Chapter 3
A Sequential Monte Carlo Method for Multi-target Tracking with the Intensity Filter

Marek Schikora, Wolfgang Koch, Roy Streit, and Daniel Cremers

Abstract. Multi-target tracking is a common problem with many applications. In most of these the expected number of targets is not known a priori, so that it has to be estimated from the measured data. Poisson point processes (PPPs) are a very useful theoretical model for diverse applications. One of those is multi-target tracking of an unknown number of targets, leading to the intensity filter (iFilter) and the probability hypothesis density (PHD) filter. This chapter presents 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.

3.1 Introduction

This chapter presents a novel sequential Monte Carlo approach for multi-target tracking, called the intensity filter. In general, multi-target tracking involves the joint estimation of states and number of targets from a sequence of observations in the presence of detection uncertainty, association uncertainty and clutter [2]. Classical approaches such as the Joint Probabilistic Data Association filter (JPDAF) [9] and

Marek Schikora · Wolfgang Koch
Department of Sensor Data and Information Fusion,
Fraunhofer FKIE, Wachtberg, Germany
e-mail: {marek.schikora,wolfgang.koch}@fkie.fraunhofer.de,

Roy Streit
Metron Inc., Reston VA, USA
e-mail: r.streit@ieee.org

Daniel Cremers
Department of Computer Science, TU Munich, Garching Germany
e-mail: cremers@tum.de

P. Georgieva et al. (Eds.): Adv. in Intelligent Signal Processing and Data Mining, SCI 410, pp. 55–87.
springerlink.com

multi hypothesis tracking (MHT) [16] need in general the knowledge of the expected number of targets. Recently, the intensity filter (iFilter) [31, 28, 29] has been presented, which is similar to the well-known probability density hypothesis (PHD) filter [12]. Both filters give estimates to multi-target and multi-measurement states along with the number of targets. While the PHD filter was originally derived using finite set statistics, the iFilter was derived through Poisson point processes (PPPs). Furthermore it was shown that the PHD filter is a special case of the iFilter [29]. From an engineering point of view the main difference between the PHD and iFilter is the clutter rate model, which has to be known for the PHD filter a priori and is estimated by the iFilter. Many implementations of the PHD filter have been proposed either using sequential Monte Carlo methods [27, 37, 32], or with Gaussian mixtures [33]. An implementation and analysis of the iFilter was published in [24].

The main contribution of [24] is the first presentation of an implementation scheme for the iFilter using a sequential Monte Carlo method, also called particle filtering. The authors present a performance analysis of this new filter on simulated and real data. To obtain an objective judgement the PHD filter was also used for the same scenarios. This chapter however intends to improve the implementation proposed in [24] by adding a measurement steered birth model and a novel state extraction schema without the need of clustering techniques. This improvement is mainly based on the ideas presented by Ristic et al. [18]. The state extraction schema especially is well designed for scenarios with a high probability of detection. In this paper we will limit ourself to such experiments.

Moreover, we analyze the iFilter in demanding situations with various clutter rates and show the correct behavior of it. One example is the case, where no clutter measurements are present. The iFilter models its birth model through a clutter space $\mathscr{S}_\phi$, which is mainly responsible for the clutter rate estimation and will be explained in this chapter. If there is no clutter, the expected number of elements in this space will tend to zero and so one might think that target birth is no longer possible. However, this guess is not correct. In various experiments we show that the filter increases the intensity on $\mathscr{S}_\phi$, when a new target appears, and then can model target birth correctly, even in the case of no clutter measurements.

The remaining part of this book chapter is organized in the following way: Section 3.2 contains an introduction to Poisson Point Processes with a strong focus on multi-target tracking. The following Section 3.3 describes the intensity filter and how it can be implemented by a sequential Monte Carlo method. An intensive study of performance for simulated data is given in Section 3.4. In addition this section provides a comparison to the PHD filter and reveals the connection between both filters. Section 3.5 illustrates two applications in which the iFilter is used for the purpose of multi-target tracking. The first application is tracking of multiple objects with bearings as measurements, often called bearings-only tracking (BOT). The second presents a combination of high-accuracy optical flow and multi-target tracking algorithms for video tracking. Conclusions are drawn in the final Section 3.6.

3.2 Poisson Point Processes (PPPs)

This section gives a short introduction to basics of Poisson Point Processes (PPPs), which are used in the following. See [29] for further background. They are named after Siméon Denis Poisson (1781-1840).

Definition 1. ***(Poisson distribution)*** *A random variable $x : \Omega \to \mathbb{N}_0$ is said to be Pois(λ) distributed , with $\lambda > 0$, if*

$$p(x = n) = \exp(-\lambda)\frac{\lambda^n}{n!}, \tag{3.1}$$

for all $n \in \mathbb{N}_0$. $p(x = n)$ denotes the probability for n occurrences of x.

Every PPP is defined on a general set $\mathscr{S}$. In most of our applications this space will be considered as Euclidean, $\mathscr{S} \subseteq \mathbb{R}^d$, with $d \geq 1$ denoting the dimension, e.g. $d = 2$ for targets located in the (x,y)-plane. Other more complicated spaces are also possible. Realizations of PPPs on $\mathscr{R} \subset \mathscr{S}$ consist of $n > 0$ points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in \mathscr{R}$. We denote a realization as ordered pair

$$\xi = (n, \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}). \tag{3.2}$$

If n is equal to zero we set $\xi = (0, \emptyset)$, with $\emptyset$ the empty set. Through this notation we emphasize that the ordering of $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ is irrelevant, but not that the points are necessarily distinct.

Definition 2. ***(Event Space)*** *The event space of a PPP is defined as*

$$\mathscr{E}(\mathscr{S}) := \{(0, \emptyset)\} \cup \{(n, \{\mathbf{x}_1, ..., \mathbf{x}_n\}) : \mathbf{x}_i \in \mathscr{S}, i = 1, ..., n\}_{n=1}^{\infty}. \tag{3.3}$$

Definition 3. ***(Intensity)*** *Every PPP is parameterized by only one function $g : \mathscr{S} \to \mathbb{R}, s \mapsto g(s)$, $s \in \mathscr{S}$, called the intensity. We call $g(s)$ the intensity at point s. For all $s \in \mathscr{S}$, if $g(s) = c, c \geq 0$, where c is constant, the PPP is called homogeneous; otherwise it is non-homogeneous.*

It is assumed that

$$0 \leq \int_{\mathscr{R}} g(s)\mathrm{d}s < \infty \tag{3.4}$$

holds for all bounded subsets $\mathscr{R}$ of $\mathscr{S}$, $\mathscr{R} \subset \mathscr{S}$. One realization of the PPP with intensity $g(s)$ comprises the number and the locations of points in $\mathscr{R}$.

3.2.1 PPP Sampling Procedure

A two step sampling procedure reveals the basic structure. Firstly, the number, $n \geq 0$, of points is determined by sampling the discrete Poisson variable with probability mass function given by

$$\Pr[n] = \exp\left(-\int_{\mathscr{R}} g(s)\mathrm{d}s\right) \frac{\left(\int_{\mathscr{R}} g(s)\mathrm{d}s\right)^n}{n!}, n = 0,1,2,... \tag{3.5}$$

It follows from (3.5) that

$$\mathrm{E}[n] = \int_{\mathscr{R}} g(s)\mathrm{d}s. \tag{3.6}$$

Secondly, the n points in $\mathscr{R}$ are obtained as independent and identically distributed (i.i.d.) samples of the PDF (probability density function) given by $g(s)/\int_{\mathscr{R}} g(s)\mathrm{d}s$.

Two PPPs on $\mathscr{S}$ with intensities g and h are linearly superposed, if independent realizations of each are combined into one event. If $\xi_1 = (n, \{\mathbf{x}_1, ..., \mathbf{x}_n\})$ and $\xi_2 = (m, \{\mathbf{y}_1, ..., \mathbf{y}_m\})$ are two such realizations, the combined event is $\xi_3 = (n+m, \{\mathbf{x}_1, ..., \mathbf{x}_n, \mathbf{y}_1, ..., \mathbf{y}_m\})$. This event is probabilistically equivalent to a realization of a PPP whose intensity is $g+h$. In words, superposition of linearly independent PPPs yields another PPP whose intensity is the sum of the intensities of the superposed PPPs.

3.2.2 PPPs for Multi-target Tracking

In multi-target tracking applications two sequences of PPPs are usually used: one which corresponds to the multi-target state $\mathscr{X}_0, \mathscr{X}_1, ..., \mathscr{X}_k$ and one that corresponds to measurements $\mathscr{Z}_1, \mathscr{Z}_2, ..., \mathscr{Z}_k$. Both are bound to discrete time steps $t_0, t_1, ..., t_k$, with $t_{j-1} < t_j$ for $j = 1, ...k$. 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 $\mathscr{X}_k$. Similarly, measurement sets are not assumed to be PPPs. However, under the approximate PPP target models, the measurements are realizations of PPPs. These are the $\mathscr{Z}_k$.

We define now $\mathscr{S} \subseteq \mathbb{R}^{n_x}$, with $n_x \geq 1$, an n_x dimensional bounded single target state space. The multi-target state space is then an augmented space $\mathscr{S}^+ = \mathscr{S} \cup \mathscr{S}_\phi$, where $\mathscr{S}_\phi$ represents space of the "target absent" hypothesis ϕ. $\mathscr{S}^+$ is a discrete-continuous space. The main concepts of PPPs can be adapted to this space , but some modifications are needed. Here, $g(s)$ is a intensity defined for all $s \in \mathscr{S}^+$. Integrals of $g(s)$ over bounded subsets $\mathscr{R}$ of $\mathscr{S}^+$ must be finite, giving a discrete–continuous integral:

$$\begin{aligned} 0 &\leq \int_{\mathscr{R}^+} g(s)\mathrm{d}s \\ &\equiv g(\phi) + \int_{\mathscr{R}} g(\mathbf{x})\mathrm{d}\mathbf{x} < \infty, \end{aligned} \tag{3.7}$$

with $\mathscr{R}^+ \subset \mathscr{S}^+$, $\mathscr{R} \subset \mathscr{S}$, $\phi \in \mathscr{S}_\phi$ and $g(\phi)$ being a dimensionless intensity on $\mathscr{S}_\phi$. The number of copies of ϕ, or "clutter targets", in a realization is Poisson distributed with mean $g(\phi)$. In other words, $g(\phi)$ is the expected number of targets in ϕ. The augmented state space enables estimates of both target birth and measurement clutter. The integral $\int_{\mathscr{R}} g(\mathbf{x})\mathrm{d}\mathbf{x}$ is the expected number of targets in $\mathscr{S}$. The measurement sequence is defined on the measurement space $\mathscr{Z} \subset \mathbb{R}^{n_z}$, with $n_z \geq 1$ being the dimension of the individual measurement.

