Box-Particle PHD Filter for Multi-Target Tracking

Marek Schikora
Dept. Sensor Data and Information Fusion,
Fraunhofer FKIE, Wachtberg Germany,
marek.schikora@fkie.fraunhofer.de

Amadou Gning
Dept. of Computer Science,
University College London,
United Kingdom,
a.gning@cs.ucl.ac.uk

Lyudmila Mihaylova
School of Computing and Communications,
University of Lancaster,
United Kingdom
mila.mihaylova@lancaster.ac.uk

Daniel Cremers
Dept. Computer Science,
Technical University of Munich, Germany,
daniel.cremers@in.tum.de

Wolfgang Koch
Dept. Sensor Data and Information Fusion,
Fraunhofer FKIE, Wachtberg Germany,
wolfgang.koch@fkie.fraunhofer.de

***Abstract*—This paper develops a novel approach for multi-target tracking, called box-particle probability hypothesis density filter (box-PHD filter). The approach is able to track multiple targets and estimates the unknown number of targets. Furthermore, it is capable to deal with three sources of uncertainty: stochastic, set-theoretic and data association uncertainty. The box-PHD filter reduces the number of particles significantly, which improves the runtime considerably. The small particle number makes this approach attractive for distributed computing. A box-particle is a random sample that occupies a small and controllable rectangular region of non-zero volume. Manipulation of boxes utilizes methods from the field of interval analysis. The theoretical derivation of the box-PHD filter is presented followed by a comparative analysis with a standard sequential Monte Carlo (SMC) version of the PHD filter. To measure the performance objectively three measures are used: inclusion, volume and the optimum subpattern assignment metric. Our studies suggest that the box-PHD filter reaches similar accuracy results, like a SMC-PHD filter but with much considerably less computational costs. Furthermore, we can show that in the presence of strongly biased measurement the box-PHD filter even outperforms the classical SMC-PHD filter.**

***Index Terms*—Multi-Target Tracking, Box-Particle Filters, Random Finite Sets, PHD Filter, Interval Measurements**

I. Introduction

Multi-target tracking is a common problem with many applications. In most of these the expected target number is not known a priori, so that it has to be estimated from the measured data. In general, multi-target tracking involves the joint estimation of states and the number of targets from a sequence of observations in the presence of detection uncertainty, association uncertainty and clutter [1]. Classical approaches such as the Joint Probabilistic Data Association filter (JPDAF) [2] and multi-hypothesis tracking (MHT) [3] need in general the knowledge of the expected number of targets. The *finite set statistics* (FISST) approach proposed by Mahler [4] is a systematic treatment for multi-target tracking with an unknown and variable number of objects. To reduce the complexity Mahler proposed an approximation of the original Bayes multi-target filter, the *Probability Hypothesis Density* filter (PHD). In [5], [6] it was shown that the PHD filter outperforms the classical approaches such as the Kalman Filter, standard particle filters and the Multiple Hypothesis Tracking. Many implementations of the PHD filter have been proposed, either using sequential Monte Carlo (SMC) methods [7]–[10], or with Gaussian mixtures [11].

The traditional measurement noise expresses uncertainty due to randomness, often referred to as statistical uncertainty. In many practical applications, however, the standard measurement model is not adequate. Complex distributed surveillance systems, for example, are often operating under unknown synchronization biases and/or unknown system delays. The resulting measurements are affected by bounded errors of typically unknown distribution and biases, and can be expressed rather by intervals than by point values. An interval measurement expresses a type of uncertainty which is referred as the set-theoretic uncertainty [12], vagueness [13] or imprecision [14]. The concept of box-particle filtering in the context of tracking was introduced in [15]. In [16] it was shown that box-particles can be seen as supports of uniform probability density functions (PDF), leading to Bayesian understanding of box-particle filters. In [17] a single target box-particle Bernoulli filter with box measurements is presented.

The main contribution of this work is a derivation of box-particle methods in the context of multi-target tracking with an unknown number of targets, clutter and false alarms. We present here a box-particle version of the PHD filter. In addition, a comparison of the Box-PHD filter with a standard SMC- PHD Filter is performed. The optimum subpattern assignment (OSPA) metric [18] is used for performance measure, together with the criteria for measuring the *inclusion* of the true state and the *volume* of the posterior PDF [17].

The remaining part of this article is structured as follows. A brief introduction to FISST is given in Section II. The necessary interval methodology is explained in Section III. Section IV contains a general description of the PHD filter. The following Section V describes the steps needed to get from point particles to box-particles. The Box-PHD filter is described in Section VI. A numerical study is presented in Section VII. Conclusions are drawn in the final Section VIII.

II. Finite Set Statistics

In a single-object system, the state and measurement at time k are represented as two random vectors of possibly different dimensions. These vectors evolve in time, but maintain their initial dimension. However, this is not the case in a multi-object system. Here the multi-object state and multi-object measurement are two collections of individual objects and measurements. The number of these may change over time and lead to another dimensions of the multi-object state and multi-object measurement. Furthermore, there exist no ordering for the elements of the multi-object state and measurement. Using the theory proposed in [19], the multi-object state and measurement are naturally represented as finite subsets X_k and Z_k defined as follows:

Let $N(k)$ be a random number of objects, which are located at $\mathbf{x}_{k,1}, ..., \mathbf{x}_{k,N(k)}$ in the single-object state space E_S, e.g. R^d then,

$$X_k = \left\{\mathbf{x}_{k,1}, ..., \mathbf{x}_{k,N(k)}\right\} \in \mathcal{F}(E_S) \tag{1}$$

is the multi-object state, where $\mathcal{F}(E_S)$ denotes the collection of all finite subsets of the space E_S. Analogous to this, we define the multi-object measurement

$$Z_k = \left\{\mathbf{z}_{k,1}, ..., \mathbf{z}_{k,M(k)}\right\} \in \mathcal{F}(E_O), \tag{2}$$

assuming that at the time step k we have $M(k)$ measurements $\mathbf{z}_{k,1}, ..., \mathbf{z}_{k,M(k)}$ in the single-object space E_O, which correspond to real targets and clutter. The sets X_k and Z_k are also called random finite sets. In analogy to the expectation for a random vector, a first-order moment of the posterior distribution for a random set is of interest here, which is the so called probability hypothesis density.The integral value of the PHD over a given region in state space leads to the expected number of objects within this region. Denote $f_{k|k}(\mathbf{x}_k)$ as the PHD associated with the multi-object posterior $p(X_k|Z^k)$ at a time step k, with Z^k denoting the accumulated measurement from the time steps 1 to k. The PHD filter consists of two steps: prediction and update. The prediction can be realized through the following equation:

$$f_{k|k-1}(\mathbf{x}_k) = b(\mathbf{x}_k) + \int \left[p_s(\mathbf{x}_{k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1}) + b(\mathbf{x}_k|\mathbf{x}_{k-1})\right] f_{k-1|k-1}(\mathbf{x}_{k-1})\mathrm{d}\mathbf{x}_{k-1}, \tag{3}$$

where $b(\mathbf{x}_k)$ denotes the intensity function of spontaneous birth of new objects, $b(\mathbf{x}_k|\mathbf{x}_{k-1})$ describes the intensity function of the random finite set of objects spawned from the previous state $\mathbf{x}_{k-1}$. $p_s(\mathbf{x}_{k-1})$ is the probability that the object still exists at the time step k given its previous state $\mathbf{x}_{k-1}$, and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition probability density of the individual objects. The update equation can be written as

$$f_{k|k}(\mathbf{x}_k) \cong F(Z_k|\mathbf{x}_k) f_{k|k-1}(\mathbf{x}_k), \tag{4}$$

$$F(Z_k|\mathbf{x}_k) = 1 - p_D(\mathbf{x}_k) + \sum_{\mathbf{z}\in Z_k} \frac{p_D(\mathbf{x}_k)p(\mathbf{z}|\mathbf{x}_x)}{\lambda c(\mathbf{z}) + \int p_D(\mathbf{x}_k)p(\mathbf{z}|\mathbf{x}_k)f_{k|k-1}(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k}, \tag{5}$$

with $p_D(\mathbf{x}_k)$ denoting the probability of the detection of the state $\mathbf{x}_k$. Furthermore, $p(\mathbf{z}|\mathbf{x}_k)$ is the measurement likelihood, $c(\mathbf{z})$ the probability distribution for every clutter point and λ is the average number of clutter points per scan.

III. Interval Analysis

This section gives a short introduction to the field of interval analysis, which will be used in this article. For more informations see [20]. The original idea of interval analysis was to deal with intervals instead of real numbers for exact computation in the presence of rounding errors. However, this field has strongly increased its potential applications. We will use the main concepts to represent particles not as delta-peaks but as boxes in the state space. An interval $[x] = [\underline{x}, \overline{x}] \in \mathbb{IR}$ is a closed and connected subset of the real numbers $\mathbb{R}$, with $\underline{x} \in \mathbb{R}$ representing its lower bound and $\overline{x} \in \mathbb{R}$ its upper bound. In multiple dimensions d this interval becomes a box $[\mathbf{x}] \in \mathbb{IR}^d$ defined as a cartesian product of d intervals: $[\mathbf{x}] = [x_1] \times ... \times [x_d]$. Here the operator $|[.]|$ will be used as the size of a box $[\mathbf{x}]$. The function mid$([\mathbf{x}])$ returns the center of a box. Elementary arithmetic operations, basic functions and operations between sets have been naturally extended to the interval analysis context.

For general functions the concept of *inclusion functions* has been developed. An *inclusion function* of a given function g is defined such that the image of a box $[\mathbf{x}]$ is a box $[g]([\mathbf{x}])$ containing $g([\mathbf{x}])$. Of course, the goal is to use only inclusion functions, which are minimal in the sense that the size of the box $[g]([\mathbf{x}])$ is minimal but still covers the whole image of a box $[\mathbf{x}]$. An important class in the context of tracking are the *natural inclusion functions*.

Theorem 1. *Assume $g\colon \mathbb{R}^d \to \mathbb{R}$, $(x_1, ..., x_d) \mapsto g(x_1, ..., x_d)$ is a function expressed as a finite composition of the operators $+, -, *, /$ and elementary functions (sin, cos, exp, ...). A* ***natural inclusion function*** *is obtained by replacing each real variable and each operator or function by its interval counterpart.*

In general, natural inclusion functions are not minimal, but many functions can be modified in order to satisfy the conditions in the following theorem and then their natural inclusion functions are minimal. Proofs and examples can be found in [20].

Definition 1. *An inclusion function $[g]$ for g is convergent if, for any sequence of boxes $[\mathbf{x}](k)$,*

$$\lim_{k\to\infty} |[\mathbf{x}](k)| = 0 \Rightarrow \lim_{k\to\infty} [g]([\mathbf{x}](k)) = 0. \tag{6}$$

Theorem 2. *If g involves only continuous operators and continuous elementary functions then $[g]$ is convergent. If,* *furthermore, each of the variables $x_1, ..., x_2$ occurs at most once in the formal expression of g, then $[g]$ is minimal.*

The next needed concept is *contraction*, which will be used in the definition of likelihood functions and the update step of the proposed filters. A Constraint Satisfaction Problem (CSP), often denoted by $\mathcal{H}$, can be written as:

$$\mathcal{H} = (\mathbf{g}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in [\mathbf{x}])\,. \tag{7}$$

A common interpretation of (7) is: find the optimal box enclosure of the set of vectors $\mathbf{x}$ belonging to a given prior domain $[\mathbf{x}]$ satisfying a set of m contraints $\mathbf{g} = (g_1, ..., g_m)^T$, with g_i a real valued function. The solution consists of all $\mathbf{x}$, that satisfy $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ or written as a set: $\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{g}(\mathbf{x}) = \mathbf{0}\}$. A *contraction* of $\mathcal{H}$ means replacing $[\mathbf{x}]$ by a smaller box $[\mathbf{x}]'$ under the constraint $\mathbb{S} \subseteq [\mathbf{x}]' \subseteq [\mathbf{x}]$. There are several methods to build a contractor for $\mathcal{H}$, e.g. by the Gauss elimination, Gauss-Seidel algorithm and linear programming. In this work, however, we will use *Constraint Propagation* (CP), or sometimes referred as forward-backward propagation, for its good suitability in the context of tracking problems. An example of a CP algorithm is given in [21].

