

# Sequential Monte Carlo Method for the iFilter

Marek Schikora, Wolfgang Koch  
SDF Dept.  
Fraunhofer FKIE  
Wachtberg, Germany  
Email: {marek.schikora,  
wolfgang.koch}@fkie.fraunhofer.de

Roy L. Streit  
Metron, Inc.  
Reston, VA, U.S.A  
Email: r.streit@ieee.org

Daniel Cremers  
Computer Science Dept.  
Tech. University of Munich  
Garching, Germany  
Email: cremers@tum.de

**Abstract**—Poisson point processes (PPP's) are very useful theoretical models for diverse applications. One of those is multi-target tracking of an unknown number of targets, leading to the intensity filter (iFilter) as a generalization of the probability hypothesis density (PHD) filter. This article develops a sequential Monte Carlo (SMC) implementation of the iFilter. In theory it was shown that the iFilter can estimate a clutter model from the measurements and thus does not need it as a-priori knowledge, like the PHD filter does. Our studies show that this property holds not only in simulations but also in real world applications. In addition it can be shown, that the performance of the PHD filter decreases substantially if the a-priori knowledge of the clutter intensity is chosen incorrectly.

**Keywords:** Intensity Filter, Sequential Monte Carlo, Multi-target tracking, PHD Filter, Poisson point processes (PPP's)

## 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 from a sequence of observations in the presence of detection uncertainty, association uncertainty and clutter [1]. Classical approaches like the Joint Probabilistic Data Association filter (JPDAF) [2] and multi hypothesis tracking (MHT) [3] need in general the knowledge of the expected target number. In recent time the intensity filter (iFilter) [4], [5] has been presented as a generalization of the probability density hypothesis (PHD) filter [6]. Both filters use multi-target and multi-measurement states along with the estimation of the number of target. While the PHD filter was originally derived using finite set statistics the iFilter was derived through Poisson point processes (PPP's). Furthermore it was shown, that the PHD filter is a special case of the iFilter [7]. From an engineering field of view the main difference between both is the clutter model, which has to be known for the PHD filter a-priori and is estimated by the iFilter. For the PHD filter many implementations, either using sequential Monte Carlo methods [8]–[10] or closed form solutions [11], have been presented, an implementation and analysis of the iFilter was till now not published.

The main contribution of this work is the first presentation of an implementation scheme for the iFilter using a sequential Monte Carlo method, also called particle filtering. Secondly,

a performance analysis of this new filter is illustrated on simulated and real data. To obtain an objective judgement the PHD filter is also used for the same scenarios. This article is structured as follows: Firstly, some basic theory about PPP's is described to make the article self-contained. Secondly, the iFilter and its SMC implementation is derived. Followed by numerical studies on simulated and real data for linear and non linear scenarios. We close with a discussion about the results. In the appendix the relationship between the iFilter and the PHD filter is presented.

## II. POISSON POINT PROCESSES (PPP'S)

This section gives a short introduction to basics of PPP's, which are used in the following. For further background see [7]. Every PPP defined on a general set  $\mathcal{S}$  is parametrized by a non-negative function  $g$ , called the intensity, with  $\int_{\mathcal{S}} g(s) ds < \infty$ . In multi-target tracking applications  $g$  is not constant, so that the PPP is called non-homogenous. One realization of the PPP with intensity  $g(s)$  comprises the number and the locations of points in  $\mathcal{S}$ . A two step sampling procedure reveals the basic structure. First, the number,  $n \geq 0$ , of points is determined by sampling the discrete Poisson variable with probability mass function given by

$$\Pr[n] = \exp\left(-\int_{\mathcal{S}} g(s) ds\right) \frac{\left(\int_{\mathcal{S}} g(s) ds\right)^n}{n!}, n = 0, 1, 2, \dots \quad (1)$$

Take into account that

$$\mathbb{E}[n] = \int_{\mathcal{S}} g(s) ds. \quad (2)$$

The  $n$  points in  $\mathcal{S}$  are obtained as independent and identically distributed (i.i.d.) samples of the pdf given by  $g(s)/\int_{\mathcal{S}} g(s) ds$ . The event space of a PPP is defined as

$$\mathcal{E}(\mathcal{S}) := \{(0)\} \cup \{(n, \{\mathbf{x}_1, \dots, \mathbf{x}_n\}) : \mathbf{x}_i \in \mathcal{S}, i = 1, \dots, n\}_{n=1}^{\infty}. \quad (3)$$

Two PPP's on  $\mathcal{S}$  with intensities  $g$  and  $h$  are linearly superposed if independent realizations of each are combined into one event. If  $(n, \{\mathbf{x}_1, \dots, \mathbf{x}_n\})$  and  $(m, \{\mathbf{y}_1, \dots, \mathbf{y}_m\})$  are two such realizations, the combined event is  $(n + m, \{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}_1, \dots, \mathbf{y}_m\})$ . This event is probabilistically equivalent to a realization of a PPP whose intensity is  $g + h$ . Linearly superposing PPP's yields another PPP whose intensity is the sum of the intensities of the superposed PPP's.