Further discussion of PPPs defined on discrete-continuous and other spaces is given in the book by Streit [29, Section 2.12.].

3.3 The Intensity Filter

3.3.1 General Overview

The iFilter operates on the augmented space $\mathscr{S}^+$. In the same way as a standard single target filter consists of two main steps (Prediction and Update) also the iFilter predicts the intensity over $\mathscr{S}^+$ and then updates this intensity every time when new measurements arrive. Hidden here lies the main difference between a single target filter (e.g. a Kalman filter) and the iFilter. While a Bayesian filter can predict and update the posterior PDF, the iFilter only predicts and updates a first moment of this PDF. This first moment is called the intensity or probability hypothesis density (PHD). The only way to handle the posterior PDF in a multi-target case statistically correct would be by applying the multi-target Bayes filter, which is in practice not feasible.

In the following sections of this chapter an index $a|b$ in the intensity function $f_{a|b}(.)$ denotes that the intensity was updated in time step t_a with all the measurements up to time step t_b.

The intensity of the PPP $\mathscr{X}_k$ is $f_{k|k}(s), s \in \mathscr{S}^+$. We split the intensity $f_{k|k}(s)$ over $\mathscr{S}^+$ into two intensities $f_{k|k}(\mathbf{x})$ and $f_{k|k}(\phi)$. In general we can write

$$f_{k|k}(s) = \begin{cases} f_{k|k}(\mathbf{x}), & s = \mathbf{x} \in \mathscr{S} \\ f_{k|k}(\phi), & s = \phi \in \mathscr{S}_\phi \end{cases} \tag{3.8}$$

with $f_{k|k}(\mathbf{x})$ being the intensity over $\mathscr{S}$ and $f_{k|k}(\phi)$ the intensity for $\mathscr{S}_\phi$.

In order to describe the iFilter the following probabilities and PDFs have to be defined:

$\psi_k(\mathbf{x} \mid \phi)$	transition probability for new targets	(3.9)
$\psi_k(\mathbf{x} \mid \mathbf{y})$	target transition probability	(3.10)
$\psi_k(\phi \mid \phi)$	transition probability in $\mathscr{S}_\phi$	(3.11)
$\psi_k(\phi \mid \mathbf{x})$	transition probability for target death	(3.12)
$p_k(\mathbf{z} \mid \mathbf{x})$	measurement likelihood	(3.13)
$p_k(\mathbf{z} \mid \phi)$	likelihood for measurement fromϕ	(3.14)

$$p_k^D(\mathbf{x}) \qquad \text{detection probability for } \mathbf{x} \tag{3.15}$$

$$p_k^D(\phi) \qquad \text{detection probability for } \phi \tag{3.16}$$

with $\mathbf{x}, \mathbf{y} \in \mathscr{S}$ and $\mathbf{z} \in \mathscr{Z}$. Let us assume that we have the intensities $f_{k-1|k-1}(\mathbf{x})$ and $f_{k-1|k-1}(\phi)$, from the previous time step t_{k-1}. Similarly to most stochastic filtering techniques, the iFilter admits the Markovian assumption that the current state is only dependent of the last state.

In every time step t_k the likelihood (3.13) is set, according to:

$$p_k(\mathbf{z} \mid \phi) = \frac{(f_{k-1|k-1}(\phi))^{m_k}}{m_k!} e^{(-f_{k-1|k-1}(\phi))}, \tag{3.17}$$

with m_k the number of measurements in time step t_k.

In the prediction phase of the algorithm we have to predict the intensity on $\mathscr{S}^+$, denoted by $f_{k|k-1}(s), s \in \mathscr{S}^+$, as a convolution:

$$f_{k|k-1}(s) = \int_{\mathscr{S}^+} \psi_k(s|y) f_{k-1|k-1}(y) \mathrm{d}y. \tag{3.18}$$

Using the definition (3.7) of a discrete–continuous integral gives the predicted intensity in the form

$$f_{k|k-1}(\mathbf{x}) = \psi_k(\mathbf{x} \mid \phi) f_{k-1|k-1}(\phi) + \int_{\mathscr{S}} \psi_k(\mathbf{x} \mid \mathbf{y}) f_{k-1|k-1}(\mathbf{y}) \mathrm{d}\mathbf{y} \tag{3.19}$$

and

$$f_{k|k-1}(\phi) = \psi_k(\phi \mid \phi) f_{k-1|k-1}(\phi) + \int_{\mathscr{S}} \psi_k(\phi \mid \mathbf{y}) f_{k-1|k-1}(\mathbf{y}) \mathrm{d}\mathbf{y}, \tag{3.20}$$

for all $\mathbf{x} \in \mathscr{S}$ and $\phi \in \mathscr{S}_\phi$, respectively. At time t_k we receive m_k measurements

$$\boldsymbol{Z}_k = \{\mathbf{z}_1, ..., \mathbf{z}_{m_k}\}, \tag{3.21}$$

with $\mathbf{z}_j \in \mathbb{R}^{n_z}$ and $j = 1, .., m_k$. 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 should be omitted. If $\boldsymbol{Z}_k$ is not the empty set, then the next step is to predict the measurement intensities. We do so by evaluating

$$\begin{aligned} \lambda_{k|k-1}(\mathbf{z}_j) &= \int_{\mathscr{S}^+} p_k(\mathbf{z}_j|s) p_k^D(s) f_{k|k-1}(s) \mathrm{d}s \\ &= p_k(\mathbf{z}_j|\phi) p_k^D(\phi) f_{k|k-1}(\phi) + \int_{\mathscr{S}} p_k(\mathbf{z}_j|\mathbf{x}) p_k^D(\mathbf{x}) f_{k|k-1}(\mathbf{x}) \mathrm{d}\mathbf{x} \end{aligned} \tag{3.22}$$

for every measurement $\mathbf{z}_j$. The term $\lambda_{k|k-1}(\mathbf{z}_j)$ can also be called in the language of thermodynamics as a partition function evaluated at $\mathbf{z}_j$ for the space $\mathscr{S}^+$ [4].

Both intensities can be now updated at the time step t_k, giving:

$$f_{k|k}(\mathbf{x}) = (1 - p_k^D(\mathbf{x}))f_{k|k-1}(\mathbf{x}) + \left[\sum_{j=1}^{m_k} \frac{p_k(\mathbf{z}_j|\mathbf{x})p_k^D(\mathbf{x})}{\lambda_{k|k-1}(\mathbf{z}_j)}\right] f_{k|k-1}(\mathbf{x}) \tag{3.23}$$

and

$$f_{k|k}(\phi) = (1 - p_k^D(\phi))f_{k|k-1}(\phi) + \left[\sum_{j=1}^{m_k} \frac{p_k(\mathbf{z}_j|\phi)p_k^D(\phi)}{\lambda_{k|k-1}(\mathbf{z}_j)}\right] f_{k|k-1}(\phi). \tag{3.24}$$

Keep in mind these intensities are not the posterior PDF of $p_k(\mathbf{X}|\boldsymbol{Z}_k)$ but only a first moment of the posterior point process, i.e., in general:

$$\int_{\mathscr{S}} f_{k|k}(\mathbf{x})\mathrm{d}\mathbf{x} \neq 1. \tag{3.25}$$

In fact, it can be shown (cf. (3.6)) that the above integral is the expected number of targets for t_k, denoted by:

$$\eta_k = \int_{\mathscr{S}} f_{k|k}(\mathbf{x})\mathrm{d}\mathbf{x} \tag{3.26}$$

The main drawback of the above filter equations is that in general the involved integrals cannot be solved analytically. Therefore an appropriate numerical solution is needed. In the following we show a sequential Monte Carlo (SMC) version of the iFilter in which the intensity $f_{k|k}(\mathbf{x})$ will be approximated by particles (delta peaks) drawn from this intensity. Actually the particles approximate the involved integrals and the intensities. Another name for this kind of technique is particle based filtering [17].

3.3.2 The SMC-iFilter

The works of Vo et al. [32] and Ristic et al. [18] give efficient sequential Monte Carlo methods for the PHD filter. We present here a sequential Monte Carlo method for the iFilter. The following implementation is an improved version of our previously published work [24]. The main improvements are a measurement steered particle placement for target birth, and target state and covariance matrix estimation without the need of clustering.

The improved SMC-iFilter can be summarized in eight steps, which will be presented in the following. Here the particle set represents the target intensity of the PPP, which corresponds to the multi-target state $\mathscr{X}_k$. By analogy to the PHD-filter, 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:

$$\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k}, \tag{3.27}$$

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}. This set represents the target intensity. In addition we have the intensity of the space $\mathscr{S}_\phi$ denoted by $f_{k-1|k-1}(\phi)$, c.f. (3.8). For the sake of simplicity, we will assume in the following uniformly distributed clutter. With this assumption the intensity $f_{k-1|k-1}(\phi)$ can be represented by a single number, called the number of ϕ hypotheses.

To initialize the particle cloud at time step t_0, $N_0 \in \mathbb{N}^+$ particles are distributed uniformly across the state space $\mathscr{S}$, e.g. $N_0 = 1000$. The weights are set to $w_i = 1/N_0$. $f_{0|0}(\phi)$ is set to a initial number, e.g. 2.

The implementation details using a particle representation are presented in the following. Steps 1 and 2 correspond to the prediction phase,steps 3-7 to the correction phase and step 8 to the resampling phase of a sequential Monte Carlo algorithm. A brief summary can be found in Algorithm 3.1.

1. **Predict target intensity**
   The resampled particle set gained from the previous step is denoted by $\{\mathbf{x}_i, w_i\}_{i=1}^{N_k}$, where N_k was estimated in time step t_{k-1}, c.f. Step 8. These particles represent the intensity over $\mathscr{S}$. Another interpretation is that every particle represents a possible target state (called microstates in the language of thermodynamics) in $\mathscr{S}$, so that the prediction of the whole set can be modeled by applying the Markovian transition model ψ_k to every particle. The weights are unchanged. In practical implementations this has the same effect as predicting the intensity distribution over $\mathscr{S}$ with a closed formula.

