Appears in *International Conference on Information Fusion*, pp 800–806, Cairns, Australia 2003.

Multi-Target Particle Filtering for the Probability Hypothesis Density

Hedvig Sidenbladh

Department of Data and Information Fusion
Division of Command and Control Systems
Swedish Defence Research Agency
SE–172 90 Stockholm, Sweden
hedvig@foi.se

Abstract – *When tracking a large number of targets, it is often computationally expensive to represent the full joint distribution over target states. In cases where the targets move independently, each target can instead be tracked with a separate filter. However, this leads to a model-data association problem. Another approach to solve the problem with computational complexity is to track only the first moment of the joint distribution, the* probability hypothesis density (PHD). *The integral of this distribution over any area S is the expected number of targets within S. Since no record of object identity is kept, the model-data association problem is avoided.*

The contribution of this paper is a particle filter implementation of the PHD filter mentioned above. This PHD particle filter is applied to tracking of multiple vehicles in terrain, a non-linear tracking problem. Experiments show that the filter can track a changing number of vehicles robustly, achieving near-real-time performance.

Keywords: Bayesian methods, finite set statistics, particle filters, random sets, probability hypothesis density, sequential Monte Carlo, terrain tracking

1 Introduction

When tracking multiple targets in general, the size of the state-space for the joint distribution over target states grows exponentially with the number of targets. When the number of targets is large, this makes it impossible in practice to maintain the joint distribution over target states. However, if the targets can be assumed to move independently, the joint distribution does not have to be maintained. A straight-forward method is to assign a separate filter to each target [3, 16]. A drawback with this approach is that it leads to a model-data association problem [3].

A mathematically principled alternative to the separate filter approach is to propagate only the first moment of the joint distribution, the *probability hypothesis density (PHD)* [12, 13]. This entity is described in Section 4.1, and is defined over the state-space for one target. It has the property that for each sub-area S in the state-space, the integral of the PHD over S is the expected number of targets within this area. Thus, peaks in the PHD can be regarded as estimated target states. Since the identities of objects are not maintained, there is no model-data association problem.

The main contribution of this paper is a particle filter [5, 7] implementation of PHD tracking, the *PHD particle filter*. The PHD particle filter implementation is described in Section 4.2.

Particle filtering (Section 3.1) is suited for tracking with non-linear and non-Gaussian motion models. Here, the PHD particle filter is applied to tracking of multiple vehicles in terrain (Section 5), a problem which is highly non-linear due to the terrain (Section 5.3). The vehicles are observed by humans situated in the terrain. Two things should be noted about this application: Since the observations originate from humans rather than automatic sensors, the degree of missing observations is much higher than the degree of spurious observations. Furthermore, the time-scale is quite long – one time-step is on the order of a few seconds. Thus, the relatively high computational complexity of particle filters compared to, e.g., Kalman filters provides less of a problem for real-time implementation than it would in many other applications. Experiments in Section 6 show the PHD particle filter to be a fast, efficient and robust alternative to tracking of the full joint distribution over targets.

2 Related work

Multi-target tracking. The problem of tracking multiple targets is more difficult than the tracking of a single target in two aspects.

If the number of targets is known and constant over time, the state-space (spanned by the indiviual state-spaces of all targets) has a known and constant dimensionality. However, if the number of targets is unknown or varies over time, the number of targets, N, is itself a (discrete) random variable, and a part of the state-space. Since the dimensionality of the state-space varies with N (e.g., two targets are described by twice as many parameters as a single target), it is not possible to compare two states of different value n of N using ordinary Bayesian statistics. One way to address this problem [3, 6] is to estimate N separately from the rest of the state-space, and then, given this, estimate the other state variables knowing the size of the state-space. Another [19] is to assume N known and constant, and model some of the targets as "hidden". A third approach [1, 8] is to do the likelihood evaluation in a space of constant dimensionality (the image space), thus avoiding the problem of comparing spaces of different dimensionality. However, the problem can also be addressed by employing *finite set statistics (FISST)* [4, 11] which is an extension of Bayesian analysis to incorporate comparisons of between state-spaces of different dimensionality. Thus, a distribution over N can be estimated with the rest of the state-space. FISST has been used extensively for tracking [11, 12, 13, 15], mainly implemented as a set of Kalman or α-β-γ-filters. The particle filter presented here is formulated within this framework.

The second problem with multi-target tracking in general is that the size of the state-space grows exponentially with the number of targets. Even with tracking algorithms that very efficiently search the state-space, it is not possible to estimate the joint distribution over a large number of targets with a limited computational effort. However, if the targets move independently, simplifications can be introduced. One approach is simply to track each target using a separate filter, e.g. [3, 16]. This simplification allows for tracking of a large number of targets, but leads to a model-data association problem, addressed by e.g. joint probabilistic data association (JPDA) [3]. To avoid this problem, Mahler and Zajic [12, 13] formulate an algorithm for propagating a combined density (PHD) over all targets, instead of modeling the probability density function (pdf) for each individual target. We present a particle filter implementation of this PHD filter.