### A. PPP's for multi-target tracking

In multi-target tracking applications two sequences of PPP's are usually used: one which corresponds to the multi-target state  $\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_k$  and one that corresponds to measurements  $\mathcal{Z}_1, \mathcal{Z}_2, \dots, \mathcal{Z}_k$ . Both are bound to discrete time steps  $t_0, t_1, \dots, t_k$ , with  $t_{j-1} < t_j$  for  $j = 1, \dots, k$ . Measurements are assumed to be only available for time steps  $j > 0$ . An important but subtle point is hidden in this language. The multi-target process is not assumed to be a PPP, but it is approximated at every time step by a PPP. These PPP approximations are the  $\mathcal{X}_k$ . Similarly, measurements sets are not assumed to be PPP's, but under the approximate PPP target models, the measurements are realizations of PPPs. These are the  $\mathcal{Z}_k$ . The multi-target state space is an augmented space  $\mathcal{S}^+ = \mathcal{S} \cup \mathcal{S}_\phi$ , where  $\mathcal{S}_\phi$  is a "clutter-target" space and the target state space  $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ , with  $n_x \geq 1$  the dimension of the individual target state, is bounded. The augmented state space enables estimates of both target birth and measurement clutter. The measurement sequence is defined on the measurement space  $\mathcal{Z} \subset \mathbb{R}^{n_z}$ , with  $n_z \geq 1$  the dimension of the individual measurement.

### III. THE SMC-iFILTER

The iFilter can be summarized in six steps, which will be presented in the following using a sequential Monte Carlo representation. Here the particle set represents the target intensity of the PPP, which corresponds to the multi-target state  $\mathcal{X}_k$ . Analog to the PHD-filter the integral over this intensity (or sum, if using particles) is the estimated number of targets and is not necessary equal to one. Given from the previous time step we have the particle set:

$$\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k}, \quad (4)$$

with  $\mathbf{x}_i \in \mathbb{R}^{n_x}$  and  $w_i$  the corresponding weight.  $N_k$  denotes the number of particles estimated at time step  $t_{k-1}$ . This set represents the target intensity. In addition we have the intensity of the clutter space  $\phi$  denoted by  $f_{k-1|k-1}(\phi)$ . In practical implementations this intensity can be represented by a single number, called the number of  $\phi$  hypotheses. In order to model the filter fully the following probabilities have to be defined:

$$\psi_k(\mathbf{x} | \phi) \quad \text{transition probability for new targets} \quad (5)$$

$$\psi_k(\mathbf{x} | \mathbf{y}) \quad \text{target transition probability} \quad (6)$$

$$\psi_k(\phi | \phi) \quad \text{probability for no target presence} \quad (7)$$

$$\psi_k(\phi | \mathbf{x}) \quad \text{transition probability for target death} \quad (8)$$

$$p_k(\mathbf{z} | \mathbf{x}) \quad \text{measurement likelihood} \quad (9)$$

$$p_k(\mathbf{z} | \phi) \quad \text{probability for measurement from clutter} \quad (10)$$

$$p_k^D(\mathbf{x}) \quad \text{detection probability for } \mathbf{x} \quad (11)$$

$$p_k^D(\phi) \quad \text{detection probability for } \phi \quad (12)$$

with  $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ . At time  $t_k$  we get measurements

$$\mathbf{z}_{1:m} = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}, \quad (13)$$

with  $\mathbf{z}_j \in \mathbb{R}^{n_z}$  and  $j = 1, \dots, m$ . The time step  $t_0$  contains no measurements, so it is reserved for initialization. It may happen that the measurement set is empty for a given time step, in that case the update steps can be omitted. The implementation using a particle representation is summarized as follows:

#### A. Predict target intensity

The resampled particle set gained from the previous step is denoted by  $\{\mathbf{x}_i, w_i\}_{i=1}^{N_k}$ , where  $N_k$  was estimated in time step  $t_{k-1}$ , c.f. III-F. These particles represent the intensity over  $\mathcal{S}$ . Another interpretation is, that every particle represents a possible target state in  $\mathcal{S}$ , 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  $\mathcal{S}$  with a closed formula.

The iFilter models the birth process by itself, so that the particle number has to be increased in order to represent newly born targets correctly.

$$N_{k,new} = N_k \cdot \psi_k(\mathbf{x} | \phi) \cdot f_{k-1|k-1}(\phi) \quad (14)$$

denotes the additional number of particles. Newly created particles are sampled uniformly over  $\mathcal{S}$  with weights

$$w_i = \frac{\psi_k(\mathbf{x} | \phi) \cdot f_{k-1|k-1}(\phi)}{N_k + N_{k,new}}, \quad i = 1, \dots, N_{k,new} \quad (15)$$

This sampling is an approximation of the transition model  $\psi_k(\mathbf{x} | \phi)$ , which has proven very stable in experiments. We define  $\{\tilde{\mathbf{x}}_i, w_i\}_{i=1}^{N_k + N_{k,new}}$  as the predicted particle set containing the newly created and shifted particles.