IV. The SMC–PHD Filter

Inspired by the works of Vo et al. [9] and Ristic et al. [10] on efficient sequential Monte Carlo methods for the PHD filter we present here, to make this article self-contained, briefly an improved SMC-PHD filter. The main improvements are a measurement steered particle placement for target birth. In addition, a target state and covariance matrix estimation without the need of clustering is introduced. The state of an individual object will be represented by $\mathbf{x}_k \in \mathbb{R}^{n_x}$. and each measurement as $\mathbf{z}_k \in \mathbb{R}^{n_z}$. For the sake of simplicity, we assume that the object motion model of each target is linear with a constant velocity. With this, the object state prediction can be written as: $\mathbf{x}_k = \Phi(k, k-1)\mathbf{x}_{k-1} + \mathbf{s}_k$, with $\mathbf{s}_k$ a zero mean Gaussian white process noise and $\Phi(k, k-1)$ the transition matrix from time step $k-1$ to k.

The SMC-PHD filter can be summarized in 6 steps, which will be presented in the following. Here the particle set represents the target intensity $f_{k|k}(\mathbf{x})$ of the PHD filter, which corresponds to the multi-target state. Recall that the integral over this intensity (or sum, if using particles) is the estimated expected number of targets and it is not necessary equal to one. Given from the previous time step we have the particle set:

$$\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k}, \tag{8}$$

with $\mathbf{x}_i \in \mathbb{R}^{n_x}$, w_i the corresponding weight and N_k denoting the number of particles, estimated at time step t_{k-1}. The implementation details using a particle representation are presented in the following.

1) **Predict target intensity**
The resampled particle set gained from the previous step is denoted by $\{\mathbf{x}_i, w_i\}_{i=1}^{N_k}$. These particles represent the intensity over the state space. Another interpretation is, that every particle represents a possible target state (called microstates in the language of thermodynamics), so that the prediction of the whole set can be modeled by applying a transition model to every particle and adding some noise to it. The weights are unchanged. In practical implementations this has the same effect as predicting the intensity distribution over the state space with a closed formula.
In order to avoid a high number of additional particles $N_{k,new}$ the authors in [10] propose to sample new born particles according to the measurements $\boldsymbol{Z}_{k-1}$ from the previous time step. Let m_{k-1} denote the number of measurements in time step t_{k-1}, then for each of these we sample

$$N_{k,new}^{j} = \lceil N_{k,new}/m_{k-1} \rceil, \quad j = 1, ..., m_{k-1} \tag{9}$$

many particles $\tilde{\mathbf{x}}_i$ drawn from the distribution $\mathcal{N}([\mathbf{z}_j^{k-1}), \boldsymbol{\Sigma})$, centered around an old measurement $\mathbf{z}_j^{k-1}$ with the measurement covariance matrix $\boldsymbol{\Sigma}$. The weights of the new born particles are set to

$$w_i = \frac{p_b}{N_{k,new}}, \quad i = 1, ..., N_{k,new}, \tag{10}$$

with p_b the probability of birth. We define $\{\tilde{\mathbf{x}}_i, w_i\}_{i=1}^{N_k+N_{k,new}}$ as the predicted particle set containing the new born and persistent particles.

2) **Compute Correction Term**
For all new measurements $\mathbf{z}_j$, with $j = 1, ..., m_k$ compute:

$$\lambda_{k|k-1}(\mathbf{z}_j) = \lambda c(\mathbf{z}_j) + \sum_{i=1}^{N_k+N_{k,new}} p_k(\mathbf{z}_j \mid \tilde{\mathbf{x}}_i) p_k^D(\tilde{\mathbf{x}}_i) w_i \tag{11}$$

3) **Estimate target states**
To avoid a clustering step we use the methodology presented in [10]. First, compute the following weights for all new measurements $\mathbf{z}_j, j = 1, ..., m_k$ and all persistent particles, i.e. not the new born, $\mathbf{x}_i, i = 1, ..., N_k$.

$$w_{j,i} = \frac{p_k(\mathbf{z}_j \mid \tilde{\mathbf{x}}_i) p_k^D(\tilde{\mathbf{x}}_i)}{\lambda_{k|k-1}(\mathbf{z}_j)} \cdot w_i \tag{12}$$

Then compute the following sum

$$W_j = \sum_{i=1}^{N_k} w_{j,i}, \tag{13}$$

which can be seen as a probability of existence for target j, similarly to the multi-target multi-Bernoulli filter. For further analysis, only those j for which W_j is above a specified threshold τ are considered, i.e.

$$\mathcal{J} = \{j | W_j > \tau, j = 1, ..., m_k\} \tag{14}$$

For all $j \in \mathcal{J}$ the estimated point states are then:

$$\hat{\mathbf{y}}_j = \sum_{i=1}^{N_k} \tilde{\mathbf{x}}_i \cdot w_{j,i}. \tag{15}$$

Note that only targets that have been detected at time step t_k can be reported as present. This follows the lack of "memory" of a PHD filter. The full characteristics are discussed in [22]. In practice τ is usually set as $\tau = 0.75$.