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

$$\tilde{\mathbf{x}}_i = \begin{pmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_i + \boldsymbol{\nu}, \tag{3.28}$$

   with $\Delta_t = t_k - t_{k-1}$ and $\boldsymbol{\nu} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{\Sigma})$ a realization of a normal distributed random vector. 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. Then

$$N_{k,new} = \left\lceil \frac{N_k}{\eta_{k-1}} \cdot (1 - \psi_k(\phi \mid \phi)) \cdot f_{k-1|k-1}(\phi) \right\rceil \tag{3.29}$$

   denotes the additional number of particles. The term η_{k-1} denotes the estimated expected number of targets from the previous time step t_{k-1}, c.f. Step 8. The most general sampling for new born particles can be realized through a uniform sampling procedure in $\mathscr{S}$. The new born particles must cover the whole state space. In order to avoid a high number of additional particles in scenarios with

Algorithm 3.1. The SMC-iFilter

In: $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k}, f_{k-1|k-1}(\phi), \mathbf{Z}_k, \mathbf{Z}_{k-1}$
Out: $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k+1}, f_{k|k}(\phi), \{\hat{\mathbf{y}}_j, \hat{\mathbf{P}}_j\}$

1. Predict target intensity
   - For $i = 1,...,N_k$ apply (3.28) to get $\tilde{\mathbf{x}}_i$.
   - Sample $N_{k,new}$ (3.29) many new particles; measurement steered according to $\boldsymbol{Z}_{k-1}$ and (3.30)
   - Weights for new particles are w_i (3.31)
2. Predict ϕ intensity
   - $f_{k|k-1}(\phi) = \hat{b}_k(\phi) + \hat{d}_k(\phi)$ (3.34)
3. Predict measurement intensity
   - $\lambda_{k|k-1}(\mathbf{z}_j) = \tilde{\lambda}_k(\mathbf{z}_j) + \nu_k(\mathbf{z}_j)$ (3.37)
4. Compute target states
   - Compute the set $\mathscr{J}$ (3.40)
   - For all $j \in \mathscr{J}$:
     $\hat{\mathbf{y}}_j = \frac{1}{W_j} \sum_{i=1}^{N_k} w_{j,i} \tilde{\mathbf{x}}_i$ (3.41)
5. Compute covariance matrices
   - For all $j \in \mathscr{J}$:
     $\mathbf{P}_j = \frac{1}{W_j} \sum_{i=1}^{N_k} w_{j,i} (\tilde{\mathbf{x}}_i - \hat{\mathbf{y}}_j)(\tilde{\mathbf{x}}_i - \hat{\mathbf{y}}_j)^T$ (3.42).
6. Update target intensity
   - For every particle $(\tilde{\mathbf{x}}_i, w_i)$, with $i = 1,..,N_k + N_{k,new}$ set the new weight according to (3.43).
7. Update ϕ intensity
   - Set $f_{k|k}(\phi)$ according to (3.44).
8. Resample
   - Compute $N_{k+1} = \left(N_k + N_{k,new}\right) \cdot p_S$ (3.47).
   - Use some standard resampling strategy to get $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_k+1}$

a high probability of detection, the authors in [18] proposed to 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

$$N_{k,new}^{j} = \lceil N_{k,new}/m_{k-1} \rceil, \quad j = 1,...,m_{k-1} \tag{3.30}$$

many particles $\tilde{\mathbf{x}}_i$ drawn from a distribution proportional to the distribution $p_k(\mathbf{z}_j^{k-1}|\mathbf{x})$ centered around an old measurement $\mathbf{z}_j^{k-1}$. The operator $\lceil . \rceil$ rounds to the next bigger integer.

The weights of the new born particles are set to

$$w_i = \frac{\psi_k(\mathbf{x}_i \mid \phi) \cdot f_{k-1|k-1}(\phi)}{N_{k,new}}, \quad i = 1,...,N_{k,new}. \tag{3.31}$$

This sampling is an approximation of the transition model $\psi_k(\mathbf{x} \mid \phi)$, 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 the shifted particles.

2. **Predict hypothesis intensity**
In order to predict the number of ϕ hypotheses, compute the predicted number of persistently absent $\tilde{b}_k$ and newly absent targets $\tilde{d}_k$ as

$$\tilde{b}_k(\phi) = \psi_k(\phi \mid \phi) \cdot f_{k-1|k-1}(\phi), \tag{3.32}$$

$$\tilde{d}_k(\phi) = \sum_{i=1}^{N_k} \psi_k(\phi \mid \tilde{\mathbf{x}}_i) \cdot w_i. \tag{3.33}$$

The predicted number is then:

$$f_{k|k-1}(\phi) = \tilde{b}_k(\phi) + \tilde{d}_k(\phi). \tag{3.34}$$

3. **Predict measurement intensity**
For all new measurements $\mathbf{z}_j$, with $j = 1,...,m_k$ compute, according to (3.22), the partition functions evaluated at $\mathbf{z}_j$ for the state space and ϕ:

$$\nu_k(\mathbf{z}_j) = \sum_{i=1}^{N_k+N_{k,new}} p_k(\mathbf{z}_j \mid \tilde{\mathbf{x}}_i) p_k^D(\tilde{\mathbf{x}}_i) w_i \tag{3.35}$$

$$\tilde{\lambda}_k(\mathbf{z}_j) = p_k(\mathbf{z}_j \mid \phi) p_k^D(\phi) f_{k|k-1}(\phi). \tag{3.36}$$

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

$$\lambda_{k|k-1}(\mathbf{z}_j) = \tilde{\lambda}_k(\mathbf{z}_j) + \nu_k(\mathbf{z}_j) \tag{3.37}$$

4. **Estimate target states**
To avoid a clustering step the methodology presented in [18] is used and adopted to the iFilter. 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, $\mathbf{x}_i, i = 1,...,N_k$.

$$w_{j,i} = \frac{p_k(\mathbf{z}_j|\tilde{\mathbf{x}}_i)p_k^D(\tilde{\mathbf{x}}_i)}{\lambda_{k|k-1}(\mathbf{z}_j)} \cdot w_i \tag{3.38}$$

Then compute the following sum

$$W_j = \sum_{i=1}^{N_k} w_{j,i}, \tag{3.39}$$

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.

$$\mathscr{J} = \{j|W_j > \tau, j = 1,...,m_k\} \tag{3.40}$$

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

$$\hat{\mathbf{y}}_j = \frac{1}{W_j} \sum_{i=1}^{N_k} w_{j,i}\tilde{\mathbf{x}}_i. \tag{3.41}$$

In Equation (3.41) we added, in contrast to [18], the normalization term $\frac{1}{W_j}$ to receive more accurate state estimates when W_j is not practically one. 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 iFilter still suffers from this effect. The full characteristics are discussed in [8]. In experiments τ is usually set as $\tau = 0.75$.

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

$$\mathbf{P}_j = \frac{1}{W_j} \sum_{i=1}^{N_k} w_{j,i}(\tilde{\mathbf{x}}_i - \hat{\mathbf{y}}_j)(\tilde{\mathbf{x}}_i - \hat{\mathbf{y}}_j)^T. \tag{3.42}$$

   In Equation (3.42) we added, in contrast to [18], 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 $\hat{\mathbf{y}}_j$.

6. **Update target intensity**
   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:

$$\hat{w}_i = \left[(1 - p_k^D(\tilde{\mathbf{x}}_i)) + \sum_{j=1}^{m_k} \frac{p_k(\mathbf{z}_j \mid \tilde{\mathbf{x}}_i)p_k^D(\tilde{\mathbf{x}}_i)}{\lambda_{k|k-1}(\mathbf{z}_j)} \right] \cdot w_i \tag{3.43}$$

7. **Update hypothesis intensity**
Adjust also the number of ϕ hypotheses:

$$f_{k|k}(\phi) = \left[(1 - p_k^D(\phi)) + \sum_{j=1}^{m_k} \frac{p_k(\mathbf{z}_j \mid \phi) p_k^D(\phi)}{\lambda_{k|k-1}(\mathbf{z}_j)} \right] \cdot f_{k|k-1}(\phi) \tag{3.44}$$

8. **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 estimated expected number of targets

$$\eta_k = \sum_{i=1}^{N_k + N_{k,new}} \hat{w}_i. \tag{3.45}$$

Then compute the following probability:

$$p_S = \frac{\eta_k}{\eta_k + f_{k|k}(\phi)}. \tag{3.46}$$

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

$$N_{k+1} = \left(N_k + N_{k,new}\right) \cdot p_S. \tag{3.47}$$

The estimation of N_{k+1} at every time step prevails the particle number from growing against infinity. Given N_{k+1} any standard resampling technique for particle filtering can be used, e.g. the following:
Initialize the cumulative probability with $c_1 = 0$ and set

$$c_i = c_{i-1} + \frac{\hat{w}_i}{\eta_k}, \text{ for } i = 2, \ldots, N_k + N_{k,new}. \tag{3.48}$$

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

$$a_j = a_1 + N_{k+1}^{-1} \cdot (j-1) \tag{3.49}$$

while $a_j > c_i$,

$$i = i + 1.$$

end while. (3.50)

$$\mathbf{x}_j = \tilde{\mathbf{x}}_i \tag{3.51}$$

$$w_j = N_{k+1}^{-1} \tag{3.52}$$

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

Remark 1. In practical implementations, we found it useful to limit the number of particles to a maximum number in the prediction step. In addition we use a minimum number of particles per target in the resampling step, to ensure a good approximation.

Remark 2. After every time step t_k we generate a particle cloud, which represents the PPP over $\mathscr{S}$. The estimation of target states can also be done by applying a clustering method. To make this section self-contained we briefly present the main idea of how to use a clustering technique. 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 [10]. However, the estimated object number has a high variance. This behavior was already shown for the PHD filter [13]. The iFilter still suffers from this problem. To deal with it, one can use the adaptive resonance theory (ART) clustering [5], which estimates the number of clusters automatically, with a distance parameter as predefined user input. ART can be used to estimate the target count and the individual target states from the particle cloud. In fact, best results are achieved if one only uses a subset

$$S \subset \left\{\mathbf{x}_j, w_j\right\}_{j=1}^{N_{k+1}}, \tag{3.53}$$

with

$$(\mathbf{x}_j, w_j) \in S \text{ if } w_j \geq \tau. \tag{3.54}$$

In general we recommend the usage of the proposed state estimation scheme (steps 4 and 5) and not a clustering approach.

In situations with a low probability of detection (e.g. $p_k^D(\mathbf{x}) = 0.3$ or lower) the hybrid intensity and likelihood ratio (iLRT) filter [30] is better suited.

3.3.3 Relationship to the PHD Filter

The proposed derivation of the iFilter is very general and specializations for different applications 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 [28] and [29].