#### B. Predict hypothesis intensity

In order to predict the number of  $\phi$  hypotheses, compute the number of persistently absent and newly absent targets

$$\hat{b}_k(\phi) = \psi_k(\phi | \phi) \cdot f_{k-1|k-1}(\phi) \quad (16)$$

$$\hat{d}_k(\phi) = \sum_{i=1}^{N_k + N_{k,new}} \psi_k(\phi | \tilde{\mathbf{x}}_i) \cdot w_i \quad (17)$$

The predicted number is then:

$$f_{k|k-1}(\phi) = \hat{b}_k(\phi) + \hat{d}_k(\phi) \quad (18)$$

#### C. Predict measurement intensity

For all measurements  $\mathbf{z}_j$ , with  $j = 1, \dots, m$  compute the partition functions evaluated at  $\mathbf{z}_j$  for the state space and  $\phi$ :

$$\nu_k(\mathbf{z}_j) = \sum_{i=1}^{N_k + N_{k,new}} p_k(\mathbf{z}_j | \tilde{\mathbf{x}}_i) p_k^D(\tilde{\mathbf{x}}_i) w_i \quad (19)$$

$$\tilde{\lambda}_k(\mathbf{z}_j) = p_k(\mathbf{z}_j | \phi) p_k^D(\phi) f_{k|k-1}(\phi). \quad (20)$$

The sum of both is the measurement intensity for  $\mathbf{z}_j$

$$\lambda_{k|k-1}(\mathbf{z}_j) = \tilde{\lambda}_k(\mathbf{z}_j) + \nu_k(\mathbf{z}_j) \quad (21)$$

#### D. Update target intensity

Given  $m$  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  $i = 1, \dots, N_k + N_{k, \text{new}}$  set:

$$\hat{w}_i = \left[ (1 - p_k^D(\tilde{\mathbf{x}}_i)) + \sum_{j=1}^m \frac{p_k(\mathbf{z}_j | \tilde{\mathbf{x}}_i) p_k^D(\tilde{\mathbf{x}}_i)}{\lambda_{k|k-1}(\mathbf{z}_j)} \right] \cdot w_i \quad (22)$$

#### E. Update hypothesis intensity

Since we use the augmented space  $\mathcal{S}^+$  we have to adjust also the number of  $\phi$  hypotheses:

$$f_{k|k}(\phi) = \left[ (1 - p_k^D(\phi)) + \sum_{j=1}^m \frac{p_k(\mathbf{z}_j | \phi) p_k^D(\phi)}{\lambda_{k|k-1}(\mathbf{z}_j)} \right] \cdot f_{k|k-1}(\phi) \quad (23)$$

#### F. Resampling

The number of particles in the state space may and should vary over time in order to represent the current situation better, e.g. more targets need more particles, so that the particle approximation accuracy is still sufficient. To estimate the correct number of particles resampled for the next time step compute first the number of targets

$$\eta_k = \sum_{i=1}^{N_k + N_{k, \text{new}}} w_i. \quad (24)$$

Then compute the following probability:

$$p_S = \frac{\eta_k}{\eta_k + f_{k|k}(\phi)}. \quad (25)$$

The number of resampled particles  $N_{k+1}$  is then the expectation of a binomial distribution with the probability  $p_S$  and runs equal to  $N_k + N_{k, \text{new}}$ , i.e.

$$N_{k+1} = (N_k + N_{k, \text{new}}) \cdot p_S. \quad (26)$$

The estimation of  $N_{k+1}$  in every time step prevails the particle number from growing against infinity. Given  $N_{k+1}$  a standard resampling technique for particle filtering can be used, e.g. the following:

Initialize the cumulative probability with  $c_1 = 0$  and set

$$c_i = c_{i-1} + \frac{\hat{w}_i}{\eta_k}, \text{ for } i = 2, \dots, N_k + N_{k, \text{new}}. \quad (27)$$

Draw a uniformly distributed starting point  $a_1$  from the interval  $[0, N_{k+1}^{-1}]$ .

For  $j = 1, \dots, N_{k+1}$ ,

$$a_j = a_1 + N_{k+1}^{-1} \cdot (j - 1) \quad (28)$$

while  $a_j > c_i$ ,

$$i = i + 1.$$

end while. (29)

$$\mathbf{x}_j = \tilde{\mathbf{x}}_i \quad (30)$$

$$w_j = N_{k+1}^{-1} \quad (31)$$

Rescale the weights by  $\eta_k$  to get a new particle set  $\{\mathbf{x}_j, \eta_k/N_{k+1}\}_{j=1}^{N_{k+1}}$ .

After every time step  $k$  we generate a particle cloud, which represents the PPP over  $\mathcal{S}$ . To estimate the correct object states from this cloud we have to perform a clustering. The iFilter filter estimates the number of objects for every time step, so it is possible to use a clustering technique, which requires the number of clusters, e.g.  $k$ -means clustering [12]. However the estimated object number has a high variance. This behavior was already shown for the PHD filter [13]. Since the iFilter is a generalization it still suffers from this problem. To deal with it, we use in our experiments the adaptive resonance theory (ART) clustering [14], which estimates the number of clusters automatically, with a distance parameter as predefined user input. ART is used to estimate the target count and the individual target states from the particle cloud. In fact we only use a subset

$$S \subset \{\mathbf{x}_j, w_j\}_{j=1}^{N_{k+1}}, \quad (32)$$

with

$$(\mathbf{x}_j, w_j) \in S \text{ if } w_j \geq \tau. \quad (33)$$

## IV. NUMERICAL STUDIES

In order to analyze the performance of the iFilter we test it against its specialization the PHD filter using the OSPA-metric [15]. Some words on the strong relationship between both filters can be found in the appendix. In all experiments we use a sequential Monte Carlo version of both filters. A general description of the SMC-PHD implementation can be found in [10]. The implementation used here was published in [16]. We used the PHD as it was described in those papers for a matched clutter rate. In the following we present results from simulated and real world scenarios.

