-dimensional state spaces or highly informative measurements. A previous study [43] employed particle flow to implement the PHD filter. This study introduced the Gaussian particle flow (GPF) to the Bernoulli filter to improve its performance.

The benefits of the proposed approach are twofold. First, more accurate estimation results are obtained because the proposed particle flow filter can guide the particles to a

Hyeongbok Kim, Lingling Zhao, and Xiaohong Su are with the Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China (e-mail: kimhyeongbok@naver.com, zhaoll@hit.edu.cn, sxh@hit.edu.cn).

Junjie Wang is with Department of Medical Informatics, School of Biomedical Engineering and Informatics, Nanjing Medical University, Nanjing, China e-mail: (junjie2021@njmu.edu.cn).

highly likelihood region. Second, high-speed processing is achieved as fewer samples are needed compared with the SMC Bernoulli filter.

The remainder of this paper is organised as follows: In Section II, the problem of target tracking is formulated briefly. The theory of the standard Bernoulli filter and implementation of the particle filter is reviewed in Section III. In Section IV, the GPF Bernoulli filter is proposed in detail. The simulations setting and results are provided in Section V. In Section VI, the conclusions and future work are presented.

II. Problem Formulation

We suppose X_k and Z_k denote the target state and measurements at time k, then the dynamic system of the target can be represented as:

$$theX_k = f_{k|k-1}(X_{k-1}, v_{k-1}) \tag{1}$$

$$Z_k = h_k(X_k, u_k) \tag{2}$$

where $f_{k|k-1}(\cdot)$ and v_{k-1} denote the state transition function and process noise vector, respectively; and h_k denotes the measurement function, whereas u_k denotes the measurement noise vector. The measurement function h_k is often a nonlinear function.

In general, the collected measurements can originate from the target or clutters. The clutter rate λ_k is often assumed to follow a Poisson distribution and is independent of the target state. Moreover, the measurements cannot be obtained within each time frame because of the detection probability $P_d(k) < 1$. Thus, the goal of target tracking is to estimate the target state based on the collected measurements.

III. Standard Bernoulli Filter

A. Bernoulli Filter Theory

The Bernoulli filter models the target states as a Bernoulli RFS. A Bernoulli RFS has probability $1-q$ to be empty or has only one element whose distribution is based on the probability density function (PDF) $p(x)$ with probability q. The PDF of a Bernoulli RFS is obtained as follows:

$$f(X) = \begin{cases} 1-q, & \text{if} \quad X = \emptyset \\ q * p(x), & \text{if} \quad X = \{x\} \end{cases} \tag{3}$$

The Bernoulli filter comprises two steps: prediction and update at each time step [5], [30].

We assume that at time $k-1$, the probability of existence is $q_{k-1|k-1}$ and the posterior PDF of the target is $s(x)$. According to [30], the prediction equations at time k are obtained as follows:

$$q_{k|k-1} = p_b(1 - q_{k-1|k-1}) + p_s q_{k-1|k-1} \tag{4}$$

$$s_{k|k-1} = \frac{p_b(1-q_{k-1|k-1})b_{k|k-1}(x)}{q_{k|k-1}} + \frac{p_s q_{k-1|k-1} \int f_{k|k-1}(x_k|x_{k-1}) s_{k-1|k-1}(x_{k-1}) dx_{k-1}}{q_{k|k-1}} \tag{5}$$

where p_b and p_s denote the probabilities of the target birth and the survival target, respectively; and $f_{k|k-1}(x_k|x_{k-1})$ is the dynamic equation of the target.

After the prediction equation, the predicted density is obtained from the pair $\{q_{k|k-1}, s_{k|k-1}\}$. Then, the update equation of the Bernoulli filter can be derived from:

$$q_{k|k}(x) = \frac{1-\Delta_k}{1-\Delta_k q_{k|k-1}} q_{k|k-1} \tag{6}$$

$$s_{k|k}(x_k) = \frac{s_{k|k-1}(x_k)}{1-\delta_k} \left[1 - P_d(x_k) + P_d(x_k) \sum_{z \in Z_k} \frac{g_k(z|x_k)}{\lambda_k c(z)} \right] \tag{7}$$

where $g_k(z|x_k)$ denotes the likelihood function of the measurement z; and $c(z)$ and λ denote the PDF and the average number of clutter, respectively.

The quantity Δ_k is defined as:

$$\Delta_k = P_d(x_k) \left(1 - \sum_{z \in Z_k} \frac{\int g_k(z|x_k s_{k|k-1}(x_k)) dx_k}{\lambda_k c(z)} \right) \tag{8}$$

B. Particle Implementation

The Bernoulli filter can be implemented by a particle filter. This study used a set of particles $\{x^i_{k-1}, w^i_{k-1}\}_{i=1}^{N_{k-1}}$ to approximate the spatial posterior density function at time $k-1$

$$s_{k-1|k-1}(x_{k-1}) = \sum_{i=1}^{N_{k-1}} w^i_{k-1} \delta(x^i_{k-1}) \tag{9}$$

where x^i_{k-1} denotes the state of the ith particle, and w^i_{k-1} is the corresponding weight; N_{k-1} denotes the number of particles at time $k-1$ and $\delta(\cdot)$ is the Dirac delta function. Then, the prior density $s_{k|k-1}$ can be approximated by particles as:

$$s_{k|k-1}(x_k) \approx \sum_{i=1}^{N_{k-1}+N_b} w^i_{k|k-1} \delta(x) \tag{10}$$

where N_b denotes the number of birth particles. After prediction, the total number of particles is $N_{k|k-1} = N_{k-1} + N_b$. The particle set $\left\{x^i_{k|k-1}, w^i_{k|k-1}\right\}_{i=1}^{N_{k|k-1}}$ is the union of birth and survival particle sets.

