Appears in *IEEE Workshop on Multi-Object Tracking*, Madison, WI, USA 2003.

Tracking Random Sets of Vehicles in Terrain

Hedvig Sidenbladh

Dept. Data and Information Fusion
Div. Command and Control Systems
Swedish Defence Research Agency
SE-172 90 Stockholm, Sweden
hedvig@foi.se

Sven-Lennart Wirkander

Dept. Autonomous Systems
Div. Systems Technology
Swedish Defence Research Agency
SE-172 90 Stockholm, Sweden
svenl@foi.se

Abstract

This paper presents a particle filtering formulation for tracking an unknown and varying number of vehicles in terrain. The vehicles are modeled as a random set, *i.e. a set of random variables, for which the cardinality is itself a random variable. The particle filter formulation is here extended according to* finite set statistics (FISST) *which is an extension of Bayesian theory to define operations on random sets. The filter was successfully tested on a simulated scenario with three vehicles moving in terrain, observed by humans in the terrain.*

1. Introduction

Tracking of multiple objects simultaneously over time is an important problem in many signal processing, computer vision, and information fusion applications. The objects could either be moving freely, or be dependent on or connected to each other.

The application considered here is tracking of multiple vehicles in terrain. The tracking is guided by reports of vehicle sightings from human observers on the ground, and information about the terrain from a map database. No data is available from sensors on the vehicles. The vehicles move independently of each other. However, due to the terrain, the motion is highly non-linear. Furthermore, the multi-object observation model becomes non-linear due to object-data association abiguities. Due to the non-linearities of the system, particle filters [8, 11] are used to track the vehicles. Particle filtering is presented in Section 3.

The difference between tracking a single object and a known number, n, of objects is the object-data association problem [2, 6], resulting in non-linear observation models. Using these observation models, Bayesian methods like particle filtering can be used to estimate the distribution over the concatenated state-space $[\mathbf{x}_t^1, \mathbf{x}_t^2, ..., \mathbf{x}_t^n]$ of all objects.

A new problem arises when the number of objects to track is unknown or varies over time. The number of objects, N, is then a discrete random state variable. The state-space has different dimensionality for different values n of N. This introduces difficulties, since one needs to formalize a measure for comparing state-spaces with different dimensionality. The path we take to address this problem, is to the set of objects to track as a *random set* [7, 15], i.e. a set of random variables, for which the cardinality is itself a random variable. Statistical operations on random sets are formulated in *finite set statistics (FISST)* [7, 15], which is an extension of Bayesian formalism to incorporate comparisons between state-spaces of different dimensionality. This enables us to treat N as a random variable with a discrete probability distribution and to estimate the distribution over N jointly together with the distribution over the rest of the state-space. The FISST formulation of the tracking problem is presented in [15], and the theory in [7].

The contribution of this paper is a general particle filter for random sets, the *FISST particle filter*, designed for tracking an unknown number of multiple vehicles with terrain motion models. The filter is a direct general particle filter implementation of the FISST multi-object filter equations described in [15], with two extensions:

- The formulation of the object birth model (Eq (6)).[1] Here, objects are born from observations at the previous time step, as opposed to uniformly generated objects [15]. This allows for a more efficient exploration of the state-space.
- A motion model for terrain tracking (Eq (16)), taking into account type of terrain. The terrain model is described in Section 6.1.

Results (Section 6.6) from a simulated scenario of three vehicles show that the particle filter is successful up to certain noise limits. Since the time scale of terrain tracking is

[1] It should be noted that a birth model allows for implicit initialization of the tracking. Thus, the algorithm will recover from a temporary "lost track" at the next observation of that vehicle.

quite large (compared to e.g. air target tracking) the relatively large complexity of the particle filter poses no great problems for a real-time implementation.

2. Related work

The problem of tracking in terrain is that the motion model is highly non-linear, due to the variability in the terrain. This makes linear Kalman tracking approaches like Interacting Multiple Models (IMM) [17] 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 [14]. Other type of approaches are HMM formulations [14] or potential fields [13, 23] which are computationally heavy [13].

We take a 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 [8] or Condensation [11]), which has proven useful [4] for tracking with non-linear and non-Gaussian models of motion and observations.