#### A. Simulated data

In this subsection we present tracking results for both filters based on the OSPA-metric. First, we demonstrate results using a linear scenario with two crossing inertial moved objects. We used for all experiments in this paper a PHD filter which is adopted to a low clutter scenario. Then we have used both filters to track the objects in a low, middle and high clutter scenario, c.f Figure 1. The number of clutter measurements was estimated following a poisson distribution with mean value  $A \cdot \rho_A$

$$p(n_c) = \frac{1}{n_c!} (A \cdot \rho_A)^{n_c} * \exp(-A \cdot \rho_A), \quad (34)$$

with  $A$  denoting the volume of a observed area and  $\rho_A$  a parameter describing the clutter rate. For the low clutter scenario we used  $\rho_A = 4 \cdot 10^{-6}$ ,  $\rho_A = 9 \cdot 10^{-6}$  for middle and  $\rho_A = 9 \cdot 10^{-5}$  for high clutter rates.  $n_c$  many clutter measurements are generated by a i.i.d. process.

Since the PHD filter and iFilter are close related the tracking results should be nearly the same if the clutter model is chosen correctly for the PHD filter. The whole scenario has a length of 100 time steps. The PHD filter here was realized

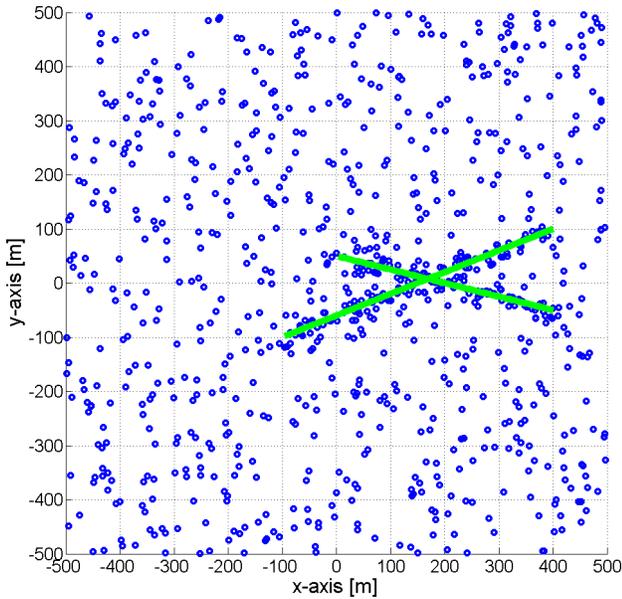


Figure 1. Linear scenario used for performance evaluation. Two targets move inertially from left to right (green lines). The blue circles denote all measurements for 100 time steps, including clutter.

with 5000 particles and its clutter rate was matched to a low clutter scenario. The iFilter started with 5000 and reached a mean particle number of about 6500. Figure 2 shows the OSPA values for 500 Monte Carlo runs with a low clutter model. Although, the results of the PHD filter are good for this scenario, the iFilter reaches better values. The estimated clutter model of the iFilter handles the data better than the a priori clutter model for the PHD filter, although we matched the a priori clutter model of the PHD filter to this scenario. The modeling of the clutter process is hard work and it can be observed that slight errors can reduce the performance. In practice the perfect modeling of a clutter rate is not possible, so that the automatic estimation of the clutter rate is desirable. In addition, it is obvious that the PHD filter converges faster in the beginning. Keeping in mind that iFilter has to estimate the clutter and birth process before it can produce reliable results, this behavior is explainable and can be seen as "normal".

For the following experiments with middle and high clutter rates we did not change the parameters of both filters, e.g. the clutter model for the PHD filter is still matched the low clutter scenario. With this experiments we want to investigate how strong the impact of a varying clutter model is for both filters if the user defined parameters are to adapted to the scenario. Figures 3 and 4 show that the PHD filter generates now worse results, but also the iFilter cannot keep its performance compared to the low clutter scenario. Therefore, it seems reasonable to conclude that the remaining free parameters of the iFilter have also a influence on the correctness of the clutter rate estimation. Even though the results of the iFilter are worse here, they are in the same range as the PHD filter was in the low clutter scenario.

In the next step we tested both filters on a non-linear

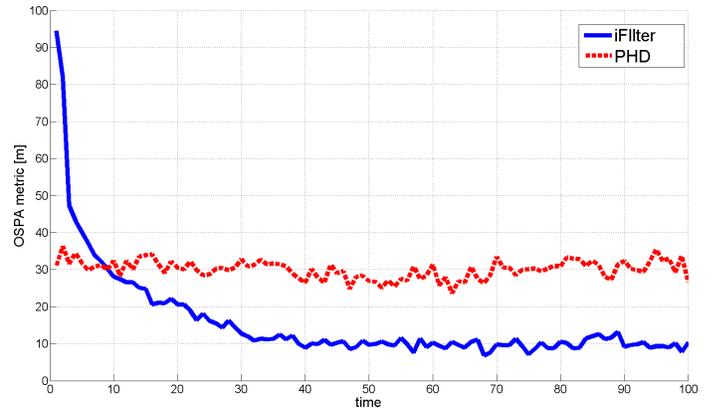


Figure 2. OSPA-metric for 500 Monte Carlo runs on a linear scenario with low clutter rates.

