Multi-Bernoulli Filter Implementation in Images

Jean-Baptiste Courbot
INRIA Paris, MOKAPLAN team
LMD, UMR 8539, PSL-ENS/CNRS/UMPC/Ecole Polytechnique
Paris, France
jean-baptiste.courbot@inria.fr

***Abstract*—This paper considers the problem of estimating the number and state of objects appearing in a sequence of noisy images. We study the practical considerations involved in the design of the multi-Bernoulli particle filter in this context. From the sequential Monte-Carlo approach, we highlight the choices that must be made in order to implement it when tracking shapes in images. Numerical results illustrate in particular the advantage of choosing non-blind proposal densities.**

***Index Terms*—Multi-Bernoulli filter, sequential Monte Carlo, particle filtering, tracking in images**

I. Introduction

We target the problem of estimating the number and the state of objects appearing in a sequence of noisy images. This kind of problem is for instance common in the area of object tracking, with applications, *e.g.*, in security (moving targets over the sky) or meteorological data analysis (cloud or plume tracking). Several filtering methods have been developed in the last decade to tackle this type of problem. The first major contributions to this field were the introduction of the Probability Hypothesis Density (PHD) filter targeting single-state spaces introduced in [1] and its cardinalized version [2]. Both are first-moment approximations (moment and cardinality) of the recursive Bayes filter and are in practice implemented within a Sequential Monte Carlo (SMC) framework [3]. More recently, the Multi-target-Multi-Bernoulli recursion introduced a way to approximate the multi-target posterior density itself [4]. Its main advantage resides in the fact that the estimation of states cardinality and values are performed jointly and not sequentially as in the PHD filters. Several improvements of the Multi-Bernoulli Particle Filter (MBPF) were introduced, including cardinality-balanced [5] or labeled [6] filters for instance.

In this paper, we study the implementation of a MBPF as used in [7], *i.e.* applied on image data under the assumption that the regions influenced by the objects do not overlap. This implementation relies on several choices required to perform the computation within the SMC framework, which are generally made depending on the expected data to process. Here, we investigate and evaluate these SMC specificities.

The outline of this paper is the following. Section II recalls the SMC formulation of the MBPF, then several practical points are studied in Section III. Then, numerical results allow the evaluation of these choices in Section IV.

II. Sequential Monte-Carlo implementation

The MBPF relies on the description of a set of random states evolving sequentially, the cardinality of which is also an evolving random variable. This modeling is handled within the finite-set statistics [4] stemming from the point-process theory [8], using Random Finite Sets (RFS). This section recalls the SMC implementation of the MBPF [7].

Let $\mathcal{X}$ be a multi-Bernoulli RFS (*i.e.* a random variable set whose cardinality is also a random variable), representing the system states at all instants. The purpose of the filter is to infer the realization $\mathcal{X} = x$ from the realization of an observed image sequence denoted $\mathcal{Y} = y \overset{\text{def.}}{=} \{y_1, \ldots, y_K\}$. This inference is made possible, in a Bayesian setting, thanks to the propagation of a posterior density from time $k = 1$ to time K.

At each step $k \in \{1, \ldots, K\}$, $\mathcal{X}$ contains M_k states and its multi-object posterior density is given by $\pi_k = \{r_k^{(i)}, p_k^{(i)}\}_{i=1}^{M_k}$:

- $r_k^{(i)}$ is the probability of the RFS $i-$th state to be non-empty at time k;
- $p_k^{(i)}$ is comprised of a set of $J_k^{(i)}$ "particles" x paired with weight w such that:

$$p_k^{(i)}(x) = \sum_{j=1}^{J_k^{(i)}} w_k^{(i,j)} \delta_{x_k^{(i,j)}}(x) \qquad (1)$$

In the following we assume $J_k^{(i)}$ fixed and equals to J.

Then, the SMC procedure propagates recursively the posterior π_k from $k = 1$ to $k = K$ based on Monte-Carlo simulation of particles. In the sequel of this section, we recall the MBPF SMC implementation from [7].