Terrain tracking. The problem of tracking in terrain is that the motion model highly non-linear, due to the variability in the terrain. This makes linear Kalman tracking approaches like Interacting Multiple Models (IMM) [14] inappropriate, since it is there difficult to model the terrain influence in a general manner. However, in a simplified environment, such as a terrain map with only on/off road information, IMM-based approaches are successful [10]. Another type of approach is to formulate the terrain as a potential field [9, 18] or an HMM [10], where the transition probabilities correspond to terrain movability in that area. This allows for modeling of the non-linearities in the terrain. However, the potential field approach is computationally expensive [9]. Furthermore, a comparison [10] between the HMM and an IMM filter shows the IMM approach to be more efficient in a linearized situation.

We take a slightly different approach. To cope with the non-linearities of the terrain tracking problem in a mathematically principled way, we use particle filtering (also known as bootstrap filtering [5] or Condensation [7]), which has proven useful for tracking with non-linear and non-Gaussian models of motion and observations.

3 Bayesian filtering

We start by describing the formulation of the discrete-time tracking problem for a single target, with exactly one observation in each time-step.

In a Bayesian filter, the tracking problem is formulated as an iterative implementation of Bayes' theorem. All information about the state of the tracked target can be deduced from the *posterior distribution* $f_{\mathbf{X}_t \,|\, \mathbf{Z}_{1:t}}(\mathbf{x}_t \,|\, \mathbf{z}_{1:t})$ over states $\mathbf{X}_t$, conditioned on the history of observations $\mathbf{Z}_{1:t}$ from time 1 up to time t. The filter consists of two steps, prediction and observation:

Prediction. In the prediction step, the *prior distribution* $f_{\mathbf{X}_t \,|\, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{z}_{1:t-1})$ at time t is deduced from the posterior at time $t-1$ as

$$f_{\mathbf{X}_t \,|\, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{z}_{1:t-1}) = \int f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) \, f_{\mathbf{X}_{t-1} \,|\, \mathbf{Z}_{1:t-1}}(\mathbf{x}_{t-1} \,|\, \mathbf{z}_{1:t-1}) \, d\mathbf{x}_{t-1} \quad (1)$$

where the probability density function (pdf) $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$ is defined by a model of motion in its most general form.

Often, however, the state at time t is generated from the previous state according to the model

$$\mathbf{X}_t = \phi(\mathbf{X}_{t-1}, \mathbf{W}_t) \quad (2)$$

where $\mathbf{W}_t$ is a noise term independent of $\mathbf{X}_{t-1}$. This gives $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) \equiv f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$, with no dependence on the history of observations $\mathbf{z}_{1:t-1}$.

Observation. In each time-step, observations of the state are assumed generated from the model

$$\mathbf{Z}_t = h(\mathbf{X}_t, \mathbf{V}_t) \quad (3)$$

where $\mathbf{V}_t$ is a noise term independent of $\mathbf{X}_t$. From this model, the *likelihood* $f_{\mathbf{Z}_t \,|\, \mathbf{X}_t}(\mathbf{z}_t \,|\, \mathbf{x}_t)$ is derived. The posterior at time t is computed from the prior (Eq (1)) and the likelihood according to Bayes' rule:

$$f_{\mathbf{X}_t \,|\, \mathbf{Z}_{1:t}}(\mathbf{x}_t \,|\, \mathbf{z}_{1:t}) \propto f_{\mathbf{Z}_t \,|\, \mathbf{X}_t}(\mathbf{z}_t \,|\, \mathbf{x}_t) \, f_{\mathbf{X}_t \,|\, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{z}_{1:t-1}) \, . \quad (4)$$

To conclude, the posterior pdf at time t is calculated from the previous posterior at $t-1$, the motion model, and the observations at time t according to Eqs (1) and (4). The iterative filter formulation requires a known initial posterior pdf $f_{\mathbf{X}_0 \mid \mathbf{Z}_0}(\mathbf{x}_0 \mid \mathbf{z}_0) \equiv f_{\mathbf{X}_0}(\mathbf{x}_0)$.

3.1 Particle implementation

If the shape of the posterior distribution is close to Gaussian, and the functions $h(.)$ and $\phi(.)$ linear, the system can be modeled analytically in an efficient manner, e.g. as a Kalman filter. However, for non-linear models of motion and observation, the posterior distribution will have a more complex shape, often with several maxima. In these cases, a Kalman filter is no longer applicable.

Particle filtering, also known as bootstrap filtering [5] or Condensation [7], has proven to be a useful tool for Bayesian tracking with non-linear models of motion and observation. Particle filtering is a sequential Monte Carlo method. For an overview of the state of the art in applications of particle filters, see [2].