$$\left\{x^i_{k|k-1}, w^i_{k|k-1}\right\}_{i=1}^{N_{k|k-1}} = \left\{x^i_{k|k-1,s}, w^i_{k|k-1,s}\right\}_{i=1}^{N_{k-1}} \cup \left\{x^i_{k,b}, w^i_{k,b}\right\}_{i=1}^{N_b} \tag{11}$$

The updated probability of existence can be expressed as:

$$q_{k|k}(x) = \frac{1-\Delta_k}{1-\Delta_k q_{k|k-1}} q_{k|k-1} \tag{12}$$

where

$$\Delta_k = P_d - \sum_{z \in Z_k} \sum_{i=1}^{N_{k|k-1}} \frac{p_D g_k(z|x^i_{k|k-1})}{\lambda_k c} w^i_{k|k-1} \tag{13}$$

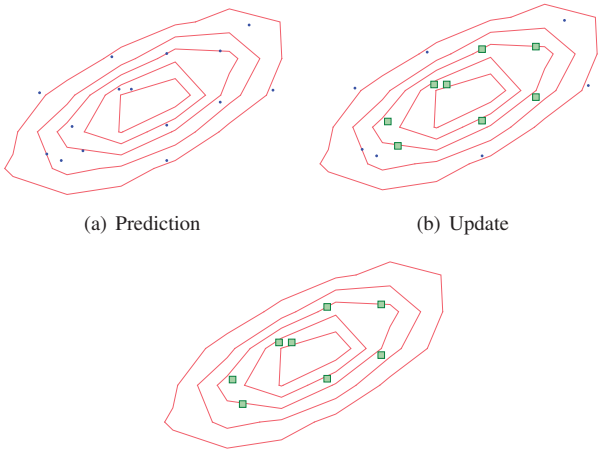
and the updated spatial PDF of the target is expressed as:

$$s_{k|k}(x) = \sum_{i=1}^{N_{k|k-1}} w^i_{k|k} \delta(x_{k^i}) \tag{14}$$

where the updated weights are computed as:

$$w^i_{k|k} = \frac{1 - p_D + \sum_{z \in Z_k} \frac{p_D g_k(z|x^i_{k|k-1})}{\lambda_k c}}{1 - \Delta_k} w^i_{k|k-1} \tag{15}$$

Finally, the resample step of the particle filter also needs to avoid the degeneration of the SMC Bernoulli filter. After the resampling step, the number of particles is changed to N_k.



(a) Prediction (b) Update

(c) Resample

Fig. 1 Abstract representation of the SMC Bernoulli filter showing its 2D state space for a single target

Fig. 1 shows the SMC Bernoulli filter at time k. In Fig. 1 (a), during the prediction of the SMC Bernoulli filter, the prior of the target density function is represented by the equal weight particles. In Fig. 1 (b), when the measurements arrive, the weights of the particles are updated using these measurements. The high-weight particles are represented by a coloured square shape. In the resampling step, the particles must be resampled based on their weights. The resampling method can abnegate low-weight particles and copy high-weight particles, as shown in Fig. 1 (c). Notably, the particle positions remained unchanged in one cycle. To obtain a better estimate result, the SMC Bernoulli filter requires more particles to guarantee that the maximum number of particles can fill the nearby target. However, this increases computational complexity.

IV. Gaussian Particle Flow Bernoulli Filter

This section describes the particle flow filter implementation of the Bernoulli filter under a Gaussian assumption. During the update step of the SMC Bernoulli filter, the particles do not change and only their weights evolve. It is difficult to guide particles to appropriate regions. The SMC Bernoulli filter does not have a state correction mechanism. The proposed method addressed this problem. The novelty of the GPF Bernoulli filter lies in the technique used for the particle flow filter. The particle flow filter migrates particles to regions of interest using a homotopy idea. The proposed method attempts to correct the particle state using a particle flow filter in the region of higher likelihood at each time step. The idea is similar to that of the GM Bernoulli filter, but each Gaussian is represented by a group of particles, where the group particles are updated by the particle flow filter.

The details of the particle flow filter implementation of the Bernoulli filter are as follows:

A. prediction

Suppose that at time $k-1$, the posterior of the target density is a pair $\pi_{k-1} = \{q_{k-1|k-1}, s_{k-1|k-1}\}$. The PDF $s_{k-1|k-1}(x_{k-1})$ is approximated as a Gaussian sum form as follows:

$$s_{k-1|k-1}(x_{k-1}) = \sum_{j=1}^{J_{k-1}} w^j_{k-1} \mathcal{N}(x; m^j_{k-1}, P^j_{k-1}) \tag{16}$$