In many applications of multi-object tracking, the number of objects to track, n, is known beforehand. Compared to single object tracking, the main issue in tracking a known number of objects is the object-data association problem. A number of approaches adressing this problem have been presented, e.g. PDA [2] and JPDA [6]. The task of tracking an unknown and changing number N of objects is, from a modeling perspective, an even more challenging one. In principle, the number of objects is then treated as a discrete state variable. However, since each object has its own vector of state variables, the dimensionality of the combined state-space changes with the number of objects.

One approach to address this problem [6, 10] is to estimate N separately from the rest of the state-space, and then, given this, estimate the other state variables. This corresponds to approximating the marginal distribution over N with a delta function – a simplification of reality that can lead to errors in the tracking. Alternatively, the model of Stone [24] consists of a constant (large) number of objects, only some of whom are visible in each time-step. In this manner, the problem with changing dimensionality is avoided. However, the method has been criticized [15] for a vague mathematical foundation.

The particle filters of Isard and MacCormick [12] and Ballantyne et al. [1] maintained distributions for all values of N, just as in our case. However, the problem of comparing states of different dimensionality was there elegantly eliminated since the observations were made in the form of images, and the likelihood was formulated in the image space, with constant dimensionality. Since our type of sensor is different, we do not have this possibility.

The mathematical foundation for multi-object tracking used in this paper, FISST, enables the modeling of a state-space with changing dimensionality. Thus, a distribution over N can be estimated with the rest of the state-space, without the need for special cases or simplifications. FISST has been used extensively for tracking [15, 16, 19] but no particle filter implementation has yet been presented.

Other approaches to tracking a changing number of objects include jump Markov systems (JMS) and IPDA/JIPDA. JMS [5, 9, 18] are Monte Carlo methods in which the state space is switched according to a Markov chain. The relationship between JMS and FISST needs to be investigated in the future [5]. IPDA [21] and JIPDA [20] are extensions of PDA and JPDA dealing with tracking of a changing number of objects. Challa showed recently [3] that these methods can be formulated within the FISST framework under some assumptions of linearity.

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 $f_{\mathbf{X}_0 \,|\, \mathbf{Z}_0}(\mathbf{x}_0 \,|\, \mathbf{z}_0) \equiv f_{\mathbf{X}_0}(\mathbf{x}_0)$.

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, if the models of motion and observation are non-linear or highly distorted by noise, the posterior distribution will have a more complex shape. In these cases, a Kalman filter is no longer applicable.

Particle filtering, also known as bootstrap filtering [8] or Condensation [11], has proven to be a useful tool for Bayesian tracking with non-linear models of motion and observation. For an overview of the state of the art in applications of particle filters, see [4].

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 [8, 11]. A time-step proceeds as follows:

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

As discussed in the introduction, Bayesian equations for tracking a single object does not readily extend to cases where the number of objects is unknown or varying. Here, we employ random set formalism to account for this. The set of objects to be tracked can be considered a random set [7, 15]. A random finite set is a finite set whose elements, as well as their number, are random.

The mathematical framework for handling random sets in a probabilistic manner is called finite-set statistics (FISST) [7, 15]. FISST is a generalization of Bayesian theory to describe statistical properties of random sets.

FISST has been used (e.g. [15]) for formulating a framework for tracking of an unknown number of objects from a set of observations. Here, it is reformulated to encompass a motion model dependent on the observations in the previous time-step, as shown in Eq (1).

The set of tracked objects at time t is a random set $\Gamma_t = \{\mathbf{X}_t^1, \ldots, \mathbf{X}_t^{N_t^X}\}$, 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 $Z_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^{n_t^Z}\}$.

The FISST equivalent of a pdf $f_{\mathbf{Y}_t}(\mathbf{y}_t)$ for a random vector $\mathbf{Y}_t$ is a *multi-object probability density function* $f_{\Upsilon_t}(Y_t)$ where $\Upsilon_t = \{\mathbf{Y}_t^1, \mathbf{Y}_t^2, \ldots, \mathbf{Y}_t^{N_t^Y}\}$ is a random set [7, 15]. The goal of the FISST multi-object filter is to maintain the posterior function $f_{\Gamma_t \,|\, \Sigma_{1:t}}(X_t \,|\, Z_{1:t})$ over Γ_t conditioned on the history of sets of observations up to t, $\Sigma_{1:t}$.

One time-step in the filter will now be described. The organization follows that of Section 3 to enable comparison.