The posterior is represented by a set of $\mathcal{N}$ state hypotheses, or particles $\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{\mathcal{N}}\}$. The density of particles in a certain point in state-space represents the posterior density in that point [5, 7]. A time-step proceeds as follows:

Prediction. The particles $\{\boldsymbol{\xi}_{t-1}^1, \ldots, \boldsymbol{\xi}_{t-1}^{\mathcal{N}}\}$, representing $f_{\mathbf{X}_{t-1} \mid \mathbf{Z}_{1:t-1}}(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1})$, are propagated in time by sampling from the dynamical model $f_{\mathbf{X}_t \mid \mathbf{X}_{t-1}}(\mathbf{x}_t \mid \boldsymbol{\xi}_{t-1}^s)$ for $s = 1, \ldots, \mathcal{N}$. The propagated particles, $\{\tilde{\boldsymbol{\xi}}_t^1, \ldots, \tilde{\boldsymbol{\xi}}_t^{\mathcal{N}}\}$, represent the prior $f_{\mathbf{X}_t \mid \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \mid \mathbf{z}_{1:t-1})$ at time t.

Observation. Given the new observation $\mathbf{z}_t$ of $\mathbf{Z}_t$, each propagated particle $\tilde{\boldsymbol{\xi}}_t^s$ is assigned a weight $\pi_t^s \propto f_{\mathbf{Z}_t \mid \mathbf{X}_t}(\mathbf{z}_t \mid \tilde{\boldsymbol{\xi}}_t^s)$. The weights are thereafter normalized to sum to one.

Resampling. Now, $\mathcal{N}$ new particles are sampled from the set of particles with attached weights, $\{(\tilde{\boldsymbol{\xi}}_t^1, \pi_t^1), \ldots, (\tilde{\boldsymbol{\xi}}_t^{\mathcal{N}}, \pi_t^{\mathcal{N}})\}$. The frequency with which each particle is resampled is proportional to the weight (Monte Carlo sampling). The result is a particle set with equal weights, $\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{\mathcal{N}}\}$, representing the posterior distribution at time t.

4 FISST multi-target filtering

We now extend the single-target particle filter to comprise an unknown and varying number of targets. The set of tracked objects at time t is a random set [4, 11] $\Gamma_t = \{\mathbf{X}_t^1, \ldots, \mathbf{X}_t^{N_t^X}\}$, where $\mathbf{X}_t^i$ is the state vector of object i and N_t^X is the number of objects in the set. A certain outcome of the random set Γ_t is denoted $X_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t^X}\}$. Similarly, the set of observations received at time t is a random set $\Sigma_t = \{\mathbf{Z}_t^1, \ldots, \mathbf{Z}_t^{N_t^Z}\}$, where N_t^Z can be larger than, the same as, or smaller than N_t^X. A certain outcome of the random set Σ_t is denoted $Z_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^{n_t^Z}\}$.

Using these random set representations, the multi-target version of Eq (4) is [4, 11]

$$f_{\Gamma_t \mid \Sigma_{1:t}}(X_t \mid Z_{1:t}) \propto \\ f_{\Sigma_t \mid \Gamma_t}(Z_t \mid X_t)\, f_{\Gamma_t \mid \Sigma_{1:t-1}}(X_t \mid Z_{1:t-1}) \quad (5)$$

where $f_{\Gamma_t \mid \Sigma_{1:t}}(X_t \mid Z_{1:t})$ is a multi-target posterior belief density function, $f_{\Sigma_t \mid \Gamma_t}(Z_t \mid X_t)$ multi-target, multi-observation likelihood, and $f_{\Gamma_t \mid \Sigma_{1:t-1}}(X_t \mid Z_{1:t-1})$ a multi-target prior. These densities are defined using finite set statistics (FISST). Details on FISST can be found in [4], while a general particle formulation is presented in [17].

4.1 PHD filtering

For a large number of targets, the computational complexity of Eq (5) will be very high due to the size of the state-space (see also discussion in Section 2). However, if the signal to noise ratio (SNR) is high and the targets move independently of each other, the full posterior $f_{\Gamma_t \mid \Sigma_{1:t}}(X_t \mid Z_{1:t})$ can in each time step be approximately recovered from the first moment of this distribution, the probability hypothesis density (PHD) [13]:

$$D_{\mathbf{X}_t \mid \Sigma_{1:t}}(\mathbf{x}_t \mid Z_{1:t}) = \int f_{\Gamma_t \mid \Sigma_{1:t}}(\{\mathbf{x}_t\} \cup Y \mid Z_{1:t})\, \delta Y \quad (6)$$

which is defined over the state-space Θ of one target, instead of the much larger space $\Theta^{N_t^X}$ in which the full posterior $f_{\Gamma_t \mid \Sigma_{1:t}}(X_t \mid Z_{1:t})$ live. This means that the computational cost of propagating the PHD over time is much lower than propagating the full posterior.