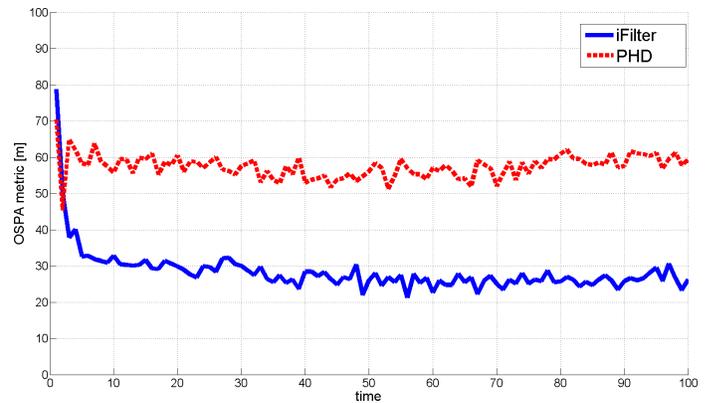


Figure 3. OSPA-metric for 500 Monte Carlo runs on a linear scenario with middle clutter rates.

scenario, c.f Figure 5. Here we use bearing measurements (azimuth and elevation) to estimate position and velocity of multiple targets. The measurement likelihood is defined

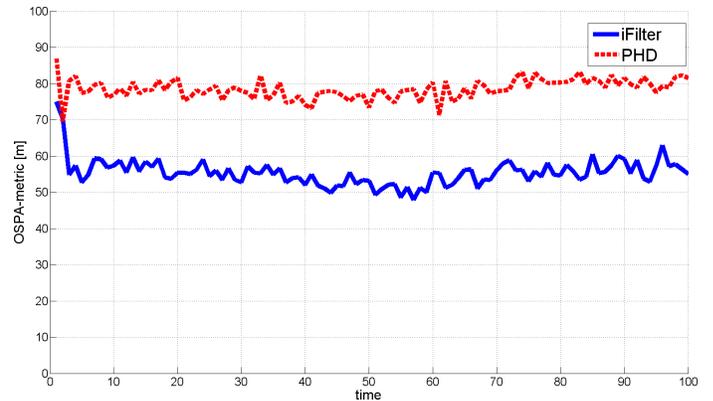


Figure 4. OSPA-metric for 500 Monte Carlo runs on a linear scenario with high clutter rates.

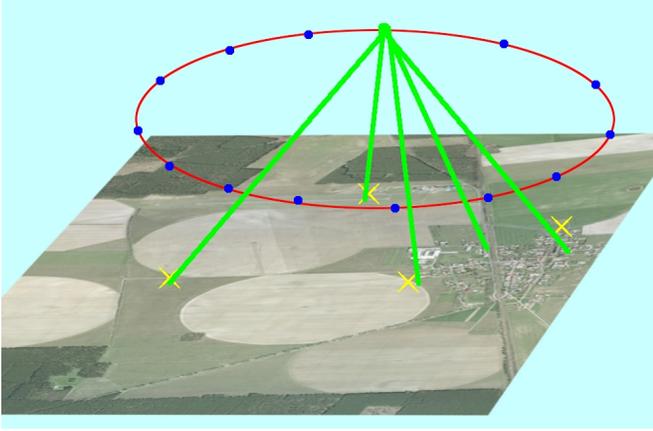


Figure 5. Non linear scenario used in our simulations and real world experiments. A possible observer path is illustrated by the red ellipse, whereby the blue dots represent the discrete measurement points. The yellow crosses are objects of interest in this scene, which should be localized and tracked from the algorithm by bearing data. The latter is represented by green rays which point in the direction resulting from processing the sensor output.

through:

$$p(\mathbf{z}|\mathbf{x}) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{z} - f(\mathbf{x}))^T \Sigma^{-1}(\mathbf{z} - f(\mathbf{x}))\right), \quad (35)$$

with  $\Sigma$  the covariance matrix of the measurement noise and

$$f(\mathbf{x}) = \begin{pmatrix} \arctan\left(\frac{\mathbf{x}(1) - \mathbf{x}_{\text{obs}}(1)}{\mathbf{x}(2) - \mathbf{x}_{\text{obs}}(2)}\right) \\ \frac{\pi}{2} + \arctan\left(\frac{\mathbf{x}(3) - \mathbf{x}_{\text{obs}}(3)}{\sqrt{(\mathbf{x}(1) - \mathbf{x}_{\text{obs}}(1))^2 + (\mathbf{x}(2) - \mathbf{x}_{\text{obs}}(2))^2}}\right) \end{pmatrix}. \quad (36)$$

An observer performs an half circle flight over a region of interest. For discrete time steps we gain bearing measurements from three targets and additionally some clutter measurements. Details on this scenario can be found in [16]. The covariance matrix  $\Sigma$  was chosen according to sensor models for small antenna arrays, i.e. high angular error. Again, we performed a Monte Carlo simulation with 500 runs, c.f. Figure 6. It can be easily seen that again the iFilter produces lower OSPA values compared to the PHD filter.