4) **Estimate covariance matrices**
For each estimated state $\hat{\mathbf{y}}_j$ compute its covariance matrix:

$$\mathbf{P}_j = \sum_{i=1}^{N_k} w_{j,i} \left[(\tilde{\mathbf{x}}_i - \hat{\mathbf{y}}_j)(\tilde{\mathbf{x}}_i - \hat{\mathbf{y}}_j)^T\right], \tag{16}$$

The matrix $\mathbf{P}_j$ is not an error covariance matrix in the sense of single target Bayes filtering, but it characterizes the particle distribution of state $\hat{\mathbf{y}}_j$.

5) **Update**
Given m_k new measurements the update of the state intensity is realized through a correction of the individual particle weights. For every particle $(\mathbf{x}_i, w_i)$, with $i = 1, .., N_k + N_{k,new}$ set:

$$\hat{w}_i = \left[(1 - p_k^D(\tilde{\mathbf{x}}_i)) + \sum_{j=1}^{m_k} \frac{p_k(\mathbf{z}_j \mid \tilde{\mathbf{x}}_i) p_k^D(\tilde{\mathbf{x}}_i)}{\lambda_{k|k-1}(\mathbf{z}_j)}\right] \cdot w_i \tag{17}$$

6) **Resampling**
Compute first the estimated expected number of targets

$$\eta_k = \sum_{i=1}^{N_k + N_{k,new}} \hat{w}_i. \tag{18}$$

Let N_{k+1} be the number of resampled particles, then any standard resampling technique for particle filtering can be used. Rescale the weights by η_k to get a new particle set $\{\mathbf{x}_i, \eta_k / N_{k+1}\}_{i=1}^{N_{k+1}}$.

V. From Particles to Boxes

A popular class of methods for implementation of Bayes-like filters are particle filters [23]. Applying these methods to the PHD filter leads to a particle approximation of the intensity $f_{k|k}(\mathbf{x})$ with a set of N_k weighted random samples $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k}$. The approximation can be written as:

$$f_{k|k}(\mathbf{x}) \approx \sum_{i=1}^{N_k} w_i \delta_{\mathbf{x}_i}(\mathbf{x}), \tag{19}$$

with $\delta_{\mathbf{x}_i}(\mathbf{x})$ the Dirac delta function concentrated at $\mathbf{x}_i$. The sum (19) converges to $f_{k|k}(\mathbf{x})$, with $N_k \to \infty$ [24]. The number of particles used is a key issue to the overall filter performance. In general, the higher the number of particles, the better the approximation and with it the performance. However, a high number leads often to a computationally demanding scenario. In [15] the authors presented a natural way to deal with the decrease of N_k by using boxes instead of point particles and combining particle filter techniques with interval analysis methods. Moreover, in [16] the authors propose to interpret box-particles as supports of uniform PDFs, so that (19) changes to:

$$f_{k|k}(\mathbf{x}) \approx \sum_{i=1}^{N_k} w_i U_{[\mathbf{x}_i]}(\mathbf{x}), \tag{20}$$

with $U_{[\mathbf{x}_i]}(\mathbf{x})$ denoting the uniform PDF over the box $[\mathbf{x}_i]$.

Bayes–like filters require the knowledge of the measurement likelihood function $p(\mathbf{z}|\mathbf{x})$. This is also true for the PHD filter. A likelihood returns values in the interval $[0, 1]$. The returned value depends on the probability that this measurement $\mathbf{z}$ was produced by the state $\mathbf{x}$. In the context of this article we assume also box measurements $[\mathbf{z}]$. We do not need to model the statistical sensor error with some error density (that in practice is mostly unknown) and we do not need to model systematical errors directly. With this assumption the only information needed from a sensor is its error range. In [16] the measurement likelihood for box measurements is derived and given as

$$p([\mathbf{z}] \mid [\mathbf{x}]) := \frac{|[\mathbf{h}_{\text{CP}}]([\mathbf{x}], [\mathbf{z}])|}{|[\mathbf{x}]|}. \tag{21}$$

The function $[h_{\text{CP}}]([\mathbf{x}], [\mathbf{z}])$ returns a contracted version of $[\mathbf{x}]$ under the constraints given by the measurement function $\mathbf{h}(\mathbf{x}) = \mathbf{z}$. An example of this contraction step can be found in [21].

VI. The Box–PHD Filter

Similarly to the scheme of the SMC-PHD filter this section presents the box-particle implementation.

The box-PHD filter can be summarized in 7 steps, which will be presented in the following. Here the box-particle set represents the target intensity of the PHD. Again, the integral over this intensity (or sum, if using particles) is the estimated expected number of targets and it is not necessary equal to one. Given from the previous time step we have the particle set:

$$\{([\mathbf{x}_i], w_i)\}_{i=1}^{N_k}, \tag{22}$$

with $[\mathbf{x}_i] \in \mathbb{IR}^{n_x}$, w_i the corresponding weight and N_k denoting the number of particles, estimated at time step t_{k-1}.

The implementation details using a box-particle representation are presented below. Step 1 corresponds to the prediction phase, steps 2-5 to the correction phase and steps 6 and 7 to the resampling phase of a sequential Monte Carlo algorithm.

1) **Predict target intensity**
The resampled particle set gained from the previous step is denoted by $\{[\mathbf{x}_i], w_i\}_{i=1}^{N_k}$. These particles are propagated using a evolution or motion model and inclusion functions. In order to avoid a high number of additional particles $N_{k,new}$, we will sample new born particles according to the measurements from the previous time step $\boldsymbol{Z}_{k-1}$. Let m_{k-1} denote the number of measurements in time step t_{k-1}, then for each of these we sample

$$N_{k,new}^j = \lceil N_{k,new} / m_{k-1} \rceil, \quad j = 1, ..., m_{k-1} \tag{23}$$

many particles $\tilde{\mathbf{x}}_i$ drawn from the distribution $\mathcal{N}([\mathbf{z}_j^{k-1}]), \mathbf{\Sigma})$, centered around an old measurement $[\mathbf{z}_j^{k-1}]$ with the measurement covariance matrix $\mathbf{\Sigma}$. The weights of the new born particles are set to

$$w_i = \frac{p_b}{N_{k,new}}, \quad i = N_k + 1, ..., N_{k,new}, \tag{24}$$

with p_b the probability of birth. We define $\{[\tilde{\mathbf{x}}_i], w_i\}_{i=1}^{N_k+N_{k,new}}$ as the predicted particle set containing the new born and persistent particles.