Prediction. The prior distribution over the random set Γ_t can be expanded [15] as

$$f_{\Gamma_t \,|\, \Sigma_{1:t-1}}(X_t \,|\, Z_{1:t-1}) = \sum_{n_{t-1}^X=0}^{\infty} \frac{1}{n_{t-1}^X!} \int f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{1:t-1}}(X_t \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{1:t-1})\; f_{\Gamma_{t-1} \,|\, \Sigma_{1:t-1}}(\{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\} \,|\, Z_{1:t-1})\; d\mathbf{x}_{t-1}^1 \ldots d\mathbf{x}_{t-1}^{n_{t-1}^X} \quad (5)$$

where n_{t-1}^X denotes a certain value of the number of objects at time $t-1$.

The motion model is used to generate the density $f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{1:t-1}}(X_t \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{1:t-1})$ for each value n_{t-1}^X. The motion model for a single object is defined as in Eq (2), i.e. with no dependence on the observations, only on the state at $t-1$. If the number and identities of objects were unchanged over time, the motion model for a random set of objects, moving independently, would be $\Gamma_t = \{\phi(\mathbf{X}_{t-1}^1, \mathbf{W}_t^1), \ldots, \phi(\mathbf{X}_{t-1}^{N_{t-1}^X}, \mathbf{W}_t^{N_{t-1}^X})\}$ where

$\mathbf{W}_t^1, \ldots, \mathbf{W}_t^{N_{t-1}^X}$ are i.i.d. random vectors. However, both birth and death of objects can occur during a step in time. The probability of death is p_D, of birth p_B.

Objects are born according to a birth model. If the inverse of the observation function $h(.)$ with respect to $\mathbf{X}_t$, $h_{\mathbf{X}_t}^{-1}(.)$, in Eq (3) exists, it is possible to generate an object from an old observation $\mathbf{Z}_{t-1}$:

$$\mathbf{X}_t = \phi(h_{\mathbf{X}_t}^{-1}(\mathbf{Z}_{t-1}, \mathbf{V}_{t-1}), \mathbf{W}_t) \quad (6)$$

where $\mathbf{V}_{t-1}$ is the observation noise at $t-1$. For our type of sensor (human observers) this is possible.[2] This enables us to explore the state-space more efficiently. The birth model defines the pdf $f_{\mathbf{X}_t \mid \mathbf{Z}_{t-1}}(\mathbf{x}_t \mid \mathbf{z}_{t-1})$ which also is a special case of the propagation pdf $f_{\mathbf{X}_t \mid \mathbf{X}_{t-1}, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$ (Eq (1)).

Using this model of birth, death and motion, the motion distribution in Eq (5) can be expressed as a combination of the motion models of the individual objects in the set. For a specified cardinality n_{t-1}^X of Γ_{t-1}, the motion distribution is, for each cardinality level n_t^X, equal to

$$f_{\Gamma_t \mid \Gamma_{t-1}, \Sigma_{t-1}}(X_t \mid \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{t-1}) = \sum_{k=\max(0, n_t^X - n_{t-1}^X)}^{\min(n_{t-1}^Z, n_t^X)} P_m(k) \binom{n_t^X}{k} \sum_{l_1 \neq \cdots \neq l_k \in [1, n_{t-1}^Z]} \; \sum_{l_{k+1} \neq \cdots \neq l_{n_t^X} \in [1, n_{t-1}^X]} F_m(\mathbf{l}) \,, \quad (7)$$

$$P_m(k) = (p_B)^k (1-p_B)^{n_{t-1}^Z - k} (p_D)^{n_{t-1}^X - (n_t^X - k)} (1 - p_D)^{n_t^X - k} \,, \quad (8)$$

$$F_m(\mathbf{l}) = f_{\mathbf{X}_t \mid \mathbf{Z}_{t-1}}(\mathbf{x}_t^1 \mid \mathbf{z}_{t-1}^{l_1}) \ldots f_{\mathbf{X}_t \mid \mathbf{Z}_{t-1}}(\mathbf{x}_t^k \mid \mathbf{z}_{t-1}^{l_k}) f_{\mathbf{X}_t \mid \mathbf{X}_{t-1}}(\mathbf{x}_t^{k+1} \mid \mathbf{x}_{t-1}^{l_{k+1}}) \ldots f_{\mathbf{X}_t \mid \mathbf{X}_{t-1}}(\mathbf{x}_t^{n_t^X} \mid \mathbf{x}_{t-1}^{l_{n_t^X}}) \,. \quad (9)$$