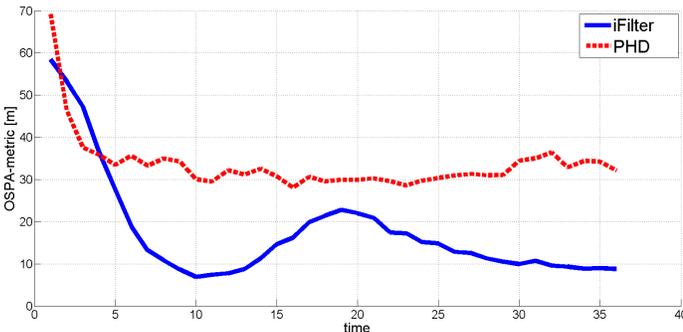


Figure 6. OSPA-metric for 500 Monte Carlo runs on a non-linear scenario.



Figure 7. Typical input images produced by our camera system. Top row: scaled image used for processing. Bottom row: cut in original size on a car and two airplanes.

## B. Real data

In this subsection we present localization and tracking results achieved with real data. As sensor platform we used an unmanned aerial system (UAS). The UAS was equipped with a Global Positioning System (GPS) and an Inertial Navigation System (INS), so that at every time step the position and attitude information of the observer is available. The UAS was flying at a height of about 1000 meters above ground level. As measurement we used here again bearings (azimuth and elevation), like in the simulated non-linear scenario. As sensors for bearing measurements we used:

### 1) Antenna Array

A three-element antenna array was mounted beneath a UAS. This small array is able to detect and compute bearing data for satellite telephone uplink communication. In order to obtain data from the received signal we used the strategy proposed in [19]. The challenge for a filter lies in a non-gaussian error distribution and additional grating lobe effects, which leads to high errors in the estimated bearings. The errors here have a systematic and statistical component. In the filter we only modeled the systematic errors.

### 2) Optical System

In addition to the antenna array we used a fixed down-looking high resolution camera system. The field of view was 114 degree horizontal and 88 vertical. To detect possible object we use the technique presented in [17]. This detection procedure uses shape and color information to find objects in color images. For the experiments presented here we limited ourselves to airplanes on the ground and cars (c.f. Figure 7). Once a

object has been detected bearing data can be computed using the position and attitude information of the UAS. The necessary formulas can be found in [18].

The results for the optical and antenna system can be seen in Figures 9 and 8, respectively. As it can be easily seen the performance for the optical sensor is much better in comparison to the antenna system. This relies on the fact, that the bearing errors of the antenna system are very high and have additionally a strong systematic component. Nevertheless, for both sensor types the iFilter produces very good results and estimates the number of targets correctly for all time steps. In addition we can observe an increase of the localization confidence given more measurements. These results state that the performance we achieved with simulated data (c.f. Figure 6) also holds for real data. In [16] comparable results with a PHD filter for this data was shown.

## V. CONCLUSIONS

This paper presents the first sequential Monte Carlo implementation of the iFilter. In addition an analysis on simulated and real data of this new filter is presented in comparison to the well-known PHD filter. In simulations, the general behavior of the iFilter was investigated. It was demonstrated in linear and non-linear scenarios that iFilter has in general a better performance than the PHD filter, especially if the clutter model for the PHD filter is not known perfectly. Even in situations, where the clutter model of the PHD was matched to the clutter rates in the scenario, the iFilter outperformed the PHD. The implementation and usage of the PHD filter was done according to the references in literature, which are published up to now. The only drawback of the iFilter is its slight slower convergence in comparison to the PHD filter. The good performance of the iFilter was also confirmed in a demanding estimation problem for real bearing data with high systematic and statistical errors.

In future work we want to investigate the performance of the iFilter in scenarios with multiple heterogeneous measurement sources in comparison to other established filter frameworks. The development of an cardinalized iFilter is desirable, since the high variance of the estimated target number from the PHD filter can be also observed for the iFilter.

## 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, *IEEE J. Oceanic 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. Streit and L. Stone, “Bayes derivation of multitarget intensity filters,” in *11th International Conference on Information Fusion*, 2008.
- [5] R. Streit, “Multisensor multitarget intensity filter,” in *11th International Conference on Information Fusion*, 2008.
- [6] R. Mahler, “Multitarget bayes filtering via first-order multitargets moments,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [7] R. Streit, *Poisson Point Processes*. Springer, 2010.
- [8] H. Sidenbladh, “Multi-target particle filtering for probability hypothesis density,” in *International Conference on Information Fusion*, Cairns, Australia, 2003, pp. 800–806.

- [9] 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.
- [10] 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.
- [11] B.-T. Vo and W.-K. Ma, “The gaussian mixture probability hypothesis density filter,” *IEEE Trans. Signal Processing*, vol. 55, no. 11, pp. 4091–4104, 2006.
- [12] S. Lloyd, “Least squares quantization in pcm,” *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [13] R. Mahler, “PHD filters of higher order in target number,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, 2007.
- [14] G. Carpenter and S. Grossberg, “ART 2: Self-organizing stable category recognition codes for analog input patterns,” *Applied Optics*, vol. 26, no. 23, pp. 4919–4930, 1987.
- [15] D. Schumacher, B. Vo, and B. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.
- [16] M. Schikora, D. Bender, D. Cremers, and W. Koch, “Passive multi-object localization and tracking using bearing data,” in *13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.
- [17] M. Schikora, “Global optimal multiple object detection using the fusion of shape and color information,” in *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, August 2009.
- [18] M. Schikora, D. Bender, W. Koch, and D. Cremers, “Multitarget multisensor localization and tracking using passive antennas and optical sensors,” *Proc. SPIE, Security + Defense*, vol. 7833, 2010.
- [19] R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” *Proc. RADC Spectrum Estimation Workshop, Griffith AFB*, pp. 243–258, 1979.