The PHD has the properties that, for any subset $S \subseteq \Theta$, the integral of the PHD over S is the expected number of objects in S at time t:

$$E[|\Gamma_t \cap S|] = \int_S D_{\mathbf{X}_t \mid \Sigma_{1:t}}(\mathbf{x}_t \mid Z_{1:t})\, d\mathbf{x}_t\ . \quad (7)$$

In other words, it will have local maxima approximately at the locations of the targets. The integral of the PHD over Θ is the expected number of targets, n_t^X.

To find the target locations, a mixture of Gaussians is fitted to the PHD in each time step. A local maximum is then found as the mean of a Gaussian in the mixture.

We now describe one time-step in the PHD filter. The PHD can not be exactly maintained over time [13]; how good the approximative estimation of the PHD is depends on the SNR. In the description below, $\hat{D}$ is used to denote an approximately estimated PHD [13].

Prediction. The temporal model of the targets include birth (appearance of a target in the field of view), death (disappearance of a target from the field of view) and temporal propagation. Probability of target death is p_D and of target birth p_B. Both these probabilities are state independent.

Target hypotheses are, as in the single target case, propagated from earlier hypotheses according to the dynamical model in Eq (2), which defines the motion pdf $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$, a special case of the general motion pdf in Eq (1).

In [12], target hypotheses are assumed to be born from a uniform distribution over Θ. Here, to better explore the state-space, target hypotheses are born from observations at the previous time instant. This is possible if the observation function $h(.)$ (Eq (3)) can be inverted with respect to $\mathbf{X}_t$:[1]

$$\mathbf{X}_t = \phi(h^{-1}_{\mathbf{X}_t}(\mathbf{Z}_{t-1}, \mathbf{V}_{t-1}), \mathbf{W}_t) \ . \tag{8}$$

This model defines the birth pdf $f_{\mathbf{X}_t \,|\, \mathbf{Z}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{z}_{t-1})$ which also is a special case of the motion pdf in Eq (1).

In the multi-target case, there is a random set of observations $\Sigma_t = \{\mathbf{Z}^1_t, \ldots, \mathbf{Z}^{N^Z_t}_t\}$. To take all observations into account for target birth, a birth PHD is defined from the set of birth pdf:s as

$$D_{\mathbf{X}_t \,|\, \Sigma_{t-1}}(\mathbf{x}_t \,|\, Z_{t-1}) = \sum_{\mathbf{z}^i_{t-1} \in Z_{t-1}} f_{\mathbf{X}_t \,|\, \mathbf{Z}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{z}^i_{t-1}) \ . \tag{9}$$

Given the models of motion, death and birth, the approximate prior PHD [13] is estimated from the approximate posterior PHD at the previous time instant [12] as

$$\begin{aligned}
&\hat{D}_{\mathbf{X}_t \,|\, \Sigma_{1:t-1}}(\mathbf{x}_t \,|\, Z_{1:t-1}) = \\
&\quad p_B D_{\mathbf{X}_t \,|\, \Sigma_{t-1}}(\mathbf{x}_t \,|\, Z_{t-1}) + \\
&\quad\quad \int (1 - p_D) f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) \\
&\quad\quad\quad \hat{D}_{\mathbf{X}_{t-1} \,|\, \Sigma_{1:t-1}}(\mathbf{x}_{t-1} \,|\, Z_{1:t-1}) \, d\mathbf{x}_{t-1} \ .
\end{aligned} \tag{10}$$

Observation. We define p_{FN} as the probability that a target is *not* observed at a given time step (the probability of false negative). Assuming that there are no spurious observations (a good approximation in our application where the observations originate from human observers, see Section 5.1), the approximate posterior PHD distribution is computed [12] as

$$\begin{aligned}
&\hat{D}_{\mathbf{X}_t \,|\, \Sigma_{1:t}}(\mathbf{x}_t \,|\, Z_{1:t}) \approx \\
&\quad \sum_{\mathbf{z}^i_t \in Z_t} f_{\mathbf{X}_t \,|\, \mathbf{Z}_t, \Sigma_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{z}^i_t, Z_{1:t-1}) + \\
&\quad\quad p_{FN} \hat{D}_{\mathbf{X}_t \,|\, \Sigma_{1:t-1}}(\mathbf{x}_t \,|\, Z_{1:t-1})
\end{aligned} \tag{11}$$

where

$$\begin{aligned}
&f_{\mathbf{X}_t \,|\, \mathbf{Z}_t, \Sigma_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{z}^i_t, Z_{1:t-1}) \propto \\
&\quad f_{\mathbf{Z}_t \,|\, \mathbf{X}_t}(\mathbf{z}^i_t \,|\, \mathbf{x}_t) \, \hat{D}_{\mathbf{X}_t \,|\, \Sigma_{1:t-1}}(\mathbf{x}_t \,|\, Z_{1:t-1}) \ ,
\end{aligned} \tag{12}$$

which is a pdf (with the integral 1 over the state-space).[2]

Using Eqs (9), (10) and (11), the PHD can be propagated in time. The result of the tracking is the estimated number of targets, and the location of the detected maxima in the posterior approximate PHD in each time step.