$P_m(k)$ is the transition probability of objects from level n_{t-1}^X at time $t-1$ to level n_t^X at t, given that k objects are born from observations. $F_m(\mathbf{l})$ is the joint distribution over objects, conditioned on a certain configuration $\mathbf{l}$ of previous objects and observations.[3]

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

[3] The parameter k is the number of objects generated from observations, while $n_t^X - k$ is the number of objects propagated from the previous objects in X_{t-1}. The binomial term $\binom{n_t^X}{k}$, the number of ways k elements can be drawn from a set of n_t^X, compensates for the fact that only the first k elements of X_t are considered drawn from observations, not any k elements. The notation $l_1 \neq \cdots \neq l_k \in [1, n_{t-1}^X]$ means k different integers (indices) $\mathbf{l} = [l_1, \ldots, l_k]$ drawn from the interval $[1, n_{t-1}^X]$. Taking the sum over $\mathbf{l}$ means taking the sum over all possible combinations of k different integers between 1 and n_{t-1}^X.

Observation. The FISST equivalent [15] of Eq (4) is

$$f_{\Gamma_t \mid \Sigma_{1:t}}(X_t \mid Z_{1:t}) \propto f_{\Sigma_t \mid \Gamma_t}(Z_t \mid X_t)\, f_{\Gamma_t \mid \Sigma_{1:t-1}}(X_t \mid Z_{1:t-1}) \quad (10)$$

where $f_{\Sigma_t \mid \Gamma_t}(Z_t \mid X_t)$ is a likelihood function. An observation is, as in the single object case in Eq (3), generated from exactly one object.

At a certain time-step, there can be both missing observations – with probability p_{FN} – and clutter (i.e. spurious observations that do not correspond to real objects) – with probability p_{FP}. Clutter is generated according to a uniform function over the observation space Θ_o, $\mathbf{Z}_t \sim \mathrm{U}(\Theta_o)$, which gives the uniform observation density $f_{\mathbf{Z}_t}$. Each of the n^S sensors (here, humans) send zero or one observation. This definition yields the likelihood density

$$f_{\Sigma_t \mid \Gamma_t}(Z_t \mid X_t) = \sum_{k=0}^{\min(n_t^Z, n_t^X)} P_o(k) (f_{\mathbf{Z}_t})^{n_t^Z - k} \sum_{l_1 \neq \cdots \neq l_k \in [1, n_{t-1}^Z]} \; \sum_{m_1 \neq \cdots \neq m_k \in [1, n_t^X]} F_o(\mathbf{l}, \mathbf{m}) \,, \quad (11)$$

$$P_o(k) = (p_{FN})^{n_t^X - k} (1 - p_{FN})^k (1 - p_{FP})^{n^S - (n_t^Z - k)} (p_{FP})^{n_t^Z - k} \,, \quad (12)$$

$$F_o(\mathbf{l}, \mathbf{m}) = f_{\mathbf{Z}_t \mid \mathbf{X}_t}(\mathbf{z}_t^{l_1} \mid \mathbf{x}_t^{m_1}) \ldots f_{\mathbf{Z}_t \mid \mathbf{X}_t}(\mathbf{z}_t^{l_k} \mid \mathbf{x}_t^{m_k}) \,. \quad (13)$$

$P_o(k)$ is the probability that k observations originate from real objects. $F_o(\mathbf{l}, \mathbf{m})$ is the joint likelihood of a certain configuration $\mathbf{l}$ of observations originating from a certain configuration $\mathbf{m}$ of objects.

Using Eqs (5)–(13), the posterior function over Γ_t can be propagated in time.

4.1. Particle implementation

We will now describe the particle filter implementation of Eqs (5)–(13).

For each level n_t^X in the random set representation, the marginal probability over the number of objects

$$p_{N_t^X \mid \Sigma_{1:t}}(n_t^X \mid \Sigma_{1:t}) = \frac{1}{n_t^X!} \int f_{\Gamma_t \mid \Sigma_{1:t}}(\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n^X}\} \mid Z_{1:t}) d\mathbf{x}_t^1 \ldots d\mathbf{x}_t^{n^X} \quad (14)$$

is maintained. Furthermore, each level $n_t^X > 0$ is represented by a separate set of particles. The particle set at level n_t^X thus represent the joint pdf over the state-space of all objects given that the number of objects is known to be n_t^X. If the particle filter at level 1 has dimensionality d, then the filter at level 2 has dimensionality $2d$ and so on.