Given a posterior density π_{k-1}, it relies on two main steps. The *prediction* step consists in computing $\pi_{k|k-1}$ from the previous posterior distribution π_{k-1} using *proposal densities*. The *update* step consists in computing π_k using $\pi_{k|k-1}$ and the *object likelihood.*

A. Prediction step

The posterior $\pi_{k|k-1}$ describes two phenomena: the propagation ("P" subset) of an object from time $k-1$ to time k, and the creation of new objects at time k ("C" subset). Hence, the SMC prediction step consists in computing:

$$\pi_{k|k-1} = \left\{ r_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)} \right\}_{i=1}^{M_{k-1}} \bigcup \left\{ r_{C,k}^{(i)}, p_{C,k}^{(i)} \right\}_{i=1}^{M_{C,k}}, \qquad (2)$$

where $M_{C,k}$ is the number of created objects at time k.

ISBN 978-90-827970-1-5 © EURASIP 2018

The propagation steps are computed as:

$$\left| \begin{array}{ll} r_{P,k|k-1}^{(i)} & = r_{k-1}^{(i)} \sum\limits_{j=1}^{J} w_{k-1}^{(i,j)} p_{S,k}\left(x_{k-1}^{(i,j)}\right) \\ p_{k|k-1}^{(i)}(x) & = \sum\limits_{j=1}^{J} w_{P,k|k-1}^{(i,j)} \delta_{x_{P,k|k-1}^{(i,j)}}(x) \end{array} \right. , \tag{3}$$

where $p_{S,k}(\cdot)$ is the survival probability at time k, and the second line of (3) reads similarly to (1). The states $x_{P,k|k-1}^{(i,j)}$ are sampled from the proposal density $q_k^{(i)}$ (see below):

$$\forall j \in \{1, \ldots, L\} : x_{P,k|k-1}^{(i,j)} \sim q_k^{(i)}\left(\cdot \,|\, x_{k-1}^{(i,j)}, y_k\right) \tag{4}$$

and the weights are computed using the single-target transition density $f_{k|k-1}(\cdot|x_{k-1}^{(i,j)})$:

$$w_{P,k|k-1}^{(i,j)} = \frac{w_{k-1}^{(i,j)} f_{k|k-1}\left(x_{P,k|k-1}^{(i,j)}|x_{k-1}^{(i,j)}\right) p_{S,k}\left(x_{k-1}^{(i,j)}\right)}{q_k^{(i)}\left(x_{P,k|k-1}^{(i,j)}|x_{k-1}^{(i,j)}, y_k\right)} \tag{5}$$

and normalized so that $\sum_{j=1}^{J} w_{P,k|k-1}^{(i,j)} = 1$.

The creation steps are computed as:

$$\left| \begin{array}{ll} r_{C,k}^{(i)} & = \text{“birth” parameter} \\ p_{C,k}^{(i)}(x) & = \sum\limits_{j=1}^{J} w_{C,k}^{(i,j)} \delta_{x_{C,k}^{(i,j)}}(x) \end{array} \right. , \tag{6}$$

where the $x_{C,k}^{(i,j)}$ are sampled from the proposal density $b_k^{(i)}$:

$$\forall j \in \{1, \ldots, L\} : x_{C,k}^{(i,j)} \sim b_k^{(i)}\left(\cdot|y_k\right), \tag{7}$$

and the weights $w_{C,k}^{(i,j)}$ are computed as:

$$w_{C,k}^{(i,j)} = \frac{p_{C,k}\left(x_{C,k}^{(i,j)}\right)}{b_k^{(i)}\left(x_{C,k}^{(i,j)}|y_k\right)} \tag{8}$$

and normalized so that $\sum_{j=1}^{J} w_{C,k}^{(i,j)} = 1$.

B. Update step

The update step does not require distinction between the propagated and the created states as in (2) , so we denote for concision:

$$\pi_{k|k-1} = \left\{ r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)} \right\}_{i=1}^{M_{k-1}+M_{C,k}}. \tag{9}$$

Denoting $g_k(\cdot|y_k)$ the likelihood of a given state, the update step is computed as:

$$\left| \begin{array}{ll} r_k^{(i)} & = \frac{r_{k|k-1}^{(i)} \varrho_k^{(i)}}{1-r_{k|k-1}^{(i)}+r_{k|k-1}^{(i)} \varrho_k^{(i)}} \\ p_k^{(i)}(x) & = \frac{1}{\varrho_k^{(i)}} \sum\limits_{j=1}^{J} w_{k|k-1}^{(i,j)} g_k\left(x_{k|k-1}^{(i,j)}|y_k\right) \delta_{x_{k|k-1}^{(i,j)}}(x) \end{array} \right. \tag{10}$$

where $\varrho_k^{(i)} = \sum_{j=1}^{J} w_{k|k-1}^{(i,j)} g_k\left(x_{k|k-1}^{(i,j)}|y_k\right)$.