The main differences are reducible to the augmented state space $\mathscr{S}^+$. While the iFilter uses $\mathscr{S}^+ = \mathscr{S} \cup \mathscr{S}_\phi$, the PHD filter only uses $\mathscr{S}$. The basis for the on-line estimation of the intensities of the target birth and measurement clutter PPPs is the state ϕ. If, however, the birth and clutter rates are known a priori then the state ϕ 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 $\psi_k(\phi \mid \mathbf{x})$, because transition of a target into ϕ can be seen as target "death". A given a priori clutter rate can replace $\tilde{\lambda}_k(\mathbf{z}_j)$ in (3.22). An a priori birth model has to be considered in step 1 of the algorithm, see [32, 21] for details on this step of the PHD filter.

3.4 Numerical Studies

In order to analyze the performance of the iFilter we test it against its specialization the PHD filter using the OSPA-metric [26]. Some words on the strong relationship between both filters can be found in 3.3.3. In all experiments we use a sequential Monte Carlo implementation of both filters. A general description of the SMC-PHD implementation can be found in [32]. The implementation used here was published in [21]. To have a fair comparison we modified the implementation by the ideas presented in [18]. This method works well when the probability of detection $p_k^D(\mathbf{x})$ is high, as in the examples and applications presented in this paper. We used the PHD as it was described in those papers for a matched clutter rate. In the following we present results from simulated scenarios.

3.4.1 Scenario - 1

First, we analyze the behavior of the iFilter in a demanding linear scenario. Herein six inertial moved targets are placed in an area $A = [-500,500] \times [-500,500]$. The unit is assumed to be meters. The state space is $\mathscr{S} \subset \mathbb{R}^4$, 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 $\mathscr{Z} \subset \mathbb{R}^2$. New measurements occur for the sake of simplicity every second. The measurement noise is white gaussian noise with a standard deviation $\sigma_x = \sigma_y = 15$. The probability of detection is set equal for all states to $p_k^D(\mathbf{x}) = 0.95,\ \mathbf{x} \in \mathscr{S}$. Target placement and direction of movement is visualized in Figure 3.1. 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 transition probabilities were set to $\psi_k(\mathbf{x} \mid \phi) = 0.2$, $\psi_k(\phi \mid \phi) = 0.01$ and $\psi_k(\phi \mid \mathbf{x}) = 0.1$ and the probabilities of detection were set to $p_k^D(\mathbf{x}) = 0.95$ and $p_k^D(\phi) = 0.3$. The number of targets in the following experiments are the result of (3.45). The number of states is the number of states, which were extracted in Step 4 of the SMC-iFilter algorithm. Both numbers are average results after 500 Monte Carlo trials.

In a first experiment the iFilter is tested in a case where no clutter measurements are present. The iFilter models its birth process through the state ϕ, so if no clutter measurements are present $f_{k|k}(\phi)$ should tend to be zero. The question that then arises is: Will the iFilter be able to deal with this and produce a reliable birth model? We run 500 Monte Carlo simulations on the above scenario. As a modification we set $p_k^D(\phi) = 0$ for this experiment. Figure 3.2 demonstrates that the iFilter is capable to produce a good target birth model if no clutter is present. In time step 15 a new target appears, because of this $f_{k|k}(\phi)$ reaches a value above one in this time step (this is denoted as 'no. Phi" in Figure 3.2). From this increase the filter can model a new target state in the next time step. Similar behavior can be observed in time step 30, where two new targets appear. $f_{k|k}(\phi)$ reaches a value of above two and from

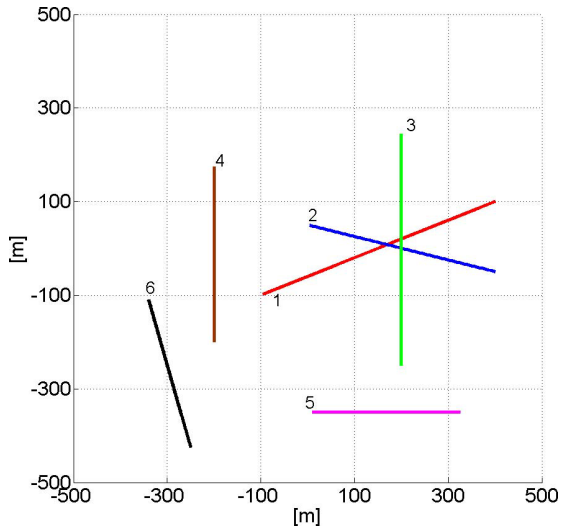

Fig. 3.1. Linear scenario used for performance evaluation. Six targets move inertially. The individual starting points of each target correspond to the denoted target number. Targets 1 - 3 are present for all time steps. Target 4 is presented between time step 15 and 90. Target 5 and 6 are present between time step 30 and 75.

this it can produce new target states. Even when the targets disappear the correct number targets is estimated.

In the second experiment, we will investigate how a estimated clutter rate can improve the results from a multi-target tracker in comparison to results gained from a multi-target tracker that needs a priori knowledge about the clutter rate. The later will be an SMC implementation of the PHD filter. In its standard formulation, the PHD needs exact knowledge about the underlying mean clutter rate for a given scenario. However, the clutter rate is hard to know in advance for a given real scenario and sensor setup. In total we perform three experiments with low, middle and high clutter rates. For each we evaluate the mean results after 500 Monte Carlo trials on the linear scenario, see Figure 3.1. The number of clutter measurements is estimated following a Poisson distribution with the mean value $|A| \cdot \rho_A$:

$$p(n_c) = \frac{1}{n_c!}(A \cdot \rho_A)^{n_c} \exp(-|A| \cdot \rho_A), \tag{3.55}$$

with $|A|$ denoting the volume of a observed area and ρ_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. Clutter measurements are generated by a i.i.d. process. We will match the PHD filter to a middle clutter rate to see its behavior in situations with less clutter on the one hand and more clutter on the other

hand. In addition we will use the iFilter, which does not need any prior knowledge about the clutter rate.

Figure 3.4 displays the mean OSPA values for both filters after 500 Monte Carlo trials on a middle clutter rate. It can be observed that the achieved results are similar through the whole scenario. This is an obvious result, since the PHD is matched to the used middle clutter rate and the iFilter successfully learned the clutter rate. More important is the fact that the iFilter, although it has a more complicated estimation problem to solve (additional estimation of the clutter rate), reaches the same results as the PHD filter, which has the advantage of exact a priori knowledge. A close look on Figure 3.3 reveals another interesting effect of the PHD filter. The estimated expected number of targets (which is the integral of the intensity over the state space) is biased in comparison to the ground truth. The number of state estimates, which was produced by the strategy proposed in [18] (c.f. step 4 of the iFilter algorithm), improves the result of the PHD filter. Therefore we can claim that if not using this state estimation strategy the iFilter will outperform the PHD filter even in a case where it knows the exact clutter rate. Proofs for this statement were presented in [24].

This misbehavior of the PHD arises from the fact that it assumes a mean clutter rate for all time steps. Let us assume that we have as mean value one clutter measurement per scan. If we use now a realization of a Poisson distribution with this mean value, the probability to get zero clutter measurement or more than one clutter measurement is exactly $1 - e^{-1} = 0.63$, and so these events can occur frequently leading to the errors in the PHD. The iFilter on the other hand estimates the clutter rate for every time step individually and is, because of this, robust against a changing number of clutter measurements. This can be seen in Figure 3.3, where the number of targets and the number of states are very close to the true value for

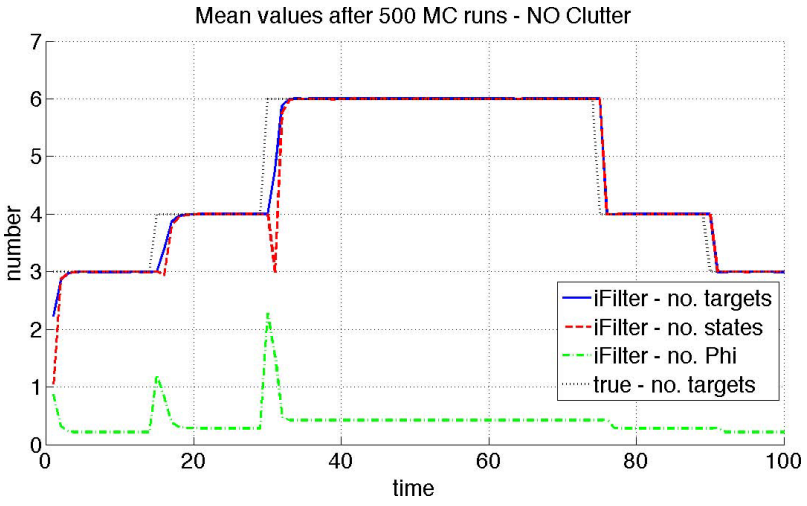

Fig. 3.2. Mean estimated target and state number after 500 Monte Carlo trials on the linear scenario.

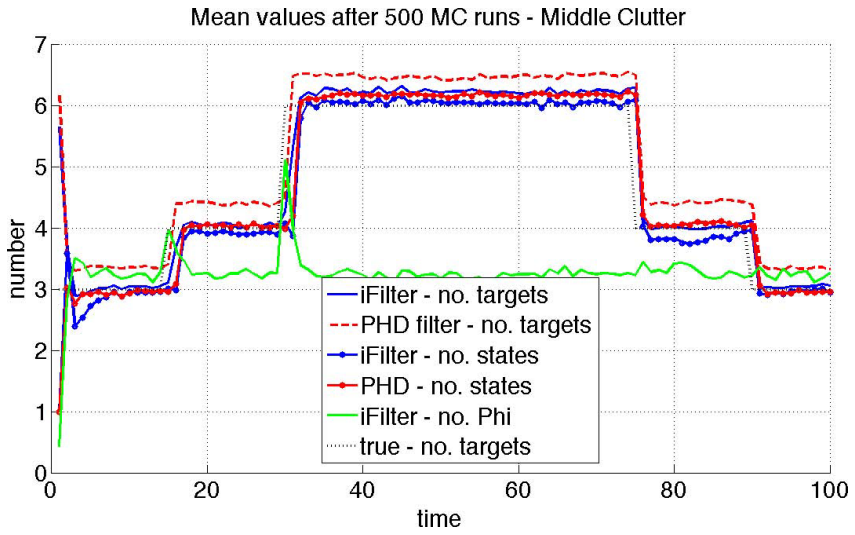

Fig. 3.3. Mean estimated target and state number after 500 Monte Carlo trials on the linear scenario with middle clutter.