4.2 Particle implementation

We will now describe the particle filter implementation of Eqs (9), (10) and (11). The presentation follows that of the ordinary particle filter (Section 3.1) to enable comparison.

A pdf (with integral 1) is usually represented with $\mathcal{N}$ particles (Section 3.1). Here, a PHD (with integral n^X_t) is represented with $n^X_t \mathcal{N}$ particles, n^X_t being the expected number of targets at time t. One time-step proceeds as follows:

Prediction. The posterior PHD at time $t-1$ is represented by a set of (unweighted) particles $\{\boldsymbol{\xi}^1_{t-1}, \ldots, \boldsymbol{\xi}^{n^X_{t-1}\mathcal{N}}_{t-1}\}$. These are propagated in time by sampling from the dynamical model $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t \,|\, \boldsymbol{\xi}^s_{t-1})$ for $s = 1, \ldots, n^X_{t-1}\mathcal{N}$. The propagated particles are each given a weight $\varpi^s_t = (1 - p_D)/\mathcal{N}$. The set of weighted propagated particles represent the second term in Eq (10).

Now, for each of the observations $\mathbf{z}^i_{t-1}, \in Z_{t-1}, i = 1, \ldots, n^Z_{t-1}$, $\mathcal{N}$ particles are sampled from the birth model $f_{\mathbf{X}_t \,|\, \mathbf{Z}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{z}^i_{t-1})$ (Eq (9)). Each particle is given a weight $\varpi^s_t = p_B/\mathcal{N}$. The resulting set of weighted particles represent the first term in Eq (10).

The two weighted particle clouds are concatenated to form a set of particles with attached weights, $\{(\tilde{\boldsymbol{\xi}}^1_t, \varpi^1_t), \ldots, (\tilde{\boldsymbol{\xi}}^{(n^Z_{t-1}+n^X_{t-1})\mathcal{N}}_t, \varpi^{(n^Z_{t-1}+n^X_{t-1})\mathcal{N}}_t)\}$, that represent the approximate prior PHD (Eq (10)) at time t.

Observation. For each new observation $\mathbf{z}^i_t \in Z_t, i = 1, \ldots, n^Z_t$, a copy i of the prior particle set is made. New weights $\pi^{i,s}_t \propto \varpi^s_t f_{\mathbf{Z}_t \,|\, \mathbf{X}_t}(\mathbf{z}^i_t \,|\, \tilde{\boldsymbol{\xi}}^s_t)$ are computed. For each set i, the weights are thereafter normalized to sum to one. The re-weighted particle set represents the i:th term $f_{\mathbf{X}_t \,|\, \mathbf{Z}_t, \Sigma_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{z}^i_t, Z_{1:t-1})$ in the sum in Eq (11).

The original prior particle set is down-weighted according to $\pi^{0,s}_t = p_{FN}\varpi^s_t$. This set now represent the last term in Eq (11).

The concatenation of these sets, $\{(\tilde{\boldsymbol{\xi}}^1_t, \pi^1_t), \ldots, (\tilde{\boldsymbol{\xi}}^{(n^Z_t+1)(n^Z_{t-1}+n^X_{t-1})\mathcal{N}}_t, \pi^{(n^Z_t+1)(n^Z_{t-1}+n^X_{t-1})\mathcal{N}}_t)\}$, is a weighted representation of the posterior PHD.

Resampling. An unweighted representation of the posterior PHD is now obtained by resampling the weighted particle set. The expected number of targets is computed as the sum over all weights in this set:

[1] In general, $h^{-1}_{\mathbf{X}_t}(.)$ exists for sensors for which the observation space Θ_o is the same as the state space Θ. Negative examples, for which $h^{-1}_{\mathbf{X}_t}(.)$ is often impossible to obtain, are image sensors.

[2] Eq (11) was wrongly derived in [13]. However, the error was pointed out and corrected in [12].

$n_t^X = \sum_{i=1}^{(n_t^Z+1)(n_{t-1}^Z+n_{t-1}^X)} \pi_t^i$. Now, $n_t^X \mathcal{N}$ new particles are Monte Carlo sampled (Section 3.1) from the weighted set. The result is an unweighted particle set $\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{n_t^X \mathcal{N}}\}$ that represents the approximate posterior PHD $\hat{D}_{\mathbf{X}_t \mid \Sigma_{1:t}}(\mathbf{x}_t \mid Z_{1:t})$ at time t.