2) **Compute Correction Term**
For all new measurements $[\mathbf{z}_j]$, with $j = 1, ..., m_k$ compute:

$$\lambda_{k|k-1}([\mathbf{z}_j]) = \lambda c([\mathbf{z}_j]) + \sum_{i=1}^{N_k+N_{k,new}} p_k([\mathbf{z}_j] \mid [\tilde{\mathbf{x}}_i]) p_k^D([\tilde{\mathbf{x}}_i]) w_i \tag{25}$$

3) **Estimate target states**
To avoid a clustering step we use the methodology presented in [10]. First, compute the following weights for all new measurements $[\mathbf{z}_j], j = 1, ..., m_k$ and all persistent particles, i.e. not the new born, $[\mathbf{x}_i], i = 1, ..., N_k$.

$$w_{j,i} = \frac{p_k([\mathbf{z}_j] \mid [\tilde{\mathbf{x}}_i]) p_k^D([\tilde{\mathbf{x}}_i])}{\lambda_{k|k-1}([\mathbf{z}_j])} \cdot w_i \tag{26}$$

Then compute the following sum

$$W_j = \sum_{i=1}^{N_k} w_{j,i}, \tag{27}$$

which can be seen as a probability of existence for target j, similarly to the multi-target multi-Bernoulli filter. For further analysis only those j are considered for which W_j is above a specified threshold τ, i.e.

$$\mathcal{J} = \{j | W_j > \tau, j = 1, ..., m_k\} \tag{28}$$

For all $j \in \mathcal{J}$ the estimated point states are then:

$$\hat{\mathbf{y}}_j = \frac{1}{W_j} \sum_{i=1}^{N_k} \text{mid}([\tilde{\mathbf{x}}_i]) \cdot w_{j,i}. \tag{29}$$

For all $j \in \mathcal{J}$ the estimated box states are then:

$$[\hat{\mathbf{y}}_j] = \frac{1}{W_j} \sum_{i=1}^{N_k} [\tilde{\mathbf{x}}_i] \cdot w_{j,i}. \tag{30}$$

In Equations (29) and (30) we added, in contrast to [10], the normalization term $\frac{1}{W_j}$ to receive more accurate state estimates when W_j is not practically one.

4) **Estimate covariance matrices**
Using the interpretation of box-particles as a mixture of uniform PDFs, the covariance matrix for each state is computed as

$$\mathbf{P}_j = \sum_{i=1}^{N_k} \frac{w_{j,i}}{W_j} \left[(\text{mid}([\tilde{\mathbf{x}}_i]) - \hat{\mathbf{y}}_j)(\text{mid}([\tilde{\mathbf{x}}_i]) - \hat{\mathbf{y}}_j)^T + \mathbf{\Sigma}_{\mathbf{U_i}} \right], \tag{31}$$

with $\mathbf{\Sigma}_{\mathbf{U_i}}$ a diagonal matrix of the form

$$\mathbf{\Sigma}_{\mathbf{U_i}} = \begin{pmatrix} |([\mathbf{x}_i])_1|^2/12 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & |([\mathbf{x}_i])_{n_x}|^2/12 \end{pmatrix} \tag{32}$$

containing the standard derivations for the individual uniform distributions. In Equation (31) we added, in contrast to [10], the normalization term $\frac{1}{W_j}$ to receive more accurate covariance matrix estimates when W_j is not practically one. The matrix $\mathbf{P}_j$ is not an error covariance matrix in the sense of single target Bayes filtering, but it characterizes the particle distribution of state $\hat{\mathbf{y}}_j$.

5) **Update**
Given m_k new measurements the update of the state intensity is realized through a correction of the individual particle weights. For every particle $([\mathbf{x}_i], w_i)$, with $i = 1, .., N_k + N_{k,new}$ set:

$$\hat{w}_i = \left[(1 - p_k^D([\tilde{\mathbf{x}}_i])) + \sum_{j=1}^{m_k} \frac{p_k([\mathbf{z}_j] \mid [\tilde{\mathbf{x}}_i]) p_k^D([\tilde{\mathbf{x}}_i])}{\lambda_{k|k-1}([\mathbf{z}_j])} \right] \cdot w_i \tag{33}$$

6) **Contract particles**
In [16], interpreting box-particles as a mixture of uniform PDFs, it was proven that contraction steps are needed in the measurement update step. Hence, we contract every box-particle $[\mathbf{x}_i], i = 1, ..., N_k + N_{k,new}$ with its corresponding measurement. The corresponding measurement is defined through:

$$[\mathbf{z}] = \arg\max_{w_{j,i}} \{[\mathbf{z}_j], w_{j,i} > 0\}. \tag{34}$$

If no $[\mathbf{z}]$ is found, this particle is not contracted, else the particle i is set to

$$([\hat{\mathbf{x}}_i], \hat{w}_i), \text{ with} \tag{35}$$

$$[\hat{\mathbf{x}}_i] = [\mathbf{h}_{\text{CP}}]([\tilde{\mathbf{x}}_i], [\mathbf{z}]). \tag{36}$$

7) **Resampling**
Compute first the estimated expected number of targets

$$\eta_k = \sum_{i=1}^{N_k+N_{k,new}} \hat{w}_i. \tag{37}$$

Let N_{k+1} be the number of resampled particles. As explained in [16], instead of replicating box-particles which have been selected more than once in the resampling step, we divide them into smaller box-particles as many times as they were selected. Several strategies of subdivision can be used (e.g. according to the largest box face). In this paper we randomly pick a dimension to be divided for the selected box-particle. Next, rescale the weights by η_k to get a new particle set $\{[\mathbf{x}_i], \eta_k / N_{k+1}\}_{i=1}^{N_{k+1}}$.