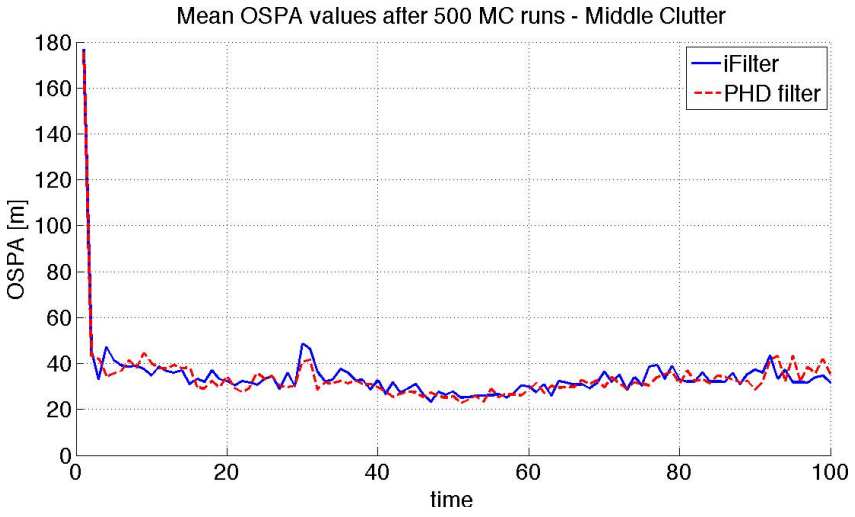

Fig. 3.4. Mean OSPA values over time after 500 Monte Carlo trials on the linear scenario with middle clutter.

all time steps. Further we can read from this figure that the mean number of clutter measurements for 500 Monte Carlo trials was estimated as 3.3 ('no. Phi" in Figure 3.3), which corresponds to the ground truth. Again we can observe that the intensity $f_{k|k}(\phi)$ increases, when new targets appear. So even in a clutter scenario the birth model of the iFilter works well.

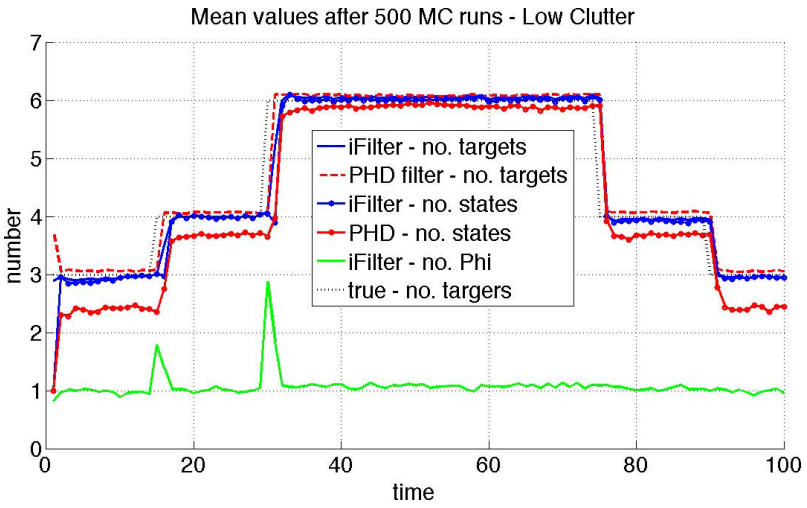

Fig. 3.5. Mean estimated target and state number after 500 Monte Carlo trials on the linear scenario with low clutter.

In the following we changed the clutter parameter of this scenario to a low value. Results obtain for both filters are illustrated in figures 3.5 and 3.6. The first thing to notice is the underestimation of the correct number of states for the PHD. This effect can be mainly observed in those time steps, where only few targets are present. Here the PHD filter matches the more measurements to clutter then the truth is. In the moment, where more targets appear, the effect is reduced but still visible. The impact can be seen in the mean OSPA values, c.f. Figure 3.6. The iFilter adapts very reliably to the changed clutter rate and gives better results.

The final experiment will use a high clutter rate. Again the PHD filter is matched to a middle clutter rate and the iFilter has no information about the clutter scenario. The corresponding results are depicted in the figures 3.7 and 3.8. Especially the PHD filter overestimates the number of targets and thus its OSPA values are higher than the corresponding values of the iFilter. Also worth mentioning is that estimated number of targets of the iFilter is slightly biased in comparison to the ground truth. But the estimated number of states keeps close to the true values. The PHD on the other hand has a strongly biased estimated number of targets.

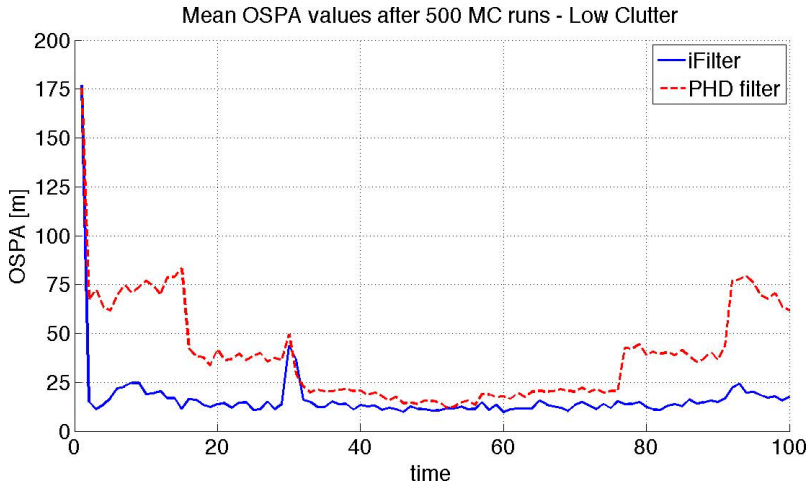

Fig. 3.6. Mean OSPA values over time after 500 Monte Carlo trials on the linear scenario with low clutter.

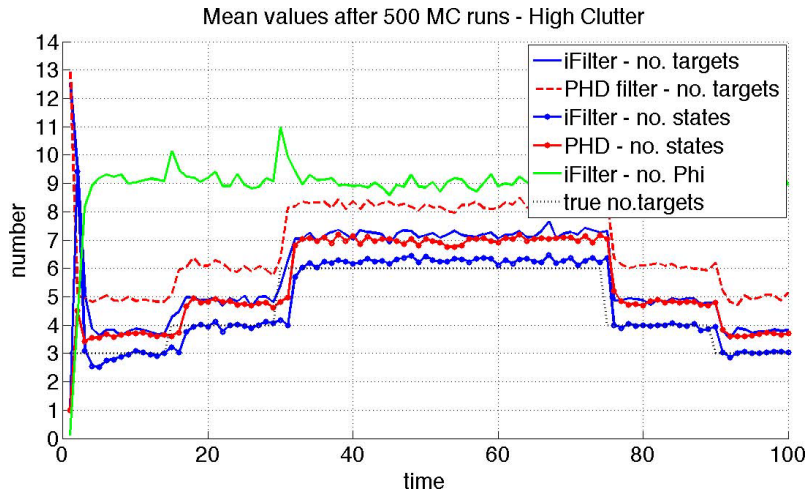

Fig. 3.7. Mean estimated target and state number after 500 Monte Carlo trials on the linear scenario with high clutter.

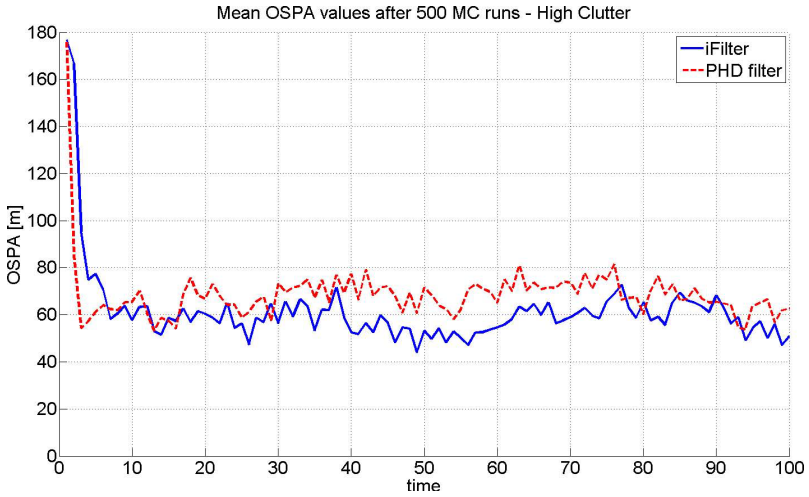

Fig. 3.8. Mean OSPA values over time after 500 Monte Carlo trials on the linear scenario with high clutter.

In general we can say that for both filters the used state estimation improves the results and makes both filters more robust against clutter. The iFilter however performs better, even in cases where the PHD filter is matched to the actual clutter rate.

Another interesting comparison can be seen in Figure 3.9. Here the number of particle used to achieve the above results is presented. The PHD filter has a constant particle number which stays the same for all time steps. The iFilter adjusts the number according to the actual number of targets. One can easily see how the number adjusts, when new targets appear or disappear from the scene. The lower number of particle of the iFilter leads to a lower computation time. A run time comparison can seen in Table 3.1. The presented processing times where achieved on a Intel Core2Duo 2.53GHz processor with 4GB of RAM.

Table 3.1. Mean runtimes for processing one time step. Values computed over 500 Monte Carlo trials and for all time steps of the linear scenario with a middle clutter rate.

	processing time (msec)	speedup
SMC-PHD filter	12.645	1.0
SMC-iFilter	7.46467	1.7

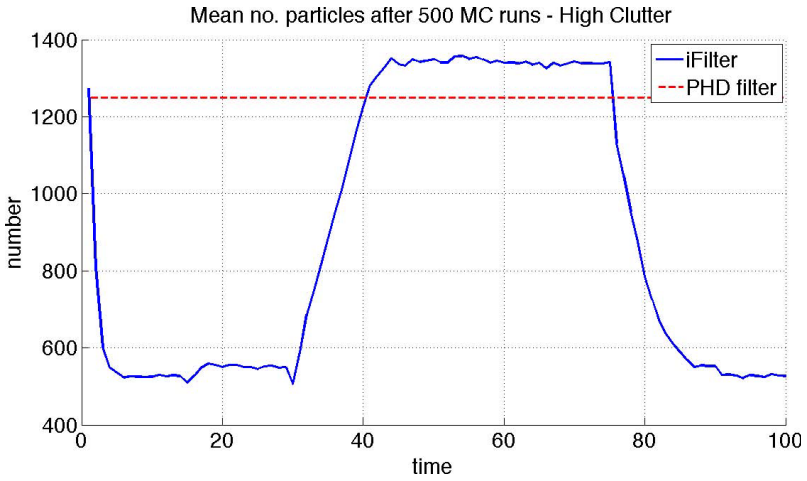

Fig. 3.9. Mean number of particles used over time after 500 Monte Carlo trials on the linear scenario with high clutter. New targets appear in time steps 15 and 30 and disappear in time step 75 and 90.