Each Gaussian $\mathcal{N}(x; m^j_{k-1}, P^j_{k-1})$ contains a corresponding particle set $\left\{x^{I,j}_{k-1|k-1}\right\}_{i=1}^{M}$. Then, the surviving particle is predicted as:

$$x^{i,j}_{k|k-1} = f_{k|k-1}(x_k|x^i_{k-1}, Z_{k-1}) \tag{17}$$

The corresponding mean $m^j_{k|k-1}$ and covariance $P^j_{k|k-1}$ were computed as:

$$\begin{aligned} m^j_{k|k-1} &= \frac{1}{M}\sum_{i=1}^{M} x^{i,j}_{k|k-1} \\ P^j_{k|k-1} &= \frac{1}{M}\sum_{i=1}^{M}(m^j_{k|k-1} - x^{i,j}_{k|k-1})(m^j_{k|k-1} - x^{i,j}_{k|k-1})^T \end{aligned} \tag{18}$$

For the Bernoulli RFS $\{p_b, b_k(x)\}$, draw M particles $\left\{x^{i,j}_{b,k}\right\}_{i=1}^{M} \sim \mathcal{N}(\cdot; m^j_{b|k}, P^j_{b|k})$ for each birth component. Then, the predicted density is a union of the survival Gaussian and birth components. The number of Gaussian components is obtained as: $J_{k|k-1} = J_{k-1} + J_{k,b}$

$$\begin{aligned} &\left\{w^j_{k|k-1}, m^j_{k|k-1}, P^j_{k|k-1}, \left\{x^{i,j}_{k|k-1}\right\}_{i=1}^{M}\right\}_{j=1}^{J_{k-1}} \\ &\bigcup \left\{w^j_{b,k} m^j_{b,k}, P^j_{b,k}, \left\{x^{i,j}_{b,k}\right\}_{i=1}^{M}\right\}_{j=1}^{J_{k,b}} \end{aligned} \tag{19}$$

B. Update

We suppose that at time k, the predicted density is in a multi-Bernoulli form as:

$$\pi_{k|k-1} = \left\{q^i_{k|k-1}, s^i_{k|k-1}(x)\right\}_{i=1}^{L_{k|k-1}} \tag{20}$$

Then, the probability of existence is updated as:

$$q_{k|k} = \frac{1 - \Delta_k}{1 - \Delta_k q_{k|k-1}} q_{k|k-1} \tag{21}$$

The PDF is updated as:

$$\begin{aligned} s_{k|k}(x) &= \frac{1 - P_{D,k}}{1 - \Delta_k} s_{k|k-1}(x) + \\ &\frac{P_{D,k}}{1 - \Delta_k} \sum_{z \in Z_k} \sum_{j=1}^{J_{k|k-1}} \frac{w^j_{k|k-1} g^j_k(z)}{\lambda c(z)} \mathcal{N}(x; m^j_{k|k}, P^j_{k|k}) \end{aligned} \tag{22}$$

Algorithm 1 Pseudo-code for particle flow motion algorithm

Given $\{x_j\}_{j=1,...,M}, P, R_k, z$
set $\Delta = 0.1, \lambda, N_\lambda = 10$
for $i = 1$ to N_λ **do**
  $\lambda = i \cdot \Delta\lambda$
  Calculate the $m_i = \frac{1}{M}\sum_{j=1}^{M} x_j$
  Calculate the H_{m_i} by linearising measurement function at m_i
  $A = -\frac{1}{2} P H_{m_i}^T (\lambda H_{m_i} P H_{m_i}^T + R_k)^{-1} H_{m_i}$
  $b = (I + 2\lambda A)[(I + \lambda A) P H_{m_i}^T R_k^{-1} z + A m_i]$
  for $j = 1$ to M **do**
    Migrate particles:$x_j = x_j + \Delta\lambda(Ax_j + b)$
  end for
end for
output: $\{x_j\}_{j=1,...,M}$

where

$$\Delta_k = P_{D,k}[1 - \sum_{z \in Z_k} \sum_{j=1}^{J_{k|k-1}} \frac{w_{k|k-1}^j g_k^j(z)}{\lambda c(z)}] \tag{23}$$

$$g_k^j(z) = \mathcal{N}(z; H_k m_{k|k-1}^j, H_k P_{k|k-1}^j H_k^T + R) \tag{24}$$

where $c(z)$ denotes the clutter distribution, λ denotes the average clutter rate, and H_k denotes the Jacobian of the measurement function.

C. Correction

For each Gaussian component $\mathcal{N}(x; m_{k|k-1}^i, P_{k|k-1}^i)$, the corresponding particle set is $\left\{x_{k|k-1}^{i,j}\right\}, i = 1, ..., M,\ j = 1, ..., J_{k|k-1}$. Therefore, particles can be directly migrated by their flow. For $j = 1, ..., J_{k|k-1}$ and for each $z \in Z_k$, the particles migrate based on Algorithm 1. In this algorithm, P denotes the covariance matrix of the corresponding particles $\{x_j\}_{j=1,...,M}$ and R_k denotes the measurement noise covariance matrix. The particle flow parameters $\Delta\lambda$ and N_λ were used to migrate the particles in the discrete pseudo-time step.