## APPENDIX

### *Relationship to the PHD filter*

The proposed derivation of the iFilter is very general and specializations for different applications and use cases are possible. The most known is the PHD filter, which was originally derived using the random finite set theory. Nevertheless, it can also be derived using PPPs, analog to the iFilter. Details on this topic can be found in [7].

The main differences are reducible to the augmented state space  $\mathcal{S}^+$ . While the iFilter uses  $\mathcal{S}^+ = \mathcal{S} \cup \mathcal{S}_\phi$ , the PHD filter only uses  $\mathcal{S}$ . The basis for the on-line estimation of the intensities of the target birth and measurement clutter PPPs is the state  $\phi$ . If, however, the birth and clutter rates are known a priori then the state  $\phi$  can be omitted, giving the PHD filter. This requires some care. In order to discard targets, so that the target count does not balloon out of control, the PHD filter uses a death probability before predicting the multi-target intensity  $f_{k|k}(\mathbf{x})$ . The iFilter models this through  $\psi_k(\phi | \mathbf{x})$ , because transition into  $\phi$  is death. A given a priori clutter rate can replace  $\tilde{\lambda}_k(\mathbf{z}_j)$  in (21). An a priori birth model has to be considered in step 1 of the algorithm, see [10], [16] for details on this step of the PHD filter.