5 Terrain application

The PHD particle filter is here applied to terrain tracking. The reason to use particle filtering for terrain tracking is clarified in Section 5.3 – the motion model of the vehicles is non-linear and dependent on the terrain. Using particle filtering, we avoid the need to construct an analytical model of the motion noise, since the particles provide a sampled representation of the motion distribution.

5.1 Scenario

The scenario is 841 s long, simulated in time-steps of five s. Three vehicles (of the same type) travel along roads in the terrain, with a normally distributed speed of mean 8.3 m/s and standard deviation 0.1 m/s. At one time, one of the vehicles travel around 500 m off-road over a field.

The terrain is represented by a discrete map m over position. A pixel in m can take any value $T = \{\mathit{road}, \mathit{field}, \mathit{forest}\}$ (exemplified in the tracking movies (Section 6) where light grey indicates road, white field, and grey forest). The probability $p_T(t)$ that a vehicle would select terrain of type t to travel in is defined to be $p_T(\mathit{road}) = 0.66, p_T(\mathit{field}) = 0.33, p_T(\mathit{forest}) = 0.01$.

At each time-step, each vehicle is observed by a human in the terrain with probability 0.9, 0.5 or 0.1. This means that $p_{FN} = 0.1$ in the first case, $p_{FN} = 0.5$ in the second, and $p_{FN} = 0.9$ in the third. For each observation, the observer generates a report of the observed vehicle position, speed and direction, which is a noisy version of the real state, and of the uncertainty with which the observation was made, expressed as standard deviation, here $\boldsymbol{\sigma}_R = [50, 50, 1, \pi/8]$ (m, m, m/s, rad).

5.2 State-space

The state vector for a vehicle is $\mathbf{x}_t = [\mathbf{p}_t, s_t, v_t]$ where $\mathbf{p}_t$ is position (m), s_t speed (m/s) and v_t angle (rad). The random set of vehicles is in every time-step limited according to $N_t^X \leq 5$ vehicles for computational reasons.

5.3 Motion model

The motion model of the vehicles is

$$\mathbf{X}_t = \mathbf{X}_{t-1} + d\mathbf{X}_{t-1} + \mathbf{W}_t \qquad (13)$$

where $d\mathbf{X}_{t-1}$ is the movement estimated from the speed and direction in $\mathbf{X}_{t-1}$. The noise term is sampled from a distribution which is the product of a normal distribution with standard deviation $\boldsymbol{\sigma}_W = [10, 10, 2, \pi/4]$, and of a terrain distribution. The terrain distribution depends on probabilities of finding a vehicle in different types of terrain. The sampling from this product distribution is implemented as follows: Sample particles $\boldsymbol{\xi}^i$ using the normally distributed noise term. Each particle i now obtains a value $\pi^i = p_T(m(\boldsymbol{\xi}^i))$. Resample the particles according to π^i using Monte Carlo sampling.

5.4 Birth model

We assume the birth rate p_B and death rate p_D of targets to be invariant to position and time-step, and only dependent on the probability of missing observations p_{FN}. The goal of the tracking is most often to keep track of all targets while not significantly overestimating the number of targets. We design the birth and death model for this purpose. A high degree of missing observations should give a higher birth rate since it takes more time steps in general to "confirm" a birth with a new observation. The mean number of steps between observations is $\frac{1}{1-p_{FN}}$. Therefore,

$$p_B = K^{1-p_{FN}} \ , \qquad (14)$$

$$p_D = K \ . \qquad (15)$$

The constant K is set empirically to 0.01.

5.5 Observation model

As mentioned in Section 5.1, observations $\mathbf{Z}_t$ are given in the target state-space, which means that Eq (3) becomes

$$\mathbf{Z}_t = \mathbf{X}_t + \mathbf{V}_t \ . \qquad (16)$$

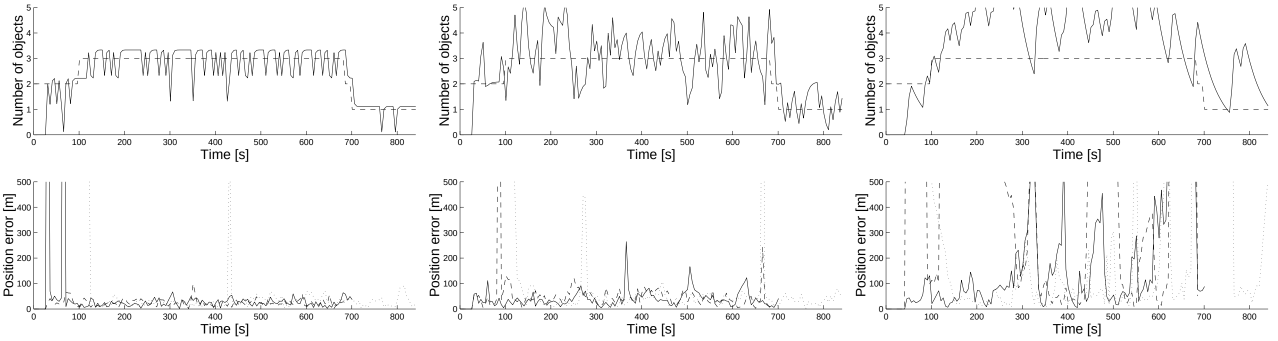
The observation noise $\mathbf{V}_t$ is normally distributed with standard deviation $\boldsymbol{\sigma}_V = \boldsymbol{\sigma}_R$ (Section 5.1).