VII. Numerical Studies

This section gives numerical studies for the proposed Box-particle PHD filter algorithm. For comparison with traditional particle filter techniques we use a point particle sequential Monte Carlo PHD filter (SMC-PHD). As performance measure the optimum subpattern assignment (OSPA) metric [18] is used for performance measure, together with the criteria for measuring the *inclusion* of the true state and the *volume* of the posterior PDF. The later two were introduced in [17], [21].

A. Testing Scenario

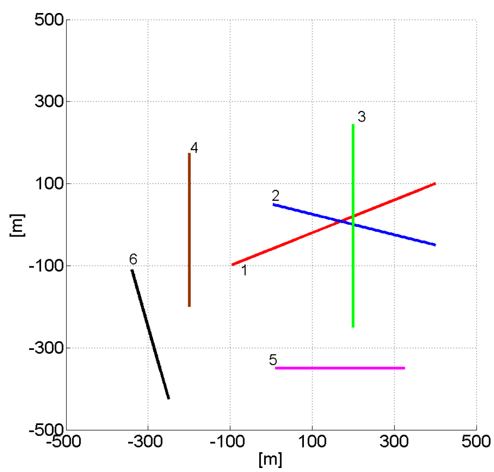

Fig. 1. Linear scenario used for performance evaluation. Six targets move inertially. The individual starting points of each target correspond to the denoted target ID number.

We analyze the behavior of both filters in a demanding linear scenario. Herein six inertial moved targets are placed in an area $A = [-500, 500]m \times [-500, 500]m$. The unit is assumed to be meters. The state space is $\mathcal{S} \subset \mathbb{R}^4$, where the first two components correspond to the x and y coordinates and the third and fourth their velocities. The measurement space consists of $[x]$ and $[y]$ measurements, so $\mathbf{Z} \subset \mathbb{IR}^2$. New measurements occur for the sake of simplicity every second. The measurement noise is white Gaussian noise with a standard deviation $\sigma_x = \sigma_y = 15m$. The probability of detection is set equal for all states to $p_k^D([\mathbf{x}]) = 0.95$. Target placement and direction of movement is visualized in Figure 1. Targets 1 – 3 are present for all time steps. Target 4 is presented between time step 15 and 90. Target 5 and 6 are present between time step 30 and 75. The whole scenario has a length of 100 time steps (seconds). The number of clutter measurements n_c is estimated following a Poisson distribution with the mean value $|A| \cdot \rho_A$:

$$p(n_c) = \frac{1}{n_c!}(|A| \cdot \rho_A)^{n_c} \exp(-|A| \cdot \rho_A), \qquad (38)$$

with $|A|$ denoting the volume of a observed area and ρ_A a parameter describing the clutter rate. For this scenario we used $\rho_A = 4 \cdot 10^{-6}$. Clutter measurements are generated by an i.i.d. process.

To initialize the particle cloud at time step $t_k = 0$, $N_0 \in \mathbb{N}^+$ particles are distributed uniformly across the state space $\mathcal{S}$, e.g. $N_0 = 1000$. The weights are set to $w_i = 1/N_0$.

Assuming a constant velocity model in two dimensions the prediction of the persistent particles can be modeled by:

$$[\tilde{\mathbf{x}}_i] = \begin{pmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} [\mathbf{x}_i] + [\boldsymbol{\nu}], \qquad (39)$$

with $\Delta_t = t_k - t_{k-1}$ and $\boldsymbol{\nu}$ a 3σ interval of some white process noise, defined by a covariance matrix $\boldsymbol{\Sigma}$. Hidden in equation (39) are inclusion functions for the individual dimension of the state space. A close look reveals that every variable only appears once (for each dimension) and that all operations are continuous, so these natural inclusion functions are minimal and the propagated boxes have minimal size. This fact holds for constant velocity models with arbitrary dimensions.

B. Performance Measures

For the OSPA metric we use directly the state estimates if using the SMC-PHD filter. To apply the OSPA metric to the Box-PHD filter we use the point state estimates $\hat{\mathbf{y}}_j$ gained in Equation (29) of the proposed algorithm. Alternatively, one can use the center points of the box states $\text{mid}([\hat{\mathbf{y}}_j])$, which have the same values as $\hat{\mathbf{y}}_j$. The *inclusion* value ρ measurers if the state vector is contained in the support of the posterior PDF, or in the case of the PHD filter the posterior intensity. Given the ground truth for all targets $\mathbf{y}_l^*$, with l a index over the true number of targets, the inclusion for the SMC-PHD filter can computed by evaluating:

$$\rho_l^{\text{SMC}} = \begin{cases} 1 & \exists j : (\hat{\mathbf{y}}_j - \mathbf{y}_l^*)\mathbf{P}_j^{-1}(\hat{\mathbf{y}}_j - \mathbf{y}_l^*)^T < \kappa \\ 0 & \text{otherwise.} \end{cases} \qquad (40)$$

The condition in (40) checks if the ground truth is contained in the error ellipse defined by covariance matrix $\mathbf{P}_j$. The term κ defines the size of the error ellipse, e.g., use $\kappa = 11.8$ for a 3σ–ellipse in two dimensions [25]. The *inclusion* for the Box-PHD filter is much simpler to compute: Check if the ground truth $\mathbf{y}_l^*$ is contained in one of the state boxes $[\hat{\mathbf{y}}_j]$. If this is true the inclusion value is one, otherwise zero. Then ρ_l for the box-PHD filter is given by:

$$\rho_l^{\text{box}} = \begin{cases} 1 & \text{for } \mathbf{y}_l^* \in [\hat{\mathbf{y}}_j] \text{ and} \\ 0 & \text{otherwise.} \end{cases} \qquad (41)$$

The *volume* criteria measures the spread of the particle distribution for a given state. To have a fair comparison between both filters we compute the volume for the SMC-PHD filter as:

$$\nu_j^{\text{SMC}} = \sqrt{6 \cdot \sqrt{\mathbf{P}_j(1,1)} + 6 \cdot \sqrt{\mathbf{P}_j(2,2)}}. \qquad (42)$$