Consider the filter for object level n_t^X. The posterior function $\frac{1}{n_t^X!} f_{\Gamma_t \mid \Sigma_{1:t}}(\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t^X}\} \mid Z_{1:t})$ at time t is represented by the marginal probability for n_t^X objects, $\Pi_t^{n_t^X} =$

$p_{N_t^X \mid \Sigma_{1:t}}(n_t^X \mid Z_{1:t})$ (see Eq (14)), and a set of $\mathcal{N}$ composite particles $\{[\boldsymbol{\xi}_t^{1,1}, \ldots, \boldsymbol{\xi}_t^{1,n_t^X}],\ \ldots, [\boldsymbol{\xi}_t^{\mathcal{N},1}, \ldots, \boldsymbol{\xi}_t^{\mathcal{N},n_t^X}]\}$. Each particle $\boldsymbol{\xi}_t^{s,i}$ corresponds to hypothesis s about the state of object i at level n_t^X. One time-step proceeds as:

Prediction. For each level n_t^X at time t, the transition probabilities $P_m(k)$ (Eq (8)) from old levels n_{t-1}^X are computed. These probabilities depend on the probability of death and birth and on the number n_{t-1}^Z of observations that were received at $t-1$.

For each new level, particles are sampled from all old levels with transition probability > 0, and propagated in time by sampling from the motion distributions $F_m(\mathbf{l})$ (Eq (9)). For each term $F_m(\mathbf{l})$ in the sum in Eq (8), $\mathcal{N}$ single-object particles (of dimensionality d) are sampled from each pdf in $F_m(\mathbf{l})$. The particles are then concatenated to form composite particles (of dimensionality $n_t^X d$). Each composite particle s is given a weight $\varpi_t^{s,n_t^X} = P_m(k)\Pi_{t-1}^{n_{t-1}^X}$. The propagated particles with attached weights, $\{([\tilde{\boldsymbol{\xi}}_t^{1,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{1,n_t^X}], \varpi_t^{1,n_t^X}), \ldots, ([\tilde{\boldsymbol{\xi}}_t^{a\mathcal{N},1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{a\mathcal{N},n_t^X}], \varpi_t^{a\mathcal{N},n_t^X})\}$, where a is a positive integer[4], represent the prior distribution $f_{\Gamma_t \mid \Sigma_{1:t-1}}(\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t^X}\} \mid Z_{1:t-1})$.

Observation. All composite particles $([\tilde{\boldsymbol{\xi}}_t^{s,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{s,n_t^X}], \varpi_t^{s,n_t^X})$ are now assigned new weights $\pi_t^{s,n_t^X} = \varpi_t^{s,n_t^X} f_{\Sigma_t \mid \Gamma_t}(Z_t \mid [\tilde{\boldsymbol{\xi}}_t^{s,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{s,n_t^X}])$ according to their likelihood (Eq (11)).

For each level n_t^X, the probability $\Pi_t^{n_t^X}$ is computed as the sum over all particle weights on that level. The probabilities are then normalized so that $\sum_{n_t^X=0}^{\infty} \Pi_t^{n_t^X} = 1$. After that, the particle weights π_t^{s,n_t^X} are normalized to sum to one for each filter n_t^X.

Resampling. For each filter n_t^X, $\mathcal{N}$ new particles are Monte Carlo sampled from the particle sets with attached weights $\{([\tilde{\boldsymbol{\xi}}_t^{1,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{1,n_t^X}], \pi_t^{1,n_t^X}), \ldots, ([\tilde{\boldsymbol{\xi}}_t^{a\mathcal{N},1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{a\mathcal{N},n_t^X}], \pi_t^{a\mathcal{N},n_t^X})\}$. (As noted in Section 3.1, this means that the frequency with which each particle is resampled is proportional to the weight.) The result is a set of particles $\{[\boldsymbol{\xi}_t^{1,1}, \ldots, \boldsymbol{\xi}_t^{1,n_t^X}], \ldots, [\boldsymbol{\xi}_t^{\mathcal{N},1}, \ldots, \boldsymbol{\xi}_t^{\mathcal{N},n_t^X}]\}$, with equal weights, for each level n_t^X. Together with the weights $\Pi_t^{n_t^X}$, the particle clouds represent the posterior multi-object probability density $f_{\Gamma_t \mid \Sigma_{1:t}}(X_t \mid Z_{1:t})$.