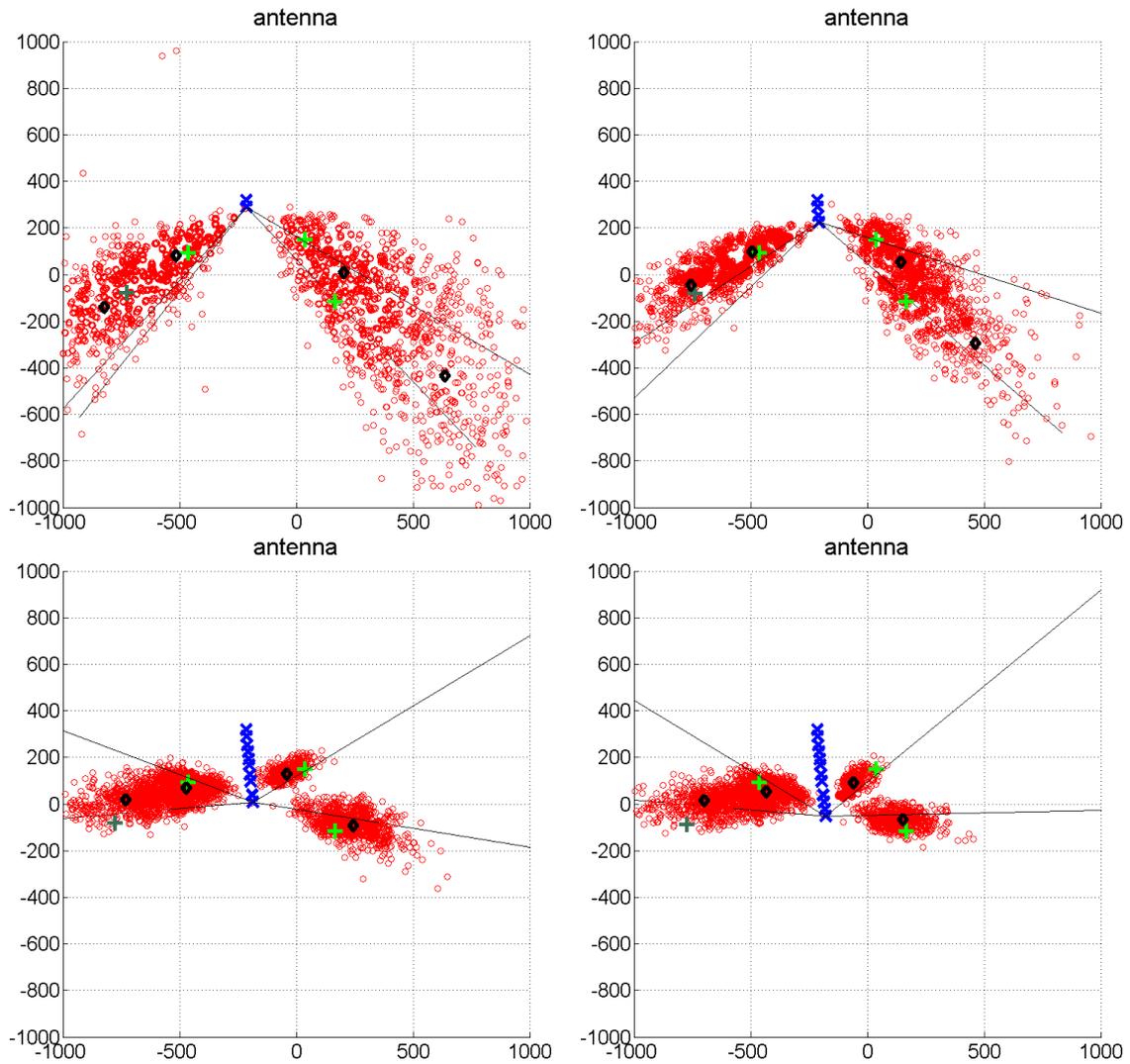


Figure 8. Particle set evolution for a antenna array system at different time steps. The blue x's represent the observer position at a bearing measurement. The corresponding bearings are represented by black rays. The individual particles are illustrated as red circles, whereby the estimated localizations are displayed as black diamonds and the ground truth is represented by green crosses. The leftmost object is moving to the left (dark green cross) and the three others are stationary targets. For a better perspicuity this Figure only shows the top view of the 3D scenario.

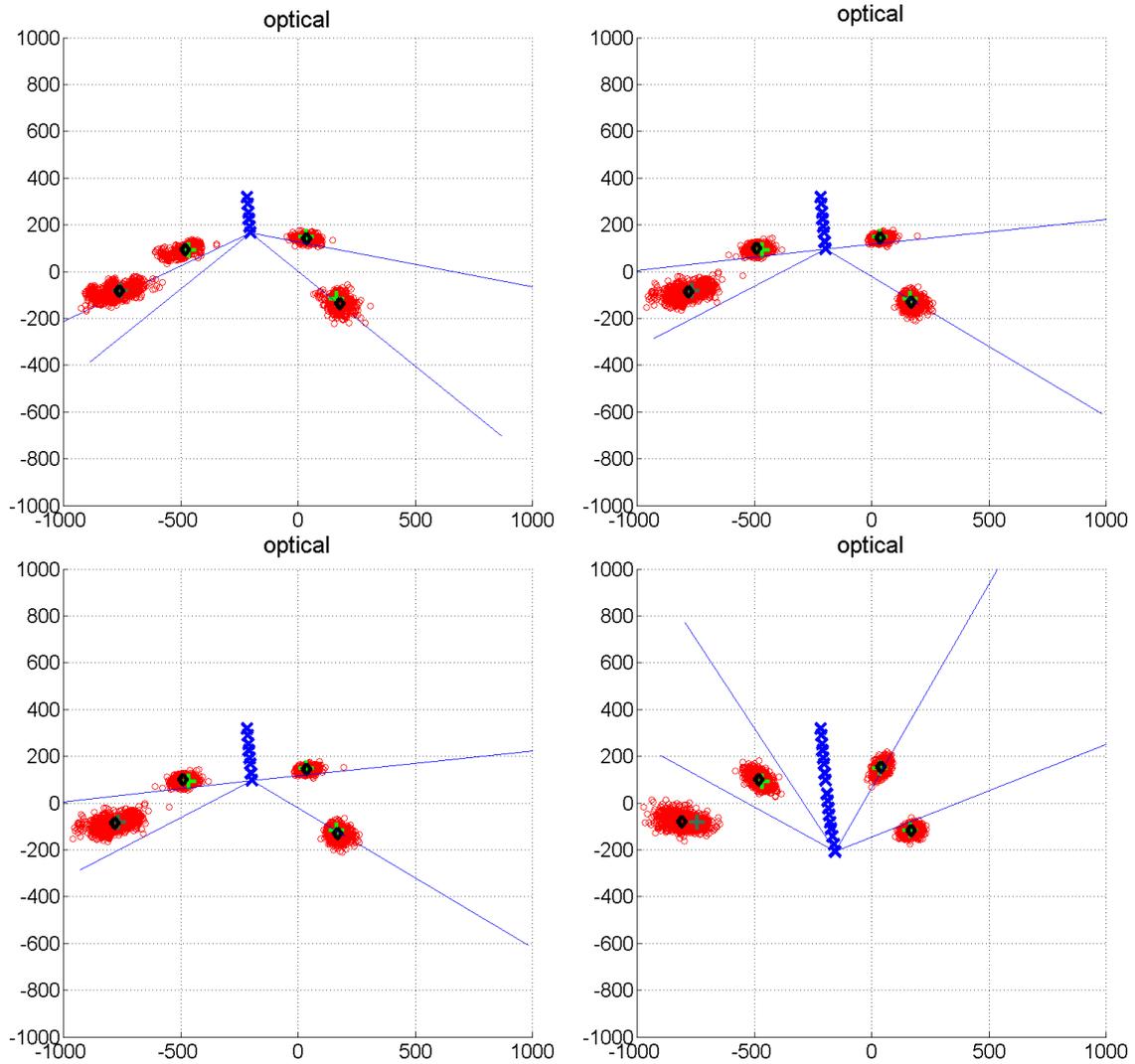


Figure 9. Particle set evolution for a optical system at different time steps. The blue x's represent the observer position at a bearing measurement. The corresponding bearings are represented by blue rays. The individual particles are illustrated as red circles, whereby the estimated localizations are displayed as black diamonds and the ground truth is represented by green crosses. The leftmost object is moving to the left (dark green cross) and the three others are stationary targets. For a better perspicuity this Figure only shows the top view of the 3D scenario.