3.4.2 Scenario - 2

In the following both filters are tested on a non-linear scenario, c.f. Figure 3.10. Here we use bearing measurements (azimuth and elevation) to estimate position and velocity of multiple targets. The measurement likelihood is defined through:

$$p(\mathbf{z}|\mathbf{x}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{z}-h(\mathbf{x}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}-h(\mathbf{x}))\right), \tag{3.56}$$

with $\boldsymbol{\Sigma}$ the covariance matrix of the measurement noise and

$$h(\mathbf{x}) = \begin{pmatrix} \arctan\left(\frac{\mathbf{x}(1)-\mathbf{x}_{\text{obs}}(1)}{\mathbf{x}(2)-\mathbf{x}_{\text{obs}}(2)}\right) \\ \frac{\pi}{2} + \arctan\left(\frac{\mathbf{x}(3)-\mathbf{x}_{\text{obs}}(3)}{\sqrt{(\mathbf{x}(1)-\mathbf{x}_{\text{obs}}(1))^2+(\mathbf{x}(2)-\mathbf{x}_{\text{obs}}(2))^2}}\right) \end{pmatrix}. \tag{3.57}$$

An observer performs a half circle flight over a region of interest. For discrete time steps we obtain bearing measurements from three targets and additionally some clutter measurements. Details on this scenario can be found in [21]. The covariance matrix $\boldsymbol{\Sigma}$ was chosen according to sensor models for small antenna arrays, i.e. high angular error. The transition probabilities where set to $\psi_k(\mathbf{x} \mid \phi) = 0.2$, $\psi_k(\phi \mid \phi) = 0.01$ and $\psi_k(\phi \mid \mathbf{x}) = 0.1$ and the probabilities of detection where set to $p_k^D(\mathbf{x}) = 0.95$ and $p_k^D(\phi) = 0.3$. Again, we performed a Monte Carlo simulation with 500 trials, c.f. Figure 3.11. It can be easily seen that also in the nonlinear scenario the iFilter produces lower OSPA values compared to the PHD filter.

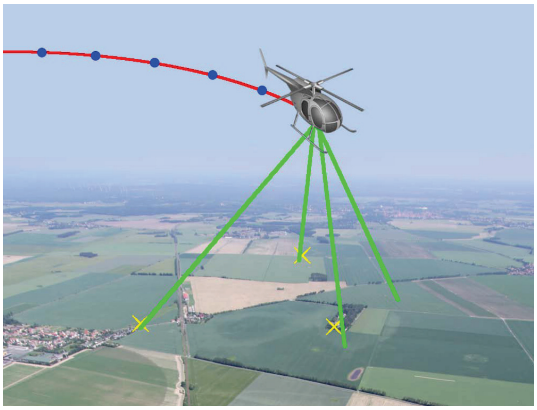

Fig. 3.10. Non linear scenario used in our simulations and real world experiments. A possible observer path is illustrated by the dotted curve, whereby the dots represent the discrete measurement positions. The crosses are objects of interest in this scene, which should be localized and tracked from the algorithm by bearing data. The latter is represented by rays which point in the directions resulting from processing the sensor output.

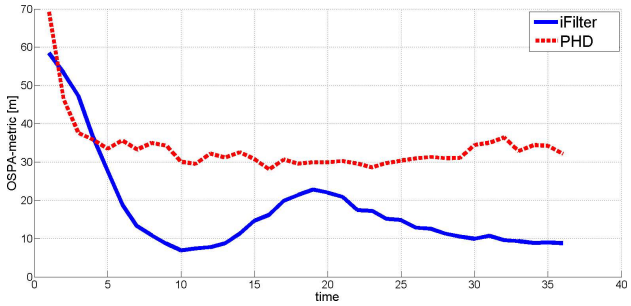

Fig. 3.11. OSPA-metric for 500 Monte Carlo runs on a non-linear scenario.

3.5 Applications

In this section, we present two example application for the usage of the iFilter for multi-target tracking. The first one is bearings-only tracking, where the states of individual targets have to be estimated from azimuth and elevation measurements. In practice, this problem involves highly non-linear measurement equations and high sensor errors with geometry depended bias. The second application is tracking in video sequences with a broad space of following application, e.g. security, surveillance and behavior analysis. Here we combine the methodology of the iFilter with a high-accuracy optical flow algorithm. In addition, we show how target labeling can be introduced to the iFilter.

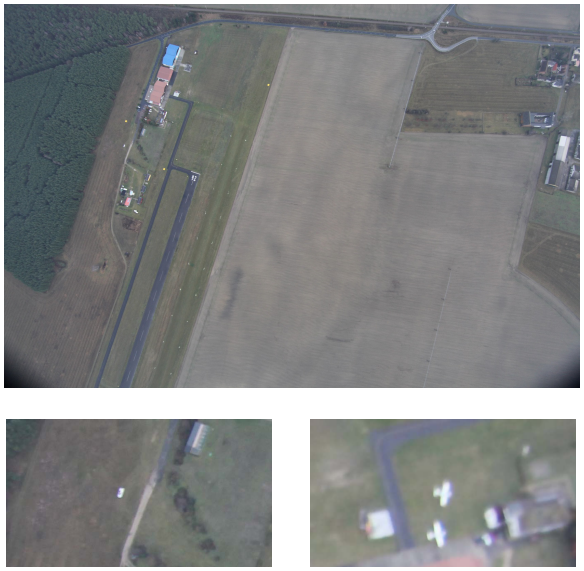

Fig. 3.12. Typical input images produced by our camera system. Top row: scaled image used for processing. Bottom row: cut in original size on a car and two airplanes.

3.5.1 Bearings-Only Tracking

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 [25]. 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 statistical 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 [19]. This detection procedure uses shape and color information to find objects in color images. For the experiments presented here we limited ourselves to cars and airplanes (c.f. Figure 3.12). Once an 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 [22].

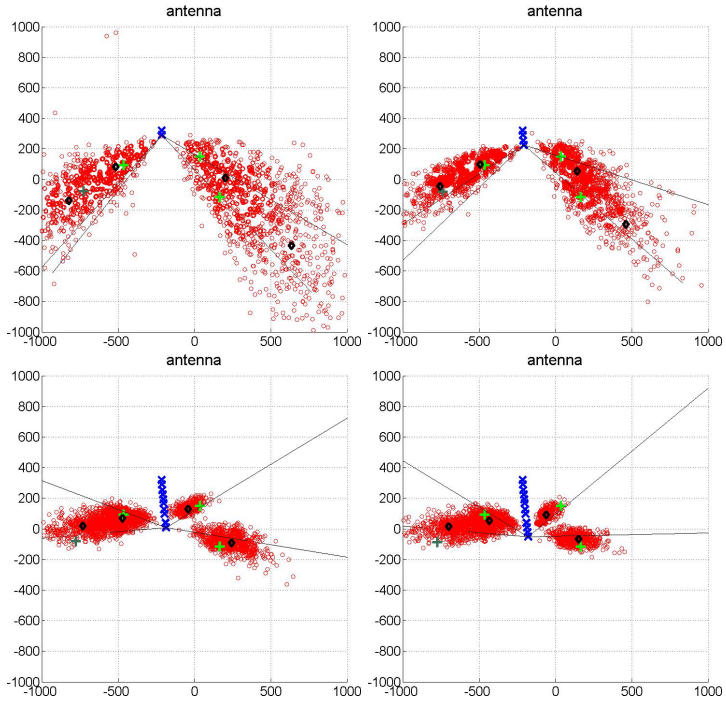

Fig. 3.13. Particle set evolution for an antenna array system at different time steps. The x's represent the observer position where bearing measurement were produced. The corresponding bearings are represented by rays. The individual particles are illustrated as circles, whereby the estimated localizations are displayed as diamonds and the ground truth is represented by crosses. The leftmost object is moving to the left and the three others are stationary targets. For a better perspective this Figure only shows the top view of the 3D scenario.

The transition probabilities of the iFilter were set to $\psi_k(\mathbf{x} \mid \phi) = 0.2$, $\psi_k(\phi \mid \phi) = 0.01$ and $\psi_k(\phi \mid \mathbf{x}) = 0.1$ and the probabilities of detection were set to $p_k^D(\mathbf{x}) = 0.95$ and $p_k^D(\phi) = 0.3$.

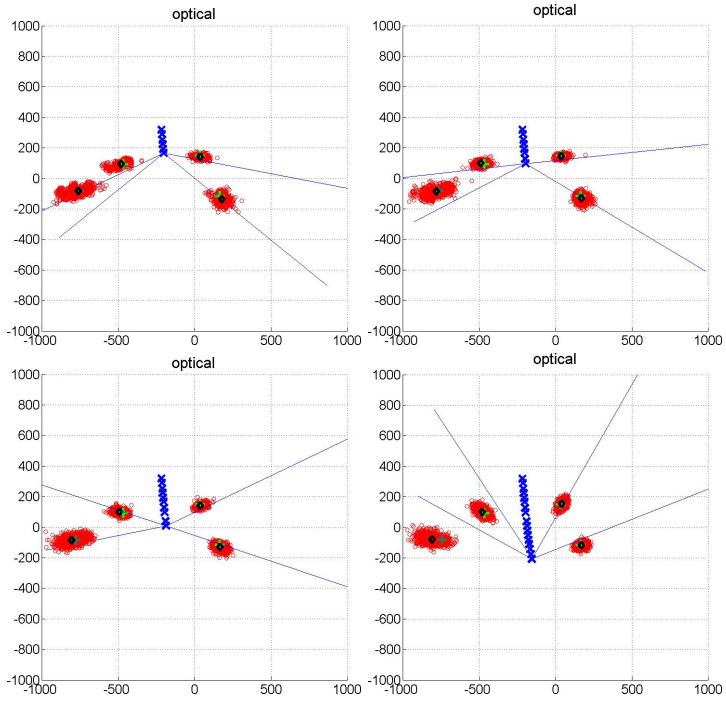

Fig. 3.14. Particle set evolution for an optical system at different time steps. The x's represent the observer position where bearing measurement were produced. The corresponding bearings are represented by rays. The individual particles are illustrated as circles, whereby the estimated localizations are displayed as diamonds and the ground truth is represented by crosses. The leftmost object is moving to the left and the three others are stationary targets. For a better perspective this Figure only shows the top view of the 3D scenario.