5. Extracting expected object states

In the single object case, the result of the tracking at time t is often considered to be the expected value of the object state. The corresponding function in FISST is the probability hypothesis density (PHD) [16]:

$$D_{\mathbf{X}_t \mid \Sigma_{1:t}}(\mathbf{x}_t \mid Z_{1:t}) = \int f_{\Gamma_t \mid \Sigma_{1:t}}(\{\mathbf{x}_t\} \cup Y \mid Z_{1:t})\, \delta Y \,. \tag{15}$$

This entity is a distribution over Θ, which 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.

Given that the objects are separated (on a certain scale) in Θ, the estimated object states can be detected as peaks in the PHD. Thus, it is important to develop a good peak-detector [16]. Here, we fit a mixture of Gaussian distributions to the PHD and take the mean of each Gaussian as the estimated state of that object.

5.1. Approximate PHD

We now describe the computation of the PHD from a FISST particle filter.

Each composite particle $[\boldsymbol{\xi}_t^{i,1}, \ldots, \boldsymbol{\xi}_t^{i,n_t^X}]$ is divided into n_t^X parts, each representing one of the objects. The parts of all particles are weighted by the probability $\Pi_t^{n_t^X}$ (Section 4.1), and collected into a histogram covering the state-space Θ. The histogram is normalized to sum to $E[N_t^X] = \sum_{n=0}^{4} n\Pi_t^n$. Now, the histogram constitute a discrete approximation of the PHD. Figure 1 show an example of such an approximated PHD.

From the PHD, local maxima are detected by fitting a mixture of Gaussians to the histogram.

6. Terrain application

A simulated scenario of three vehicles moving in terrain is here used to visualize FISST particle filtering.

The reason to use particle filtering for terrain tracking is clarified in Section 6.3 – 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.

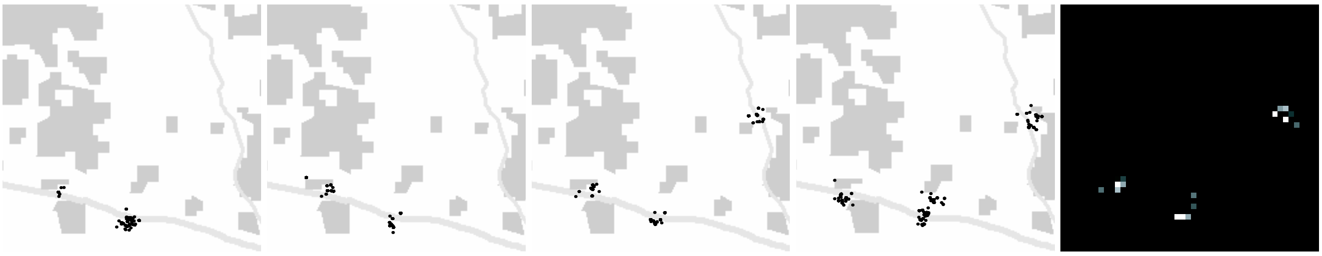
The result of the tracking is considered to be an estimated PHD over the area traveled by the vehicles. Peaks in the PHD correspond to estimated vehicle locations. These are detected by fitting a mixture of Gaussians to the PHD.

6.1. Scenario

The scenario is 841 seconds long, simulated in time-steps of five seconds. 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.

The terrain is represented by a discrete map m over position. A pixel in m can take any value $T =$

[4] The integer a depends on how many terms there are in the sums in Eq (7) since each term $F_m(\mathbf{l})$ contributes with $\mathcal{N}$ particles.



(a) 1 object, $\Pi_t^1 = 0.00$ (b) 2 objects, $\Pi_t^2 = 0.01$ (c) 3 objects, $\Pi_t^3 = 0.95$ (d) 4 objects, $\Pi_t^4 = 0.04$ (e) Estimated PHD

Figure 1: Computation of the PHD from the FISST particle filter. (a)-(d): Particles (black dots in the terrain map) conditioned on different hypotheses of object count n_t^X. Since particles simultaneously represent n_t^X objects, each particle is visualized as n_t^X dots, one for each object. This means that plot (d) contains four times as many dots as plot (a). (e): The particles for each level $n_t^X = 1$ to 4 are weighted with $\Pi_t^{N_t^X}$ and collected into a histogram that approximate the PHD over the terrain area. In this case, the PHD is mainly influenced by the level 3 particles. Each pixel is grey-level coded according to the expected number of objects in the area covered by the pixel (black – 0, white – 0.2).