The volume in Equation (42) is the square route of the widths of a box containing the 3σ–ellipse of state j. Note that we only consider here the position information, since the entries of $\mathbf{P}_j$ have different units. For the Box-PHD filter the volume is computed as the square root of the widths of the box states, giving:

$$\nu_j^{\mathrm{SMC}} = \sqrt{|[\hat{\mathbf{y}}_i](1)| + |[\hat{\mathbf{y}}_i](2)|} \tag{43}$$

C. Experiments

1) Accuracy Test: In the first experiment we investigate the accuracy achieved with the Box-PHD filter in comparison with the SMC-PHD filter. To do so we will use the linear scenario described earlier. A visualization of the Box-PHD filter for the linear scenario can be seen in Figure 2. Figure 3 visualizes the mean OSPA values achieved with both filters on the given scenario. We can observe that the OSPA values are in general very low. This means that the SMC-PHD filter and the Box-PHD filter behave very good in this scenario. However, we can also observe that the Box-PHD filter has a little higher values than the SMC-PHD filter. The authors of [17] already noticed that point estimates gained from box-particles can have a slight bias. Therefore they introduced two new measurements criteria *inclusion* and *volume*. The mean results for 1000 Monte Carlo trials and all targets are shown in Figures 4 and 5, respectively. It can be easily seen that the inclusion and volume values react to target appearance and target disappearance. In general we can say that the Box-PHD filter has a higher volume then the SMC-PHD filter. This can be seen as a drawback of the box-particle technique. However, a closer look on the inclusion values reveals that the higher volume leads to better values for the inclusion criteria. So we can state that the SMC-PHD filter converges to fast and therefore it can happen sometimes that the true target state is not in the support of any covariance matrix $\mathbf{P}_j$. From an engineering point of view both filters reach similar results in this scenario. This fact can also be seen in Figure 6. Here, the estimated mean number of states is depicted. The curves of both filters are practically identical. Nevertheless, the number of particles needed for the Box-PHD filter is much smaller in comparison with the SMC-PHD filter, which yields in a better runtime.The mean speedup factor for the Box-PHD filter is 10.9. The number of particles used in this scenario where 1875 for the SMC-PHD filter and only 63 for the Box-PHD filter. This reduces the average computation time for one update step from 10.35 msec to 0.95 msec.

2) Strong Bias: In the next experiment we investigate the behavior of both filter when the sensor measurements have a strong bias, i.e., the bias is bigger then the white process noise of the sensor. Again we used the linear scenario, but we added to every measurement a bias of 30 for the x measurement and a bias of 10 for the y measurement. The volume of both filters does not change.The inclusion criteria on the other hand changes dramatically for the SMC-PHD filter the value drops to values around 0.5, c.f. Figure 7. This means that approximately 50% of the time the true target state is not within the posterior intensity of the filter. This indicates filter divergence, which is considered a *catastrophic* event in target tracking. The Box-PHD filter, on the other hand, reaches values similar to the first experiment without bias. These result lead to the conclusion that the box-particle filter can outperform a traditional point particle filter in scenarios with strongly biased measurements.

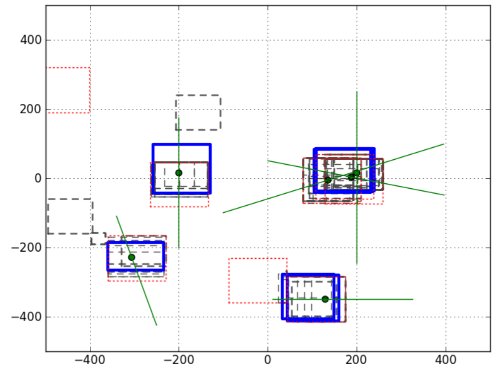

Fig. 2. Visualization of proposed Box-PHD filter. The green solid lines are the true target trajectories. The blue solid boxes correspond to a projection of the estimated box states into 2D. The box-particles are visualized as dashed black boxes, while red dotted boxes are the measurements.

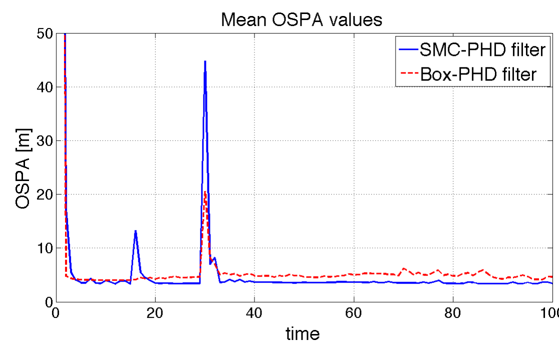

Fig. 3. Mean OSPA values for 1000 Monte Carlo trials on linear scenario for both filters.

VIII. Conclusion

In this paper we presented a novel technique for non-linear multi-target tracking with a box-particle based filter, called the Box-PHD filter. The theoretical backbone of this is the random finite set theory, which can be used to derive the general intensity filter equations. For the implementation, however, methods from interval analysis are used additionally

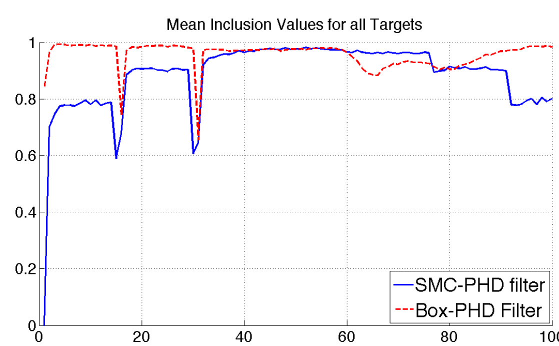

Fig. 4. Mean inclusion values for 1000 Monte Carlo trials and all targets on linear scenario without biased measurements for both filters.