For each Gaussian component, $\mathcal{N}(x; m_{k|k-1}^j, P_{k|k-1}^j)$, $m_{k|k}^j$, and $P_{k|k}^j$ are updated accordingly as:

$$\begin{aligned} m_{k|k}^j &= \frac{1}{M}\sum_{i=1}^{M} x_{k|k}^{i,j} \\ P_{k|k}^j &= \frac{1}{M}\sum_{i=1}^{M} (m_{k|k-1}^j - x_{k|k}^{i,j})(m_{k|k}^j - x_{k|k}^{i,j})^T \end{aligned} \tag{25}$$

Similar to the GM Bernoulli filter, the GPF Bernoulli filter also suffers from computational consumption problems resulting from increasing J_k as time evolves and the particle flow computational cost. Therefore, a similar pruning procedure can be exploited by discarding $s_{k|k}$ with light-associated probabilities or merging those sufficiently close to each other into one $s_{k|k}$. For particle flow computations, the computations can be reduced by only performing particle flow migration on $s_{k|k}$ with higher associated weights. Because the influence of $s_{k|k}$ with a low associated weight is negligible, its flow computation is unnecessary. The pseudocode for the GPF Bernoulli filter at time k is presented in Algorithm 2.

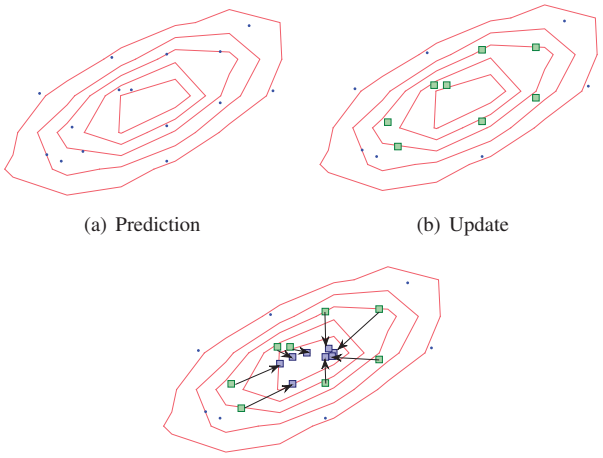
Algorithm 2 Pseudo-code for progressive GPF Bernoulli filter

Given posterior density at time $k-1$ $\{m_{k-1}^i, P_{k-1}^i, w_{k-1}^i\}_{i=1}^{J_{k-1}}$, birth intensity at time k $\gamma(x; m_{\gamma,k}^l, P_{\gamma,k}^l)_{l=1}^{J_{\gamma_k}}$, the existence probability $q_{k-1|k-1}$
for $i = 0$ to J_{k-1} **do**
  draw M particles $x_{k-1}^{i,j}, j = 1, ..., M$
  for $j = 1$ to M **do**
    $x_{k|k-1}^{i,j} = f(x_{k-1}^{i,j})$
  end for
  $w_{S,k|k-1}^i = p_{S,k} w_{k-1}^i, m_{S,k|k-1}^i = \frac{\sum_{j=1}^{M} x_{k|k-1}^{i,j}}{M}, P_{k|k}^i = \frac{\sum_{j=1}^{M}(x_{k|k-1}^{i,j} - m_{S,k|k-1}^i)(x_{k|k-1}^{i,j} - m_{S,k|k-1}^i)^T}{M}$
end for
$m_{k|k-1} = m_{S,k|k-1}, P_{k|k-1} = P_{S,k|k-1}, w_{k|k-1} = w_{S,k|k-1}$
for $l = 1$ to $J_{\gamma,k}$ **do**
  $m_{k|k-1}^{J_{k-1}+l} = m_{\gamma,k}^l, P_{k|k-1}^{J_{k-1}+l} = P_{\gamma,k}^l, w_{k|k-1}^{J_{k-1}+l} = w_{\gamma,k}^l$
end for
$J_{k|k-1} = J_{k-1} + J_{\gamma,k}$
Compute $q_{k|k-1} = p_b(1 - q_{k-1|k-1} + p_s q_{k-1|k-1})$

for $l = 1$ to $J_{k|k-1}$ **do**
  $w_{k|k}^l = (1 - P_{D,k}) w_{k|k-1}^l$,$P_{k|k}^l = P_{k|k-1}^l, m_{k|k}^l = m_{k|k-1}^l$
end for

$i = 0$
for each observation $z \in Z_k$ **do**
  $i = i + 1$
  for $l = 1$ to $J_{k|k-1}$ **do**
    $H_k^l = \frac{\partial h_k(x_k, 0)}{\partial x_k}|_{x_k = m_{k|k-1}^l}$
    $\eta_{k|k-1}^l = h_k(m_{k|k-1}^l, 0)$
    $S_k^l = R_k + H_k^l P_{k|k-1}^l H_k^{l\,T}$
    $w_k^{iJ_{k|k-1}+l} = P_{D,k} w_{k|k-1}^l g(z; \eta_{k|k-1}^l, S_k^l)$
  end for
  for $l = 1$ to $J_{k|k-1}$ **do**
    $x_{k|k}$ = ParticleFlow$(\{\hat{x}_{k|k-1}^{i,j}\}_{i=1,...,M}, P_{k|k-1}^l, R_k, z)$ according to Algorithm 1.
  end for
end for
$J_k = iJ_{k|k-1} + J_{k|k-1}$
Update $q_{k|k}$ according the equation (21)
Compute $m_{k|k}$ and $P_{k|k}$ according the equation (25)
output: $\left\{w_k^i, m_k^i P_k^i\right\}_{i=1,...,J_k}, q_{k|k}$

Fig. 2 illustrates three phases of the GPF Bernoulli Filter at time k. In the prediction and update process, it is almost the same as the standard Bernoulli Filter. However, to avoid selecting particles for the correction phase, this study employed the idea of the Gaussian mixture. Each particle set can be expressed as a Gaussian component. In



(a) Prediction (b) Update

(c) Correction

Fig. 2 Abstract representation of the GPF Bernoulli filter showing its 2D state space for a single target

the implementation, only the particle set with a high weight was migrated or corrected using a particle flow filter.