The results for the optical and antenna system are visualized in Figures 3.14 and 3.13, 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 good results and estimates the expected 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 3.11) also holds for real data. In [21] comparable results with a PHD filter for this data was shown.

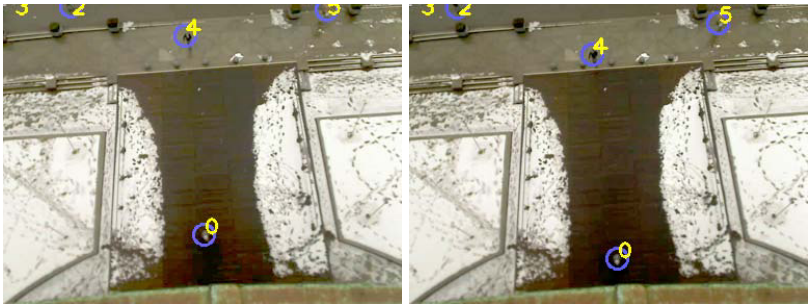

Fig. 3.15. Tracking an unknown number of people using high-accuracy optical flow. Tracking results as blue circles and person ID in yellow.

3.5.2 Video Tracking

Multi-object tracking using a monocular system is a challenging but very important problem in many computer vision applications. The aim is to estimate the number and the state information of every object for each time step in an image sequence. The problem becomes challenging when the number of objects is unknown and variable. Here we will use the proposed SMC-iFilter to estimate the number of targets and their states.

Algorithms based on the Joint Probabilistic Data Association filter (JPDAF) [3] tend to merge tracking results produced by closely spaced objects. This drawback cannot be observed when using an iFilter.

In [34, 11] the authors use a PHD filter for multi object tracking, with the drawback that only position information, gained from a detector, is used, so that the filter must estimate the velocity indirectly, which reduces the robustness of the filter. In this work we extend the classical iFilter to deal with image data and velocity information gained from optical flow. For every object tracking task some kind of measurement is needed. Using optical flow we directly obtain velocity measurements for every pixel. Unfortunately, this does not provide any information about the positions of objects in the scene. Fortunately, there has been a rapid progress in the field of object detection strategies [6, 20]. Since not all of these strategies are able to run in real-time we will present a fast strategy for moving object detection based on real-time optical flow. The main idea of our tracking algorithm is to run an object detector and additionally compute the optical flow information. This gives us two kinds of measurements, which can be used for a stable and robust multi-object tracking, sample results can be seen in Figure 3.15. Parts of this work have been previously published [23].

Optical Flow

The estimation of the optical flow between two images is a well-studied problem in low-level vision. A diverse range of optical flow estimation techniques have been developed and we refer to the survey [35] for a detailed review. Taking into account the so-called Middlebury dataset [1] the discontinuity-preserving variational models based on Total Variation (TV) regularization and L^1 data terms are among the most accurate flow estimation techniques. Because of this fact, we will use in this context the estimation technique proposed by Werlberger et. al. [36]. To make this chapter self-contained we briefly reflect their work.

For two input images $I_0, I_1 : \Omega \subset \mathbb{R}^2 \to [0,1]$ the optical flow model can be stated as

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} \sum_{d=1}^{2} |\nabla u_d| + \lambda |\rho(\mathbf{u}(\mathbf{x}))| \mathrm{d}\mathbf{x} \right\}, \tag{3.58}$$

with $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))^T$, $u_d : \Omega \to \mathbb{R}$, the free parameter λ to balance the relative weight of data and regularization term and $\rho(\mathbf{u}(\mathbf{x})) = \mathbf{u}(\mathbf{x})^T \nabla I_1(\mathbf{x}) + I_1(\mathbf{x}) - I_0(\mathbf{x})$ the optical flow constrained equation. To improve the results and the accuracy the authors extend this approach using anisotropic Huber regularization. To still be able to perform a minimization a Legendre-Fenchel (LF) dual transform is needed. The final energy functional is then

$$\min_{\mathbf{u},\mathbf{v}} \sup_{|\mathbf{p}_d| \leq 1} = \left\{ \int_{\Omega} \sum_{d=1}^{2} \left[\left(D^{\frac{1}{2}} \nabla u_d \right) \mathbf{p}_d - \varepsilon \frac{|\mathbf{p}_d|^2}{2} + \frac{1}{2\Theta} (u_d - v_d)^2 \right] + \lambda |\rho(\mathbf{v}(\mathbf{x}))| d\mathbf{x} \right\}. \tag{3.59}$$

This approach has several benefits: firstly, the energy functional is convex, which leads to globally optimal solutions, and, secondly, the minimization can be scheduled in parallel leading to a real-time computation. use the results without massive time drawbacks. Using this approach we can compute for every pixel $\mathbf{x} \in \Omega$ and each time step k the velocity $\mathbf{u}_k(\mathbf{x}) = (u_1, u_2)^T$ of this pixel.

Moving Object Detector

In this subsection we present a fast object detector, which is designed to detect moving objects. Given the flow field $\mathbf{u}_k$ at a given time step k, we can compute the probability, individually for every pixel that it belongs to a moving object:

$$p_m(\mathbf{x}) = 1.0 - \exp\left(-\frac{1}{2} \frac{(\|\mathbf{u}((x))\|_2 - \mu)^2}{\sigma^2} \right). \tag{3.60}$$

Here μ and σ correspond to a normal distribution indicating that a pixel does not move. Using a stationary camera μ would be zero. Using a flying platform with downward-looking camera μ would correspond to the actual velocity of the platform. A typical value for σ in our experiments is 0.5. Given this probability image $p_m : \Omega \to [0,1]$ we can compute the center of gravity for every region with a high probability of movement. Examples of this detector can be seen in Figure 3.16.

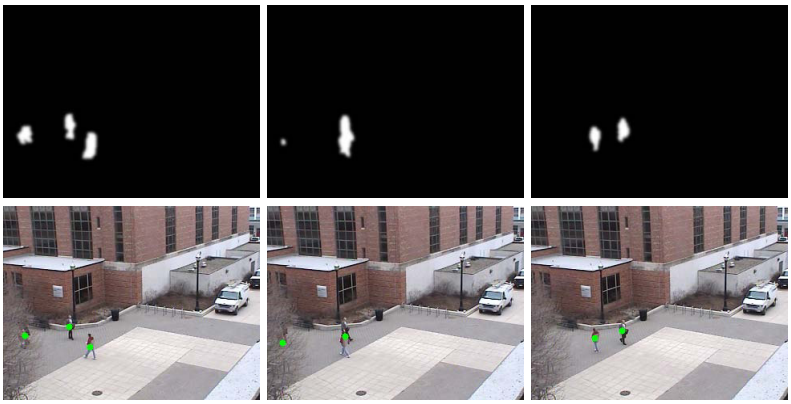

Fig. 3.16. Moving object detection for different frames in the image sequence. Top row: smoothed probability image for pixel movement; bottom row: position measurement displayed as black and white circles.

Muti-target Tracking and Labeling

We implemented the iFilter according to Section 3.3.2. In the following the state of an individual object will be represented by $\mathbf{x}_k \in \mathbb{R}^4$, with two random entries for the position and two random entries for the velocities. Each measurement $\mathbf{z}_k \in \mathbb{R}^4$ is represented analogous. The measurement and state unit is pixel. For the sake of simplicity we assume that the object motion model of each target is linear with a constant velocity. Since we use a high-accuracy optical flow with a high frame rate (e.g. 30 frames per second) we do not need a more complicated motion model in our experiments. With this the object state prediction can be written as:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{I}_2 & \Delta T \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{pmatrix} \mathbf{x}_{k-1} + \mathbf{s}_k, \tag{3.61}$$

with $\mathbf{s}_k$ a zero mean Gaussian white process noise, ΔT the time difference between step k and $k-1$. $\mathbf{I}_2$ denotes the identity matrix for two dimensions and $\mathbf{0}_2$ a 2x2 matrix with zeros. The likelihood function is given by:

$$p(\mathbf{z}|\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{x})^T \boldsymbol{\Sigma} (\mathbf{z}-\mathbf{x})\right), \tag{3.62}$$

with $\boldsymbol{\Sigma}$ the covariance matrix of the measurement noise.

At every time step k we have $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_k}$ as a particle-based approximation of the intensity over $\mathscr{S}^+$. The prediction, update and resampling for every time step and new measurements is done following the work in [24], see Section 3.3.2.

To establish an individual object trajectory we have to label each object correctly. We use two kinds of information: object state and the color distribution of the object.

For both information, we will use a likelihood-type function measuring the confidence that two objects from consecutive time steps $k-1$ and k are identical. The values of these likelihoods will be in the range of 0 (not identical) and 1 (identical). Let us assume that m is an object from the time step $k-1$ and n is a object from the time step k: then we can predict the object state of m using (3.61), so that we get $\tilde{m}$. The distance is defined as $d(\tilde{m},n) = \|\mathbf{x}^{\tilde{m}} - \mathbf{x}^n\|_2$, with $\mathbf{x}^{\tilde{m}}$ and $\mathbf{x}^n$ the state vectors of the objects $\tilde{m}$ and n. The likelihood function is then

$$L_{\text{state}}(m,n) = \exp\left(-\frac{(d(\tilde{m},n))^2}{2\sigma_d^2}\right), \tag{3.63}$$

with σ_d the standard deviation of the distance information.

The likelihood function for the color measurement is based on the idea of similarity measures on color histograms, which has the benefit to be robust against non-rigidity, rotation and partial occlusions [14]. Suppose that the distribution is discretized into η bins. The color histogram $\mathbf{p}(\mathbf{x}) = \{p(\mathbf{x}^{(c)})\}_{c=1,\ldots,\eta}$ at position $\mathbf{x}$ is calculated as

$$p(\mathbf{x}^{(c)}) = f \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x})} g\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\alpha}\right) \delta(h(\mathbf{x}_j) - c). \tag{3.64}$$

In (3.64) f is a normalization factor, α is the scaling factor, $\mathcal{N}(\mathbf{x})$ denotes the neighborhood of pixel $\mathbf{x}$, δ is the Kronecker delta function and $g(.)$ is a weighting function given by

$$g(r) = \begin{cases} 1-r^2, & r<1 \\ 0, & \text{otherwise} \end{cases}. \tag{3.65}$$

$h(\mathbf{x})$ is a function, which assigns the color at location $\mathbf{x}$ to the corresponding bin. To measure the similarity between two color distributions, which are denoted by $\mathbf{p}(\mathbf{x}) = \{p(\mathbf{x}^{(c)})\}_{c=1,\ldots,\eta}$ and $\mathbf{q}(\mathbf{x}) = \{q(\mathbf{x}^{(c)})\}_{c=1,\ldots,\eta}$, we use the Bhattacharyya coefficient.