$\{road, field, forest\}$ (exemplified in Figure 1a-d 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 $p_T(road) = 0.66$, $p_T(field) = 0.33$, $p_T(forest) = 0.01$.

At each time-step, each vehicle is observed with probability 0.9, 0.5 or 0.1 (corresponding, e.g., to different sight conditions for the observer). 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. Since the observations in this application originate from human observers rather than automatic sensors, the probability of false observations, p_{FP}, is very low, 10^{-10}. An observation 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).

6.2. State-space

The state vector for one 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.

6.3. Motion model

The motion model of the vehicles is

$$\mathbf{X}_t = \mathbf{X}_{t-1} + d\mathbf{X}_{t-1} + \mathbf{W}_t \tag{16}$$

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.

6.4. Birth model

We assume the birth rate p_B and death rate p_D of objects 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 objects (i.e. represent all objects with a particle cloud at all times) while not significantly overestimating the number of objects. 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 (while hypotheses can die at every time-step regardless of p_{FN}). The mean number of steps between observations is $\frac{1}{1-p_{FN}}$. Therefore, we set

$$p_B = K^{1-p_{FN}} \ , \tag{17}$$

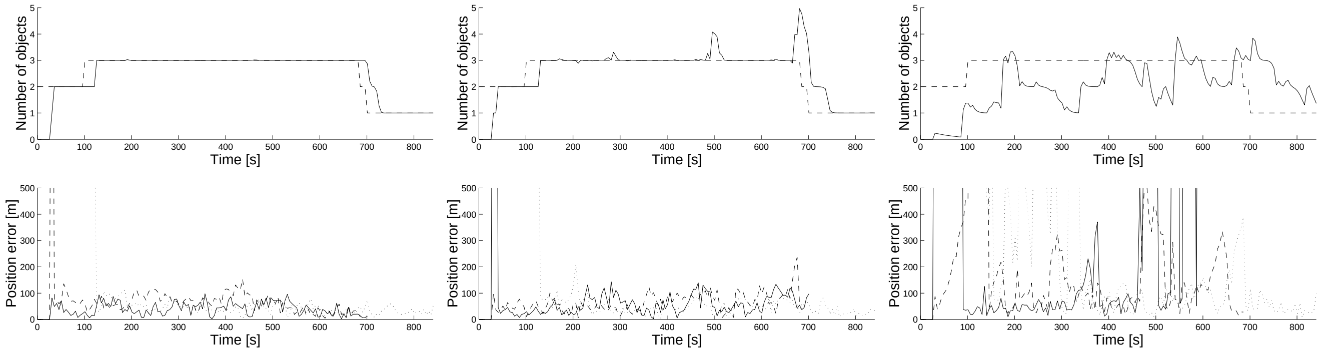
$$p_D = K \ . \tag{18}$$

The constant $K = 10^{-6}$ is set empirically.[5]

6.5. Observation model

As mentioned in Section 6.1, observations $\mathbf{Z}_t$ are of the same form as the state-space for one object $\mathbf{X}_t$ (Section 4),

[5]In principle, a high death and birth rate gives more noise in the estimated number of objects, whereas a low death and birth rate leads to low flexibility when the number of objects change and when an object is lost by the tracker. A high birth rate alone will lead to an overestimation of the number of objects, while a high death-rate will have the opposite effect. See also Section 6.6.



(a) $p_{FN} = 0.1$ (b) $p_{FN} = 0.5$ (c) $p_{FN} = 0.9$

Figure 2: Tracking errors. (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 objects, 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 object appears after 101 seconds, the dashed object disappears after 687 seconds and the solid object after 702 seconds. Position error is measured as the Euclidean distance from the true object position to the nearest detected maxima in the estimated PHD (Section 5.1).

which means that Eq (3) is simplified to

$$\mathbf{Z}_t = \mathbf{X}_t + \mathbf{V}_t \ . \tag{19}$$

The observation noise $\mathbf{V}_t$ is normally distributed with standard deviation $\boldsymbol{\sigma}_V = \boldsymbol{\sigma}_R$ (Section 6.1).

6.6. Results

Using the settings described above, three simulations were generated to show the performance of the tracker. The number of particles on each level was $\mathcal{N} = 200$. The results are shown in Figure 2.[6] The tracking performance was measured in two ways, comparing the estimated number of objects with the true value (upper graph in each subfigure), and measuring the Euclidean distance between the ground truth object positions and the local maxima in the PHD histogram.