Summing up, the SMC implementation allows a recursive computation of the posterior density π_k. Then, $\forall k, \forall i$, the estimation of $x_k^{(i)}$ is obtained by computing the weighted average of the J particles $x_k^{(i,j)}$ by $w_k^{(i,j)}$.

III. Practical design

With regards to the SMC implementation described in Section II, the choices to make are investigated in this section, with a focus on proposal densities and states appearance and disappearance.

A. General setting

State model. The first choice to make concerns the state we wish to model. We choose to use in the sequel a parametric model with the case of 2D truncated Gaussian shapes, with varying positions, scales and intensities (detailed below). Hence, for all step k and sources i, $x_k^{(i)} \overset{\text{def.}}{=} \{s, \sigma, a\}_k^{(i)}$ where $s \in \mathbb{R}^2$ represents the position, $\sigma \in \mathbb{R}$ is the 2D Gaussian standard deviation, and $a \in \mathbb{R}$ is its maximum intensity.

Likelihood function. Computing the likelihood g_k (10) of an object $x_k^{(i)}$ with respect to the image y_k often leads to computational overflow, due to the large number of pixels (*e.g.* 200^2 in Section IV). An efficient alternative consists in using the product of pixel-wise likelihood ratios within the image lattice $\mathcal{S}$:

$$g_k(x_k^{(i)}|y_k) = \prod_{s' \in \mathcal{S}} \frac{p\left(y_{k,s'}|h(s', x_k^{(i)})\right)}{p(y_{k,s'})} \tag{11}$$

where $h(s', x_k^{(i)})$ represents the intensity of the 2D truncated Gaussian parametrized by $x_k^{(i)}$ at the position s':

$$h(s', x_k^{(i)}) = a_k^{(i)} \exp\left(-\frac{\|s' - s_k^{(i)}\|^2}{2\sigma_k^{(i)2}}\right) \mathbb{1}_{\left\{\frac{\left\|s'-\overline{s}_k^{(i)}\right\|^2}{2\log(100)\sigma_k^{(i)2}}<1\right\}} \tag{12}$$

The right-hand densities in (11) are Gaussian centered on the underlying intensity, whose values are either 0 or given by h and known variance.

Initial values. In order to evaluate accurately the other MBPF implementation specificities, we consider for step 0 the true state values from step 1.

Birth parameter and survival probability. We choose to associate the birth parameter $r_{C,k}^{(i)}$ (6) to the values of the weights computed in the creation step (8):

$$r_{C,k}^{(i)} = 1 \text{ if at least one weight } w_{C,k}^{(i,j)} > 0.5 \tag{13}$$

and 0 otherwise. Beside, we choose to have no prior on the survival probability of each particle $x_k^{(i,j)}$, so $p_{S,k}(x_{k-1}^{(i,j)}) \propto 1$. Here and in the following, the $\propto$ sign means the right-hand element is a distribution over the state domain $D \subset \mathbb{R}^4$.

Resampling. In most PF methods, an additional resampling step is required to avoid weights degeneracy [9]. We choose to use the stratified resampling method [10], [11] after each update step.

B. Proposal densities

1) Propagation: Target objects are assumed to move slowly from time k to time $k+1$. In practice, this means that their movement should be well captured by normal distributions centered on the previous state values. Beside, we assume the movement to be steady: for $k > 2$, speed as evaluated between $x_{k-1}^{(i)}$ and $x_{k-2}^{(i)}$ is conserved. Let us denote $\overline{x}_k^{(i)} = x_{k-1}^{(i)}$ if $k \leq 2$, and $\overline{x}_k^{(i)} = 2x_{k-1}^{(i)} - x_{k-2}^{(i)}$ otherwise:

$$q_k^{(i)}\left(x_k^{(i)}|x_{k-1}^{(i)}, y_k\right) \propto \mathcal{N}\left(\overline{x}_k^{(i)}, \Sigma\right) \tag{14}$$

$\Sigma \in \mathbb{R}^{4\times 4}$ is a diagonal covariance matrix, whose values are set accordingly to *a priori* expected values.