6 Results

Using the settings described above experiments were performed to test the performance of the PHD particle filter (Figure 1) and to compare it with a particle implementation [17] of the FISST filter [11], which maintains the joint distribution over the full random set over time (Figure 2). $\mathcal{N} = 1000$ particles were used to represent a pdf in the PHD filter. The settings of the FISST particle filter simulation can be found in [17].

The tracking performance was measured in two ways, comparing the estimated number of targets with the true value, and measuring the Euclidean distance between the ground truth target positions and the local maxima in the PHD (Section 4). [3]

[3] Movies of the six tracking examples can be found at `http://www.foi.se/fusion/mpg/FUSION03/`. Two movies relating to each of the Figures 1a, 1b, 1c, 2a, 2b, and 2c can be found. For, e.g., Figure 1a, the movie `phdFigure1(a).mpg` shows the (discretized) PHD (blue – 0, red – 0.2) with white 95% error ellipses indicating the Gaussians fitted to the PHD. The movie `terrainFigure1(a).mpg` shows the terrain (grey-scale, Section 5.1), the particles (red) and Gaussians (deep blue for high PHD peaks, lighter for lower peaks). True vehicle positions are indicated by green +, observations by green *.

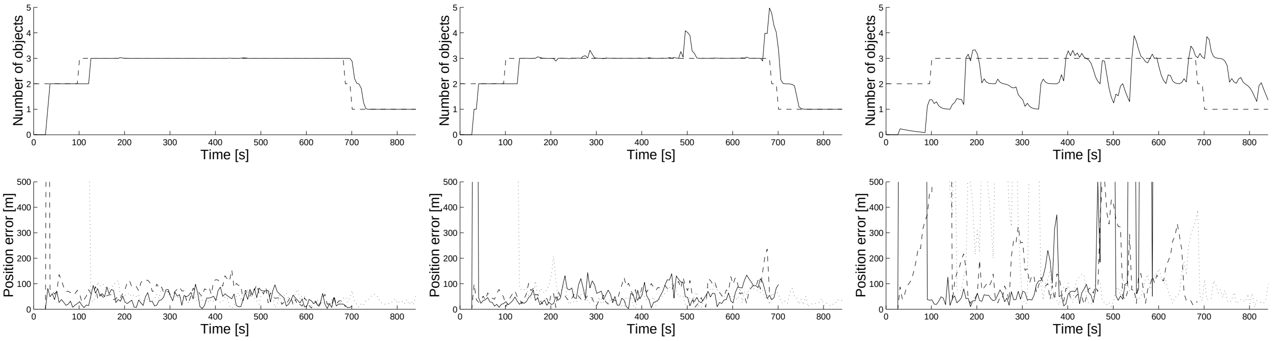


(a) $p_{FN} = 0.1$

(b) $p_{FN} = 0.5$

(c) $p_{FN} = 0.9$

Figure 1: Tracking errors for the PHD filter. (a) Observation probability 0.9. (b) Observation probability 0.5. (c) Observation probability 0.1. The upper graph in each subfigure shows estimated (solid line) number of targets, compared to the true (dashed line) number. The lower graph shows position errors for the three vehicles. Solid, dashed and dotted lines denote different vehicles. The dotted target appears after 101 s, the dashed target disappears after 687 s and the solid target after 702 s. Position error is measured as the Euclidean distance from the true target position to the nearest detected maxima in the estimated PHD.



(a) $p_{FN} = 0.1$

(b) $p_{FN} = 0.5$

(c) $p_{FN} = 0.9$

Figure 2: Tracking errors for the full random set filter, on the same scenario. (a) Observation probability 0.9. (b) Observation probability 0.5. (c) Observation probability 0.1. A complete description of this simulation can be found in [17].

Both filters were implemented in Matlab, which is a language not suited for real-time applications. However, it should be noted that both algorithms required less or marginally more time than the span of a time-step in the simulation, 5 s, running in Linux on an ordinary desktop computer. This indicates the usability of both algorithms for real-time applications.