In Figure 2a, a simulation with $p_{FN} = 0.1$ is shown. Both the number of objects (upper graph) and the location of the objects (lower graph) is accurately estimated. When $p_{FN} = 0.5$ (Figure 2b), there are a few errors in the estimated number of objects, but the model still maintains track of all objects.

However, when $p_{FN} = 0.9$ (Figure 2c), both the estimation of the number of vehicles, as well as of the vehicle locations, fail. One reason could be that each level is represented by too few particles. To test this hypothesis, the same experiment was performed with $\mathcal{N} = 1000$ particles. However, this did not improve the performance of the tracking, which lead to the conclusion that $\mathcal{N} = 200$ particles were sufficient to represent the distributions.

The reason could instead be, that even though the birth rate is very high, the hypotheses often die out before they are "confirmed" by a second observation. Therefore the number of objects is often underestimated. Another reason could be that the motion model is not informative enough to maintain track of the vehicles with such few observations.

The filter was implemented in Matlab, running in Linux on an ordinary desktop computer. One iteration of the filter required 4.9 seconds on average. Since the time-steps in this application are relatively long (here 5 seconds), this indicate the usability of the filter in a real-time system for terrain tracking.

7. Conclusions

The problem addressed in this paper was tracking of vehicles in terrain. Due to the terrain, the motion of the vehicles was highly non-linear, making Kalman approaches inappropriate. Furthermore, the number of vehicles was unknown and varied over time.

To address these problems, a generalized particle filter, approximating the optimal posterior distribution over a random set, was presented. A particle filter in its original form can be seen as a special case of this filter, when the cardinality of the set is known to be 1. The FISST filtering formulation was also extended with an birth model in which the object hypotheses were born from old observations. Re-

[6]Movies of the three tracking examples can be found at `http://www.foi.se/fusion/mpg/WOMOT03/`. Two movies relating to each of the Figures 2a, 2b, and 2c can be found. For, e.g., Figure 2a, the movie `phdFigure2(a).mpg` shows the PHD histogram (blue – 0, red – 0.2) with white 95% error ellipses indicating the Gaussians fitted to the PHD (Section 5.1). The movie `terrainFigure2(a).mpg` shows the particle clouds (deep red for levels with high probability, lighter for lower probability) and Gaussians (deep blue for high PHD peaks, lighter for lower peaks). True vehicle positions are indicated by green +, observations by green *.

sults from a simulated scenario of three vehicles showed the particle filter is successful up to certain noise limits.

In the future, the formulations of birth and death models for high p_{FN} need to be investigated further as discussed in Section 6.6. The effect of other parameter values, such as $\mathcal{N}$, should be thoroughly investigated as well.

Furthermore, it should be noted that the time complexity of the filter increases with the maximum number of targets to track. One solution could be to have multiple, spatially limited, filters. Another approach is to propagate the PHD directly, without modeling the full random set [16]. The formulation requires high signal-to-noise ratio and objects moving independently of each other. Recently, a PHD particle filter implementation [22] has been presented.

Finally, the filter should be implemented and tested in a real-time system. The low number of particles needed indicates that the filter could be suitable for a real-time implementation in an application such as this, where the iteration time is relatively long.

Acknowledgments. This work was financed by FOI project E7037. The authors wish to thank Ronald Mahler and Pontus Svenson 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] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, 1975.

[3] S. Challa, B-N. Vo, and X. Wang. Bayesian approaches to track existence – IPDA and random sets. In *International Conference on Information Fusion*, volume 2, pages 1228–1235, 2002.

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

[5] A. Doucet, B-N. Vo, C. Andrieu, and M. Davy. Particle filtering for multi-target tracking and sensor management. In *International Conference on Information Fusion*, volume 1, pages 474–481, 2002.

[6] 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.

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

[8] 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.

[9] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] J. Miller, A. Srivastava, and U. Grenander. Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition. *IEEE Transactions on Signal Processing*, 43(11):2678–2690, 1995.

[19] 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.

[20] D. Mušicki and R. Evans. Joint integrated probabilistic data association – JIPDA. In *International Conference on Information Fusion*, volume 2, pages 1120–1125, 2002.

[21] D. Mušicki, R. Evans, and S. Stanković. Integrated probabilistic data association (IPDA). *IEEE Transactions on Automatic Control*, AC-39(6):1237–1241, 1994.

[22] H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *International Conference on Information Fusion*, 2003.

[23] 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.

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