Track independence. As in [7], we assume the regions in the image which are influenced by the objects do not overlap. To enforce this, the propagation proposal density becomes:

$$q_k^{(i)}\left(x_k^{(i)}|x_{k-1}^{(i)},y_k\right) \propto \mathcal{N}\left(\overline{x}_k^{(i)},\Sigma\right) \times \mathbb{1}_{\left\{\overline{s}_k^{(i)}\notin \bigcup\limits_{i'\neq i} D_k^{(i')}\right\}}, \quad (15)$$

where $D_k^{(i)}$ and $\overline{s}_k^{(i)}$ represent the image coverage of state i at time k and its expected position, respectively. This coverage is chosen as a disk-region centered on $\overline{s}_k^{(i)}$, whose radius equals the expected scale $\overline{\sigma}_k^{(i)}$.

Transition density. In the following, we assume that $f_{k|k-1}(x_k^{(i)}|x_{k-1}^{(i)}) = q_k^{(i)}(x_k^{(i)}|x_{k-1}^{(i)},y_k)$ as defined in (15).

Observation dependency. The proposal density (15) does not account for the observed image y_k, and is referred as "blind" hereafter. We propose to use the intensity information as a clue for describing the position distribution. The image being noisy, intensities must be smoothed. Denoting $\mathcal{M}(y_k) \propto y_k * F$ with F a 2D Gaussian kernel of standard deviation $\Sigma_{3,3}$, the "Matched Filter" (MF) proposal density is:

$$q_k^{(i)}\left(s_k^{(i)}|x_{k-1}^{(i)},y_k\right) \propto \mathcal{N}\left(\overline{s}_k^{(i)},\Sigma\right) \times \mathbb{1}_{\left\{\overline{s}_k^{(i)}\notin \bigcup\limits_{i'\neq i} D_k^{(i')}\right\}} \times \mathcal{M}(y_k) \quad (16)$$

2) Creation: Choosing the proposal density $b_k^{(i)}$ also requires to consider track independence. The "blind" proposal density is

$$b_k^{(i)}(x_k^{(i)}|y_k) \propto \mathcal{U}_D\left(x_k^{(i)}\right) \times \mathbb{1}_{\left\{\overline{s}_k^{(i)}\notin \bigcup\limits_{i'\neq i} D_k^{(i')}\right\}} \quad (17)$$

where $\mathcal{U}_D$ is the uniform density over the state domain $D \subset \mathbb{R}^4$.

It is also possible to use image information, yielding similarly to (16) a MF proposal density:

$$b_k^{(i)}(x_k^{(i)}|y_k) \propto \mathcal{U}_D\left(x_k^{(i)}\right) \times \mathbb{1}_{\left\{\overline{s}_k^{(i)}\notin \bigcup\limits_{i'\neq i} D_k^{(i')}\right\}} \times \mathcal{M}(y_k). \quad (18)$$

Enforcing a dependency to the image in the prediction step may hinder the SMC processing in the case of non-ideal images. Thus, evaluation will address both the blind (15)(17) and the MF (16)(18) proposal densities.

C. Birth and death

The propagated object i at time k exists with probability $r_k^{(i)}$, and its created counterpart with probability $r_{C,k}^{(i)}$. The estimations of these parameters (10)(13) make possible to use them as a criterion to decide if a track appears ("birth") or disappears ("death"). More precisely, at time k and for object i, the following decision rules are followed:

$$\left|\begin{array}{l}\text{Birth if } r_{C,k}^{(i)} = 1 \\ \text{Death if } r_k^{(i)} < 0.5 \text{ and } r_{k-1}^{(i)} > 0.5\end{array}\right. \quad (19)$$

Once an object appeared, its state is appended to the estimated RFS of objects to propagate. If an object disappears its state is removed from this RFS, and is not tracked anymore.

Remark. The "merge" and "split" phenomena are not considered in this paper, but could be designed heuristically, *e.g.* deciding that a birth close to an existing trajectory is a split move. Such heuristic would need specific evaluation, which is out of the scope of this paper.

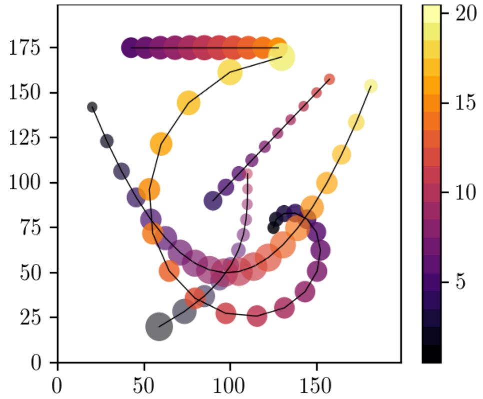


Fig. 1: Ground truth: each 2D Gaussian $x_k^{(i)}$ is represented at position $\overline{s}_k^{(i)}$ by a disk of radius $\sigma_k^{(i)}$, with color depicting k and transparency proportional to $a_k^{(i)}$. Superimposed black curves represent the trajectories of $\overline{s}_k^{(i)}$ for all objects i.