One iteration in the FISST particle filter required 4.9 s on average, while an iteration in the PHD particle filter required 0.38 s. The generation of the (discretized) PHD and the fitting of the mixture of Gaussians to the PHD were identical in the two filters, and required 1.2 s on average. Thus one time-step in the full FISST particle filter takes approximately 12 times longer than the corresponding iteration in the PHD filter. This should be kept in mind while comparing the performance of the two filters.

As expected, the FISST particle filter outperforms the PHD particle filter in estimating the number of targets (upper graph in each subfigure) for all tested values of p_{FN}. If this is an important aspect of the tracking, a filter maintaining belief over the full random set should be used.

However, the accuracy in position estimation is very similar between the two filters. With high or moderate observation probability (Figures 1a,b and 2a,b), both filters maintain track of all targets, save for a few mistakes in the PHD filter that are quickly recovered from. With a low observation probability, both filters (Figure 1c and 2c) fail to track the targets to a high degree. The reasons for that is simply that the SNR is too low [13, 17].

To conclude, the PHD particle filter's accuracy in estimating the number of targets is low, and falls quickly with the SNR. However, the positions of the targets are estimated with the same accuracy as provided by a filter representing the full random set.

Thus, the PHD particle filter is a robust and computationally inexpensive alternative to representing the full joint distribution over the random set, when estimation of the number of targets is not the primary issue.

7 Conclusions

The contribution of this paper has been a particle filtering implementation of the PHD filter presented by Mahler and Zajic [12, 13]. The PHD particle filter was applied to tracking of an unknown and changing number of vehicles in terrain, a problem incorporating highly non-linear motion, due to the terrain.

Experiments showed the PHD particle filter to be a fast and robust alternative to a filter where the full joint distribution over the set of targets was maintained over time.

7.1 Future work

This work could be extended along several avenues of research. Firstly, the effects of all parameter settings on the tracking need to be investigated. In the experiments in Section 6, only the degree of missing observations, p_{FN}, was varied.

Furthermore, it would be interesting to investigate more sophisticated observation models. The experiments here show clearly that the performance of the filter is strongly dependent on the SNR. One way to heighten the SNR with our type of sensors, human observers, is to take negative information (i.e. absence of reports) into regard. This is possible if the fields of view of the observers are known.

Finally, a real-time implementation should be made, and the filter should be tested over longer time periods with more targets. A larger testbed is currently developed for this purpose.

Acknowledgments. This work was financed by FOI project E7037. The author wishes to thank Ronald Mahler, Pontus Svenson and Sven-Lennart Wirkander for helpful comments.

References

[1] D. J. Ballantyne, H. Y. Chan, and M. A. Kouritzin. A branching particle-based nonlinear filter for multi-target tracking. In *International Conference on Information Fusion*, volume 1, pages WeA2:3–10, 2001.

[2] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, NY, USA, 2001.

[3] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8(3):173–184, 1983.

[4] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1997.

[5] N. Gordon, D. Salmond, and A. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar, Sonar and Navigation*, 140(2):107–113, 1993.

[6] C. Hue, J-P. Le Cadre, and P. Pérez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50(2):309–325, 2002.

[7] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *IEEE International Conference on Computer Vision, ICCV*, volume 2, pages 34–41, 2001.

[9] K. Kastella, C. Kreucher, and M. A. Pagels. Nonlinear filtering for ground target applications. In *SPIE Conference on Signal and Data Processing of Small Targets*, volume 4048, pages 266–276, 2000.

[10] C-C. Ke, J. G. Herrero, and J. Llinas. Comparative analysis of alternative ground target tracking techniques. In *International Conference on Information Fusion*, volume 2, pages WeB5:3–10, 2000.

[11] R. Mahler. *An Introduction to Multisource-Multitarget Statistics and its Applications*. Lockheed Martin Technical Monograph, 2000.

[12] R. Mahler. An extended first-order Bayes filter for force aggregation. In *SPIE Conference on Signal and Data Processing of Small Targets*, volume 4729, 2002.

[13] R. Mahler and T. Zajic. Multitarget filtering using a multitarget first-order moment statistic. In *SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition*, volume 4380, pages 184–195, 2001.

[14] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. IMM methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.

[15] S. Musick, K. Kastella, and R. Mahler. A practical implementation of joint multitarget probabilities. In *SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition*, volume 3374, pages 26–37, 1998.

[16] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24(6):843–854, 1979.

[17] H. Sidenbladh and S-L. Wirkander. Tracking random sets of vehicles in terrain. In *IEEE Workshop on Multi-Object Tracking*, 2003.

[18] E. P. Sodtke and J. Llinas. Terrain based tracking using position sensors. In *International Conference on Information Fusion*, volume 2, pages ThB1:27–32, 2001.

[19] L. D. Stone. A Bayesian approach to multiple-target tracking. In D. L. Hall and J. Llinas, editors, *Handbook of Multisensor Data Fusion*, 2002.