Remark: It is important to note that there exist several different realisations for the particle flow filter, such as exact Daum and Huang filter [40], non-zero diffusion particle flow filter [44], and incompressible flow filter [45]. In some cases, Gromov's method, as explained in [46]–[48], improves the particle flow filter performance.

V. Simulation Results and Discussions

In the numerical study, two-dimensional (2D) and three-dimensional (3D) target-tracking scenarios were used to evaluate the tracking performance of the proposed Bernoulli filter with a standard Bernoulli filter. The experimental environment was: IntelTM CoreTM i5, 8 GB Memory and MATLAB. The evaluation metric in the experiment was the optimal subpattern assignment (OSPA) distance [49].

A. Scenario 1

To verify the performance of the proposed filter in tracking problems, this study utilised a simulation scenario using the bearing and range tracking models. Considering the $[-1000, 1000] \times [-1000, 1000]$ region, targets move according to the nonlinear Gaussian dynamics, as expressed in (26), where the target state $x_k = [x_{1,k}, x_{2,k}, x_{3,k}, x_{4,k}]^T$ comprises the position $[x_{1,k}, x_{3,k}]^T$, velocity $[x_{2,k}, x_{4,k}]^T$ at time step k, and sampling period $T = 1$s. A nonlinear scenario with up to one target involving target birth, death, and clutter measurements is considered. The state transition model is as follows:

$$x_{k+1} = F(\omega_k)x_k + G\omega_{k+1} = \omega_k + \Delta u_k \tag{26}$$

where

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

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

$w_k \sim \mathcal{N}(\cdot; 0, \sigma_w^2 I), u_k \sim \mathcal{N}(\cdot; 0, \sigma_u^2 I), \Delta = 1$s,$\sigma_w = 15m/s^2$,and $\sigma_u = \pi/180rad/s$.

$$z_k = \begin{bmatrix} \arctan((p_x - x)/(p_y - y)) \\ \sqrt{(p_x - x)^2 + (p_y - y)^2} \end{bmatrix} + \varepsilon_k \tag{27}$$

where $\varepsilon_k \sim \mathcal{N}(\cdot; 0, R_k)$, $R_k = diag([\sigma_\theta^2, \sigma_r^2]^T), \sigma_\theta = \pi/180, \sigma_r = 1$. $x = 0, y = 0$ denote the sensor positions. In the simulation, the clutter was modelled as a Poisson RFS with a mean rate of 10 per scan. The GPF Bernoulli filter was compared with the GM Bernoulli filter, SMC Bernoulli filter1 with 3000 particles, and SMC Bernoulli filter2 with 5000 particles. In the GM Bernoulli and GPF Bernoulli filters, the maximum number of Gaussian components was set as 100, and the Gaussian component was pruned with threshold $1e-5$ and merged with 4. Each Gaussian component corresponded to 20 particles. The OSPA distance parameters are set as $p = 1$ and $c = 100$. Two different scenarios with detection probabilities $P_D = 0.7$ and $P_D = 0.9$ were considered.

Fig. 3 shows the OSPA distance for three filters. Under low probability detection, increased particle numbers cannot improve the SMC Bernoulli filter estimated result owing to the lack of measurement information. The improved performance of the GPF Bernoulli filter was attributed to the particle flow factor. At each time step, the particles are driven by measurements from prior to the posterior positions.

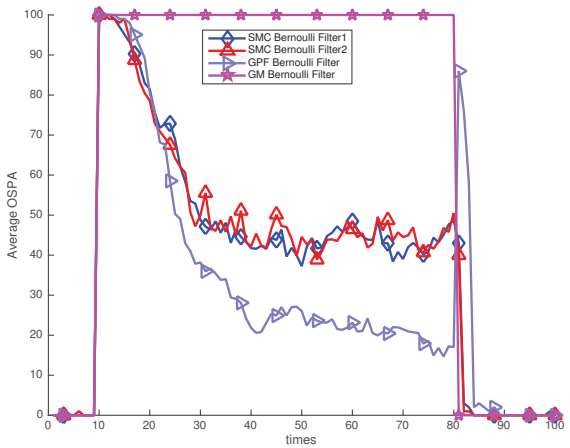


Fig. 3 Average OSPA over 100 Monte Carlo runs under $P_d = 0.7$

Fig. 4 shows the averaged OSPA distance for $P_D = 0.9$. It is clear that by increasing the number of particles, the SMC Bernoulli filter performance can be improved to obtain more measurement information. However, under different probability detections, all GPF Bernoulli filters perform better than the other filters. The benefit of using a particle flow filter is shown in Figs. 3 and 4.