IV. Numerical Study

A. Setting

The purpose of this section is to evaluate numerically the MBPF filter when tracking Gaussian shapes in images. We use as a ground truth the position, size and intensities depicted in Figure 1: $I = 5$ objects coexist in 200^2-pixel images over $K = 20$ time steps, with two objects appearing and three objects disappearing in this interval.

We evaluate the consequences of choosing blind (15)(17) versus MF (16)(18) proposal densities. We process images corrupted by a centered Gaussian additive noise with a varying standard deviation, noted σ_{noise}. Its value is tuned according to the SNR, such that: $\text{SNR} = 20\log_{10}\left(\overline{h}/\sigma_{\text{noise}}\right)$, $\overline{h}$ being the average of non-zeros intensities in a noiseless image.

Besides, we study two scenarios, depicted in Figure 2:

- an empty background, so the image contains only objects and Gaussian noise;
- a cluttered background, scaled in intensity so that the SNR of shapes and background always equals 0 dB. This scenario is expected to hinder the filtering when using MF proposal densities (16)(18).

B. Evaluation metric

The experiments yield, for each configuration, an estimation $\hat{\mathcal{X}}$ of the true track set $\mathcal{X}$. Evaluating the correctness of $\hat{\mathcal{X}}$ with respect to $\mathcal{X}$ requires a specific metric, since at a given time the two sets may differ in cardinality: one may over- or under-estimate the number of tracks.

The most common metric employed for this purpose is the Optimal Sub-Pattern Assignment (OSPA) metric [12]. It relies on a "truncated" distance, which is defined for two states $a, b \in D \subset \mathbb{R}^4$ as $d^{(c)}(a,b) = \min(c, \|a-b\|)$, where $c > 0$ is a

ISBN 978-90-827970-1-5 © EURASIP 2018

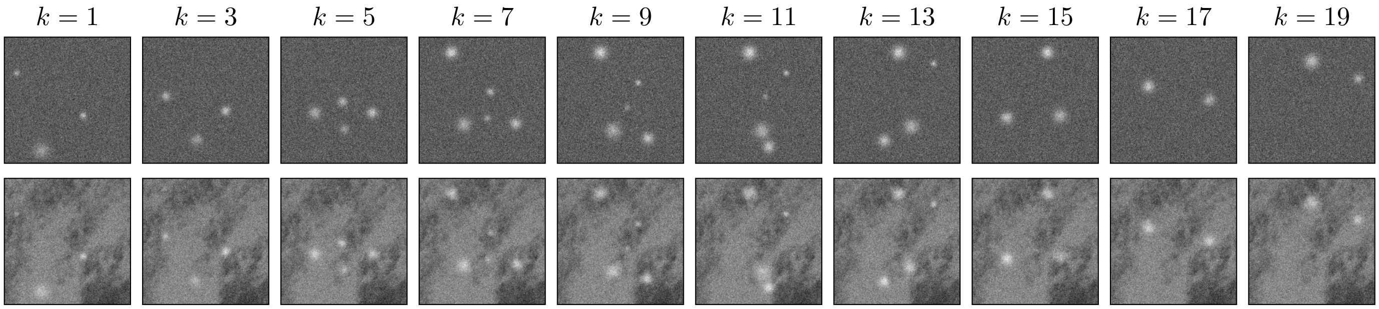


Fig. 2: Realizations of $\mathcal{Y}$, at SNR $= 0$ dB in the empty- and cluttered-background case (first and second line respectively).

distance parameter. Given two states sets $\boldsymbol{a} \in D^n$ and $\boldsymbol{b} \in D^m$ such that $m \geq n$, the OSPA metric is defined as:

$$d_p^{(c)}(\boldsymbol{a},\boldsymbol{b}) = \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m} d^{(c)}(\boldsymbol{a}_i,\boldsymbol{b}_{\pi(i)})^p + c^p(n-m)\right)\right)^{\frac{1}{p}} \tag{20}$$

where Π_n is the set of all permutations of $\{1,\ldots,n\}$, and $p \geq 1$ is an OSPA parameter.