Let $\mathbf{p}_{\tilde{m}}$ and $\mathbf{q}_n$ be the color distribution of the objects $\tilde{m}$ and n, then the likelihood is:

$$L_{\text{color}}(m,n) = \sum_{c=1}^{\eta} \sqrt{p_{\tilde{m}}^{(c)} q_n^{(c)}}. \tag{3.66}$$

The likelihood that the objects m and n are identical, is a weighted sum over both likelihoods

$$L(m,n) = w_p L_{\text{state}}(m,n) + w_c L_{\text{color}}(m,n). \tag{3.67}$$

Using this measurement (3.67) we can compute the similarity between every object from the time step $k-1$ and every object from the time step k. If the measurement exceeds a threshold value, then the objects are labeled as identical. If a new object does not match any other object from the previous time step, then a new object is added to the database. Objects that are not supported by new measurements over the time are deleted from the database, assuming that the object has left the scene.

Results

In this section, we present experimental results of our tracking algorithm. The transition probabilities of the iFilter where set to $\psi_k(\mathbf{x} \mid \phi) = 0.2$, $\psi_k(\phi \mid \phi) = 0.01$ and $\psi_k(\phi \mid \mathbf{x}) = 0.1$ and the probabilities of detection where set to $p_k^D(\mathbf{x}) = 0.95$ and $p_k^D(\phi) = 0.3$. The image sequence used in Figure 3.17 was published in [7]. The top row of it shows the position measurement. In frame 61 the motion field of two persons merges (c.f. Figure 3.16), so that the detector measures only one moving object for a couple of frames. Because of the proposed labeling and the iFilter, we are able to track and label both persons correctly when the motion fields splits again. This can be seen in the bottom row. The center of the blue circle corresponds to the position information gained though the state estimation step. The radius of this circle is fixed and only used for presentation. The yellow number is the ID of a person. In frame 86, the ID of person 2 is still displayed to indicate the last known position of this person. For this scene, we had a hand-labeled ground truth. The mean position error between the proposed algorithm and the ground through lies by 2.68 pixel with a standard deviation of 1.5 pixel.

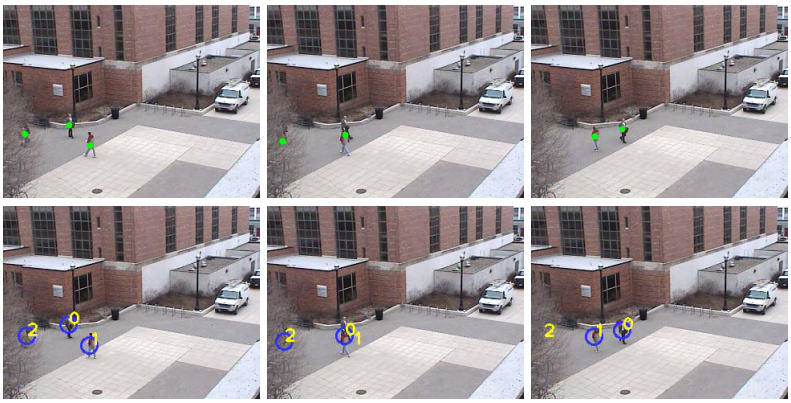

Fig. 3.17. Tracking result. Top row: position measurement displayed as black and white cirles in the image sequence. Bottom row: Position estimation of every object plotted as blue circle with fixed radius and ID-number of every object in yellow.

Tracking and labeling results for a different scene can be seen in Figure 3.15. The image sequence used here was published in [15]. The challenge in this scene lies in the high false alarm rate of about 5-10%, which comes from additional noise produced through snowfall. Nevertheless, the proposed strategy could track and label all persons in this scene correctly. We computed the achieved runtime as frame rate means from the individual runtimes for each frame in the scenes from the Figures 3.17 and 3.15: 500 particles 61.74fps, 1000 particles 54.43fps and 1500 particles 31.85fps. The results were computed on an Intel Q8220 Qaud Core CPU with 4GB RAM using a single core implementation.

3.6 Conclusions

This chapter presented an elementary introduction to PPPs. Their application for the multi-target tracking problem with an unknown number of targets was illustrated, which lead to the iFilter. For this concept, an improved sequential Monte Carlo implementation was derived. To verify the theory and analyze the filter behavior, various numerical studies were performed. 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 the literature, which are published up to now. The only drawback of the iFilter is its slight slower initial convergence in comparison to the PHD filter.

In addition two applications were presented. They demonstrate the good performance of the iFilter in real world problems. The first presented the usability of the iFilter in a demanding estimation problem for real bearings-only data with high systematic and statistical errors. The second showed that the filter can be well used in time critical estimation problems like multi-target tracking in video sequence with a high frame rate.

References

1. Baker, S., Schastein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV (2007)
2. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic, San Diego (1988)
3. Bar-Shalom, Y., Fortmann, T., Scheffe, M.: Joint probabilistic data association for multiple targets in clutter. In: Conf. on Information Sciences and Systems (1980)
4. Callen, H.: Thermodynamics and an Introduction to Thermostatistics, 2nd edn. Wiley, New York (1985)
5. Carpenter, G., Grossberg, S.: ART 2: Self-organizing stable category recognition codes for analog input patterns. Applied Optics 26(23), 4919–4930 (1987)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
7. Davis, J., Sharma, V.: Background-subtraction using contour-based fusion of thermal and visible imagery. Computer Vision and Understanding 106(2-3), 162–182 (2007)
8. Erdinc, O., Willett, P., Bar-Shalom, Y.: Probability hypothesis density filter for multitarget multisensor tracking. In: Proc. of the 8th International Conference on Information Fusion (FUSION), Philadelphia, PA, USA (July 2005)
9. Fortmann, T., Bar-Shalom, Y., Scheffe, M.: Sonar tracking of multiple targets using joint probabilistic data association. IEEE J. Oceanic Eng. 8, 173–184 (1983)
10. Lloyd, S.: Least squares quantziation in pcm. IEEE Trans. Inform. Theory 28(2), 129–137 (1982)
11. Maggio, E., Taj, M., Cavallaro, A.: Efficient multi-target visual tracking unsing random finite sets. IEEE Trans. on TCSVT 18(8), 1016–1027 (2008)

12. Mahler, R.: Multitarget Bayes filtering via first-order multitargets moments. IEEE Trans. Aerosp. Electron. Syst. 39(4), 1152–1178 (2003)
13. Mahler, R.: PHD filters of higher order in target number. IEEE Trans. Aerosp. Electron. Syst. 43(4), 1523–1543 (2007)
14. Nummiaro, K., Koller-Meier, E., Gool, L.V.: An adaptive color-based particle filter. Image and Vision Computing 21(1), 99–110 (2002)
15. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behaviour for multi-target tracking. In: ICCV (2009)
16. Reid, D.B.: An algorithm for tracking multiple targets. IEEE Trans. Autom. Control 24(6), 843–854 (1979)
17. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman filter: Particle filters for tracking applications. Artech House (2004)
18. Ristic, B., Clark, D., Vo, B.-N.: Improved SMC implementation of the PHD filter. In: Proc. of the 13th International Conference on Information Fusion, Edinburgh, UK (July 2010)
19. Schikora, M.: 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)
20. Schikora, M.: Global optimal multiple object detection using the fusion of shape and color information. In: EMMCVPR (2009)
21. Schikora, M., Bender, D., Cremers, D., Koch, W.: Passive multi-object localization and tracking using bearing data. In: 13th International Conference on Information Fusion, Edinburgh, UK (July 2010)
22. Schikora, M., Bender, D., Koch, W., Cremers, D.: Multitarget multisensor localization and tracking using passive antennas and optical sensors. In: Proc. SPIE, Security + Defense, vol. 7833 (2010)
23. Schikora, M., Koch, W., Cremers, D.: Multi-object tracking via high accuracy optical flow and finite set statistics. In: Proc. of the 36th International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prag, Czech Republic. IEEE (May 2011)
24. Schikora, M., Koch, W., Streit, R., Cremers, D.: Sequential Monte Carlo method for the iFilter. In: Proc. of the 14th International Conference on Information Fusion (FUSION), Chicago, IL, USA (July 2011)
25. Schmidt, R.O.: Multiple emitter location and signal parameter estimation. In: Proc. RADC Spectrum Estimation Workshop, Griffith AFB, pp. 243–258 (1979)
26. Schumacher, D., Vo, B.-T., Vo, B.-N.: A consistent metric for performance evaluation of multi-object filters. IEEE Trans. Signal Processing 58(8), 3447–3457 (2008)
27. Sidenbladh, H.: Multi-target particle filtering for probability hypothesis density. In: International Conference on Information Fusion, Cairns, Australia, pp. 800–806 (2003)
28. Streit, R.: Multisensor multitarget intensity filter. In: 11th International Conference on Information Fusion (2008)
29. Streit, R.: Poisson Point Processes: Imaging, Tracking, and Sensing. Springer (2010)
30. Streit, R., Osborn, B., Orlov, K.: Hybrid intensity and likelihood ratio tracking (iLRT) filter for multitarget detection. In: Proceedings of the 14th International Conference on Information Fusion (FUSION), Chicago, IL, USA (July 2011)
31. Streit, R., Stone, L.: Bayes derivation of multitarget intensity filters. In: 11th International Conference on Information Fusion (2008)
32. Vo, B.-N., Singh, S., Doucet, A.: Sequential Monte Carlo methods for multi-target filtering with random finite sets. IEEE Trans. Aerosp. Electron. Syst. 41(4), 1224–1245 (2005)

33. Vo, B.-T., Ma, W.-K.: The Gaussian mixture probability hypothesis density filter. IEEE Trans. Signal Processing 55(11), 4091–4104 (2006)
34. Wang, Y., Wu, J., Kassim, A., Huang, W.: Tracking a variable number of human groups in video using probability hypothesis density. In: ICPR (2006)
35. Weickert, J., Bruhn, A., Brox, T., Papenberg, N.: A survey on variational optic flow methods for small displacements. Mathematical Models for Registration and Applications to Medical Images, 103–136 (2006)
36. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: BMVC, London, UK (September 2009)
37. Zajic, T., Mahler, R.: A particle-systems implementation of the PHD multi-target tracking filter. In: Proc. SPIE, Signal Processing, Sensor Fusion Target Recognition XII, vol. 5096(4), pp. 291–299 (2003)