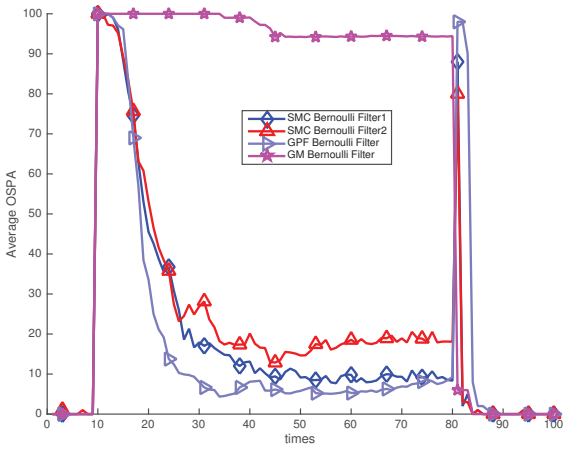


Fig. 4 Average OSPA over 100 Monte Carlo runs under $P_d = 0.9$

B. Scenario 2: 3D Target Tracking

The 3-degrees of freedom kinematic model is expressed as:

$$X(k) = FX(k-1) + Gw(k-1) \tag{28}$$

The state vector X comprises the position, velocity, and acceleration. The transition matrix F and noise gain matrix are obtained as:

$$F = diag[\Phi \quad \Phi \quad \Phi] \qquad G = diag[\eta \quad \eta \quad \eta] \tag{29}$$

where $\Phi = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$ $\eta = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}$ $T = 0.25$ is a sampling interval, the process-noise variance $q = 10$.

The measurements were captured using an infrared search-and-track sensor (IRST) and radar. The IRST can measure the azimuth θ and elevation ϕ, whereas the radar can measure azimuth θ, elevation ϕ, and range r. The measurements comprise information from the IRST and radar; thus, the measurement vector is obtained as:

$$z_k = \begin{bmatrix} z_k^{ir} \\ z_k^{rd} \end{bmatrix} \tag{30}$$

where $z_k^{ir} = [\theta_k^{ir} \quad \phi_k^{ir}]$ and $z_k^{rd} = [\theta_k^{rd} \quad \phi_k^{rd} \quad r_k^{rd}]$ correspond to the IRST and radar measurements, respectively. The corresponding measurement noise covariance can be expressed as:

$$R = \begin{bmatrix} R_k^{ir} & 0 \\ 0 & R_k^{rd} \end{bmatrix} \tag{31}$$

where $R_k^{ir} = \begin{bmatrix} (\sigma_\theta^{ir})^2 & 0 \\ 0 & (\sigma_\phi^{ir})^2 \end{bmatrix}$ and $R_k^{rd} = \begin{bmatrix} (\sigma_\theta^{rd})^2 & 0 & 0 \\ 0 & (\sigma_\phi^{rd})^2 & 0 \\ 0 & 0 & (\sigma_r^{rd})^2 \end{bmatrix}$.

The GPF Bernoulli filter was also compared with the GM Bernoulli filter, SMC Bernoulli filter1 with 10000 particles, and SMC Bernoulli filter2 with 50000 particles. In the GM Bernoulli and GPF Bernoulli filters, the maximum number of Gaussian components was set as 1000 and 100, respectively, and the Gaussian component was pruned with threshold $1e-5$ and merged with 4. Each Gaussian component corresponds to 50 particles. The OSPA distance parameters are set as $p = 1$ and $c = 100$.

Two different scenarios with detection probabilities $P_D = 0.7$ and $P_D = 0.9$ were considered.

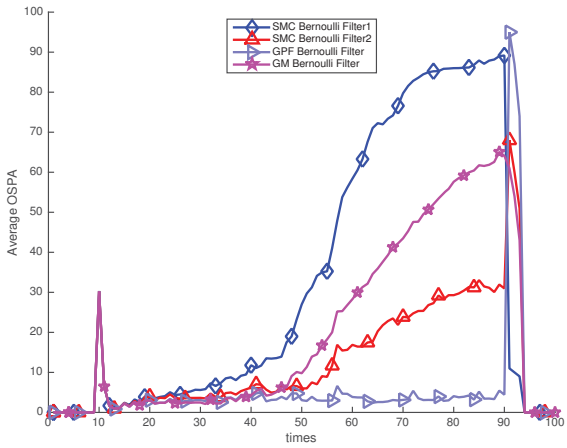


Fig. 5 Average OSPA over 100 Monte Carlo runs under $P_d = 0.7$ in the 3D scenario

Fig. 5 shows the OSPA distances for the three filters under detection probability $P_D = 0.7$. It is observed that the GPF Bernoulli filter has a lower OSPA distance compared to those of the GM and SMC Bernoulli filters, and even increases the Gaussian components and number of particles.

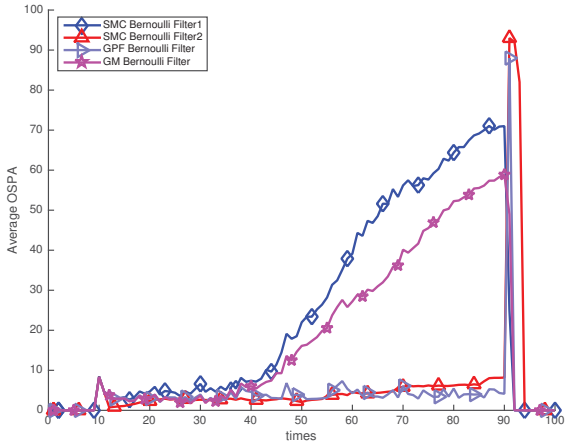


Fig. 6 Average OSPA over 100 Monte Carlo runs under $P_d = 0.9$ in the 3D scenario

Fig. 6 shows the average OSPA over 100 Monte Carlo runs under $P_d = 0.9$ in the 3D scenario. With an increased detection probability, the SMC Bernoulli filter2 performs similarly to the GPF implementation.