This distance if often used as a way to account for both cardinality and value errors, balanced by the parameter c. According to [12], choosing c close to the error average magnitude imply that value error are emphasized over cardinality errors, and conversely if c is close to the domain size. However, when considering the 4-valued Gaussian we see that:

- the OSPA distance operates over heterogeneous values, *e.g.* positions and intensities;
- the typical scales are not the same along the 4 dimensions, so there is no clearly defined "average error magnitude" to set c.

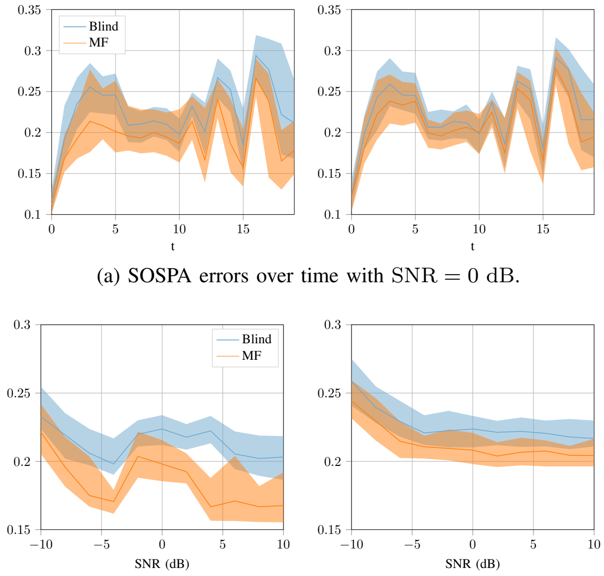
In consequence, we use a scaled version of OSPA (denoted SOSPA hereafter), in which the states values are scaled from their domain D to $[0,1]^4$. Then, we set $c = 0.5$ to balance equally cardinality and value errors, and choose $p = 1$.

C. Discussion

Figure 3 reports the averaged results over time at SNR $=$ 0 dB and the time-averaged results at all SNR. Several observations can be made:

- when processing empty-background images, the MF proposal functions yields significantly better results than its blind counterpart. This phenomenon is expected, as this situation generally favors matched-filtering methods;
- when the background is cluttered, MF proposal densities do yield higher average OSPA errors than with an empty background. However, this choice still yields better results than using the blind proposal densities.

We should however notice that errors are non-zero even in the most favorable cases. We observed two recurring phenomena: the late detection of object death, and the underestimation of object intensities which contributes notably to the SOSPA values. Future works should, among others, tackle these issues.



(a) SOSPA errors over time with SNR $= 0$ dB.

(b) Time averaged SOSPA errors with varying SNR.

Fig. 3: Average SOSPA error as evaluated on 100 filtering results, with the colored region covering the first to third quartiles. Left: empty background, right: cluttered background.

V. Conclusion

In this paper, we described thoroughly the SMC implementation of the MBPF filter for tracking multiple targets in image sequences. The design of proposal functions to account or not for the observed image was evaluated on synthetic images in favorable and unfavorable conditions. Several perspectives stem from this work, including the tracking of non-parametric shapes and the application on mesoscale convective systems in remote sensing images.

Acknowledgment

This work was completed within the French Idex with reference number ANR-10-IDEX-0001-02 PSL. The author would like to thank Jean-David Benamou, Vincent Duval, François-Xavier Vialard (Inria Paris), and Bernard Legras, Pasquale Selitto (LMD) for carefully reading this paper.

References

[1] 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.

[2] R. Mahler, “PHD filters of higher order in target number,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, 2007.

[3] A. Doucet, N. De Freitas, and N. Gordon, “An introduction to sequential Monte Carlo methods,” in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[4] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[5] 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.

[6] 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.

[7] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter, “Joint detection and estimation of multiple objects from image observations,” *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5129–5141, 2010.

[8] B.-N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo methods for multitarget filtering with random finite sets,” *IEEE Transactions on Aerospace and electronic systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[9] A. Doucet, N. De Freitas, and N. Gordon, “Sequential Monte Carlo Methods in Practice. Series Statistics For Engineering and Information Science,” 2001.

[10] G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[11] R. Douc and O. Cappé, “Comparison of resampling schemes for particle filtering,” in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. IEEE, 2005, pp. 64–69.

[12] 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.