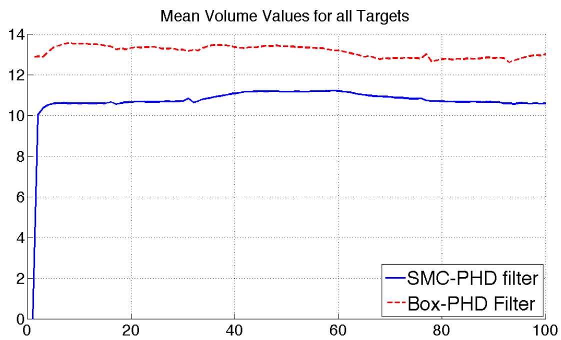

Fig. 5. Mean volume values for 1000 Monte Carlo trials and all targets on linear scenario without biased measurements for both filters.

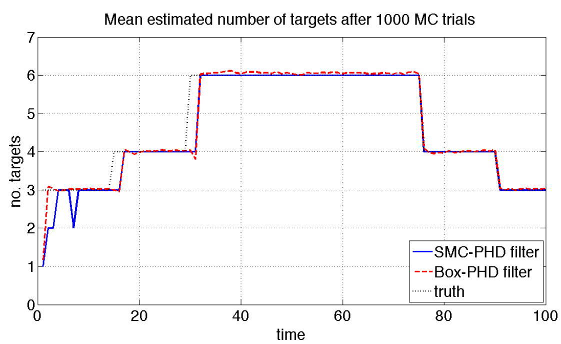

Fig. 6. Mean estimated number of states for 1000 Monte Carlo trials on linear scenario.

to get a box-particle representation of the PHD filter. This representation allows a decrement of the number of particles needed. In our experiments we could reduce the number of particles by a factor of approximately thirty and reduce the computation time by a factor of approximately eleven. On the other hand, the accuracy of the filter was not reduced. Especially in the presence of strong bias we could show that the Box-PHD filter can outperform the SMC-PHD filter.

Acknowledgment

We acknowledge the support of the European Community's Seventh Framework Program [FP7/2007-2013] under grant agreement No 238710 (Monte Carlo based Innovative Management and Processing for an Unrivalled Leap in Sensor Exploitation).

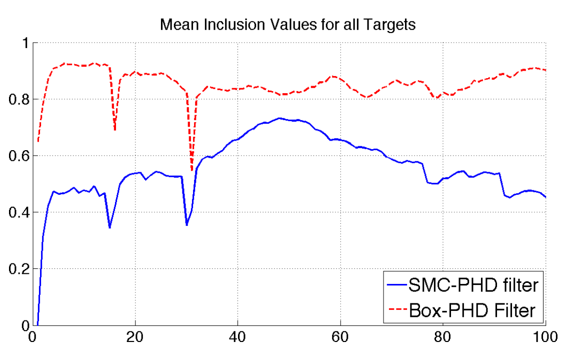

Fig. 7. Mean inclusion values for 1000 Monte Carlo trials and all targets on linear scenario with biased measurements for both filters.

References

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. San Diego, CA: Academic, 1988.

[2] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, pp. 173–184, 1983.

[3] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, 1979.

[4] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. AES*, vol. 4, no. 39, pp. 1152–1178, 2003.

[5] K. Panta, B.-N. Vo, S. Singh, and A. Doucet, "Probability hypothesis density filter versus miltiple hypothesis tracking," *Proc. SPIE*, vol. 5429, pp. 284–295, 2004.

[6] R. Juang and P. Burlina, "Comparative performance evaluation of GM-PHD filter in clutter," in *FUSION*, July 2009.

[7] H. Sidenbladh, "Multi-target particle filtering for probability hypothesis density," in *International Conference on Information Fusion*, Cairns, Australia, 2003, pp. 800–806.

[8] T. Zajic and R. Mahler, "A particle-systems implementation of the PHD multi-target tracking filter," *Proc. SPIE, Signal Processing, Sensor Fusion Target Recognition XII*, vol. 5096, no. 4, pp. 291–299, 2003.

[9] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, 2005.

[10] B. Ristic, D. Clark, and B.-N. Vo, "Improved SMC implementation of the PHD filter," in *Proc. of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

[11] B.-T. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 4091–4104, 2006.

[12] P. Combettes, "Foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, no. 2, pp. 182–208, 1993.

[13] R. Kruse, E. Schwecke, and J. Heinsohn, *Uncertainty and vagueness in knowledge based systems*. Springer-Verlag, 1991.

[14] P. Smets, "Imperfect information: Imprecision and uncertainty," *Uncertainty Management in Information Systems*, pp. 225–254, 1996.

[15] F. Abdallah, A. Gning, and P. Bonnifait, "Box particle filtering for nonlinear state estimation using interval analysis," *Automatica*, vol. 44, pp. 807–815, 2008.

[16] A. Gning, L. Mihaylova, and F. Abdallah, "Mixture of uniform probability density functions for non linear state estimation using interval analysis," in *Proc. of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

[17] A. Gning, B. Ristic, and L. Mihaylova, "A box particle filter for stochastic set-theoretic measurements with association uncertainty," in *Proc. of the 14th International Conference on Information Fusion*, Chicago, IL, USA, July 2011.

[18] D. Schumacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, 2008.

[19] R. Mahler, "Multitarget Bayes filtering via first-order multitargets moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2003.

[20] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer, 2001.

[21] A. Gning, B. Ristic, and L. Mihaylova, "Bernoulli particle / box-particle filters for detection and tracking in the presence of triple measurement uncertainty," *IEEE Trans. Signal Process.*, 2012.

[22] O. Erdinc, P. Willett, and Y. Bar-Shalom, "Probability hypothesis density filter for multitarget multisensor tracking," in *Proc. of the 8th International Conference on Information Fusion (FUSION)*, Philadelphia, PA, USA, July 2005.

[23] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.

[24] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practinioners," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, 2002.

[25] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambrige University Press, 1992.