Fig. 7 shows the average OSPA error for different P_d. The SMC Bernoulli filter1 with a low number of particles suffers from a high OSPA error. With an increase in the number

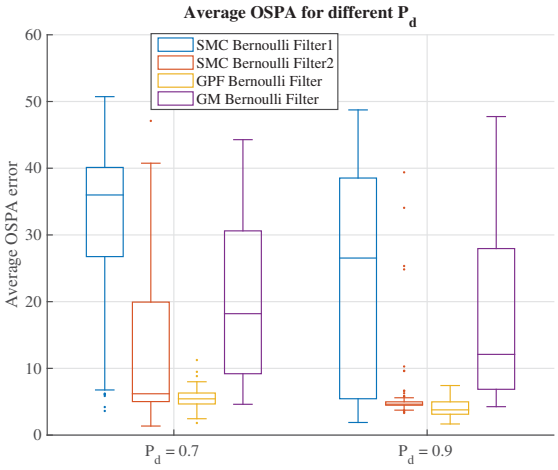


Fig. 7 OSPA error box plot for different P_d

of particles to 5000, the OSPA error of the SMC Bernoulli filter2 was slightly higher than that of the GPF Bernoulli filter. Table I lists the results of the filter performance in terms of $P_d = 0.7$ and $P_d = 0.9$. The average OSPA distance and computation time (CT) obtained for all filters averaged over 100 Monte Carlo runs. It can be observed from this table that the performance of the SMC Bernoulli filter appears to improve as the number of particles increases, accompanied by a rapid increase in the computing time. The proposed GPF Bernoulli filter exhibits superior performance in terms of the average OSPA distance under different detection probabilities.

TABLE I
FILTER PERFORMANCE IN TERMS OF DIFFERENT P_d, AVERAGE OSPA DISTANCE AND CT

P_d	Filter	OSPA(m)	CT(s)
0.7	SMC Bernoulli Filter1	32.36	54.85
	SMC Bernoulli Filter2	12.78	707.66
	GPF Bernoulli Filter	5.55	1.28
	GM Bernoulli Filter	20.33	0.63
0.9	SMC Bernoulli Filter1	24.03	57.25
	SMC Bernoulli Filter2	5.87	715.97
	GM Bernoulli Filter	4.00	1.57
	GPF Bernoulli Filter	18.25	0.58

Notably, the running time of the SMC Bernoulli filter2 is longer than that of the GPF Bernoulli filter. This can be explained by the fact that the particle flow filter requires fewer particles, even in the high-dimensional state space.

VI. CONCLUSION

This study proposed a GPF Bernoulli filter to improve the standard Bernoulli filter performance. The GPF Bernoulli filter is based on a particle flow filter, where the particles are guided by measurements. By applying the particle flow filter to the Bernoulli filter, the computational speed was significantly improved. Meanwhile, the particles migrated to the highly likelihood regions, producing a more robust and accurate tracking performance. The simulation results demonstrate that the GPF Bernoulli filter achieves a much faster processing speed compared with the standard SMC Bernoulli filter, as well as better estimation results for different detection probabilities. Future work will focus on improving the label multi-Bernoulli filter with a particle flow filter.

REFERENCES

[1] Y. Bar-Shalom and X.-R. Li, "Multitarget-multisensor tracking: principles and techniques," *Storrs, CT: University of Connecticut, 1995.*, 1995.
[2] Q. Li, J. Liang, and S. Godsill, "Scalable data association and multi-target tracking under a poisson mixture measurement process," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5503–5507.
[3] S. Casao, A. C. Murillo, and E. Montijano, "Data association tools for target identification in distributed multi-target tracking systems," in *Iberian Robotics conference*. Springer, 2023, pp. 15–26.
[4] R. P. Mahler, "A theoretical foundation for the stein-winter" probability hypothesis density (phd)" multitarget tracking approach," DTIC Document, Tech. Rep., 2000.
[5] ——, *Statistical multisource-multitarget information fusion*. Artech House Norwood, MA, USA, 2007, vol. 685.
[6] X. Cheng, H. Ji, and Y. Zhang, "Multiple extended target tracking based on gamma box particle and labeled random finite sets," *Digital Signal Processing*, p. 103902, 2022.
[7] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.
[8] T. Purushottama and P. Srihari, "Comparative analysis on diverse heuristic-based joint probabilistic data association for multi-target tracking in a cluttered environment," in *Proceedings of First International Conference on Computational Electronics for Wireless Communications: ICCWC 2021*. Springer, 2022, pp. 259–270.
[9] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
[10] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.
[11] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4696–4704.
[12] X. Weng, B. Ivanovic, and M. Pavone, "Mtp: multi-hypothesis tracking and prediction for reduced error propagation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1218–1225.
[13] R. P. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
[14] X. Shen, Z. Song, H. Fan, and Q. Fu, "A general cardinalized probability hypothesis density filter," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, p. 94, 2022.
[15] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-bernoulli filter and its implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.
[16] B. Yang, S. Zhu, X. He, K. Yu, and J. Zhu, "Robust measurement-driven cardinality balance multi-target multi-bernoulli filter," *Sensors*, vol. 21, no. 17, p. 5717, 2021.
[17] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, 2014.
[18] S. Robertson, C. van Daalen, and J. du Preez, "Efficient approximations of the multi-sensor labelled multi-bernoulli filter," *Signal Processing*, vol. 199, p. 108633, 2022.
[19] D. E. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the phd filter," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 327–334, 2005.
[20] M. Tobias and A. D. Lanterman, "Probability hypothesis density-based multitarget tracking with bistatic range and doppler observations," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 3, pp. 195–205, 2005.
[21] G. Li, P. Wei, G. Battistelli, L. Chisci, L. Gao, and A. Farina, "Distributed joint target detection, tracking and classification via bernoulli filter," *IET Radar, Sonar & Navigation*, vol. 16, no. 6, pp. 1000–1013, 2022.

[22] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, "Visual tracking of numerous targets via multi-bernoulli filtering of image data," *Pattern Recognition*, vol. 45, no. 10, pp. 3625–3635, 2012.

[23] A. Hoak, H. Medeiros, and R. J. Povinelli, "Image-based multi-target tracking through multi-bernoulli filtering with interactive likelihoods," *Sensors*, vol. 17, no. 3, p. 501, 2017.

[24] H. G. Hoang and B. T. Vo, "Sensor management for multi-target tracking via multi-bernoulli filtering," *Automatica*, vol. 50, no. 4, pp. 1135–1142, 2014.

[25] A. K. Gostar, R. Hoseinnezhad, W. Liu, and A. Bab-Hadiashar, "Sensor-management for multi-target filters via minimization of posterior dispersion," *IEEE Transactions on Aerospace and Electronic Systems*, 2017.

[26] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus cphd filter for distributed multitarget tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, 2013.

[27] B. Wang, W. Yi, S. Li, L. Kong, and X. Yang, "Distributed fusion with multi-bernoulli filter based on generalized covariance intersection," in *Radar Conference (RadarCon), 2015 IEEE*. IEEE, 2015, pp. 0958–0962.

[28] C. Fantacci, B.-N. Vo, B.-T. Vo, G. Battistelli, and L. Chisci, "Consensus labeled random finite set filtering for distributed multi-object tracking," *arXiv preprint arXiv:1501.01579*, 2015.

[29] D. Franken, M. Schmidt, and M. Ulmke, "" spooky action at a distance" in the cardinalized probability hypothesis density filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 4, 2009.

[30] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, "A tutorial on bernoulli filters: theory, implementation and applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3406–3430, 2013.

[31] B.-N. Vo, B.-T. Vo, N. T. Pham, and D. Suter, "Bayesian multi-object estimation from image observations," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 890–898.

[32] D. Cormack and D. Clark, "Tracking small uavs using a bernoulli filter," in *Sensor Signal Processing for Defence (SSPD), 2016*. IEEE, 2016, pp. 1–5.

[33] S. J. Julier and A. Gning, "Bernoulli filtering on a moving platform," in *Information Fusion (Fusion), 2015 18th International Conference on*. IEEE, 2015, pp. 1511–1518.

[34] J. Wang, W. D. Hu *et al.*, "Weak target detection exploiting bernoulli filter for ubiquitous radar," in *Progress in Electromagnetic Research Symposium (PIERS)*. IEEE, 2016, pp. 1361–1368.

[35] F. Papi, V. Kyovtorov, R. Giuliani, F. Oliveri, and D. Tarchi, "Bernoulli filter for track-before-detect using mimo radar," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1145–1149, 2014.

[36] A. Gning, B. Ristic, and L. Mihaylova, "Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2138–2151, 2012.

[37] B. Li, "An improved bernoulli particle filter for single target tracking," *Multidimensional Systems and Signal Processing*, pp. 1–21, 2017.

[38] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE*, vol. 6699, 2007, pp. 669 918–669 918.

[39] ——, "Particle flow for nonlinear filters with log-homotopy," in *SPIE Defense and Security Symposium*. International Society for Optics and Photonics, 2008, pp. 696 918–696 918.

[40] ——, "Nonlinear filters with particle flow induced by log-homotopy," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2009, pp. 733 603–733 603.

[41] ——, "Generalized particle flow for nonlinear filters," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2010, pp. 76 980I–76 980I.

[42] ——, "Particle flow with non-zero diffusion for nonlinear filters," in *Proceedings of SPIE: Signal processing, sensor fusion and target tracking XXII*, 2013, p. 87450P.

[43] L. Zhao, J. Wang, Y. Li, and M. J. Coates, "Gaussian particle flow implementation of phd filter," *SPIE Defense+ Security D*, vol. 98420, 2016.

[44] F. Daum and J. Huang, "Particle flow with non-zero diffusion for nonlinear filters," in *Proceedings of SPIE: Signal processing, sensor fusion and target tracking XXII*, 2013, p. 87450P.

[45] ——, "Nonlinear filters with log-homotopy," in *Optical Engineering+ Applications*. International Society for Optics and Photonics, 2007, pp. 669 918–669 918.

[46] F. Daum, J. Huang, and A. Noushin, "Gromov's method for bayesian stochastic particle flow: A simple exact formula for q," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2016 IEEE International Conference on*. IEEE, 2016, pp. 540–545.

[47] F. Daum, A. Noushin, and J. Huang, "Numerical experiments for gromov's stochastic particle flow filters," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2017, pp. 102 000J–102 000J.

[48] F. Daum, J. Huang, and A. Noushin, "Generalized gromov method for stochastic particle flow filters," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2017, pp. 102 000I–102 000I.

[49] 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, no. 8, pp. 3447–3457, 2008.