Particle Filtering for Random Sets

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

Tracking of multiple objects simultaneously over time is an important research problem. When the number of objects to track is known, standard Bayesian methods like PDA/JPDA can be employed. However, when the number of objects to track is unknown or varies over time, tracking hypotheses with different numbers of objects have to be compared. This can be addressed in a mathematically grounded manner by viewing the set of object as a *random set*, in which the number of objects, N, is a stochastic variable. Tracking of random sets is formulated with *finite set statistics (FISST)*.

In this paper, we present a *FISST particle filter*, which is an extension of a Bayesian particle filter to incorporate the FISST formalism. Experiments show the FISST particle filter to be able to estimate both the number of tracked objects, as well as the states of the objects, robustly from noisy observations.

Submitted to IEEE Transactions on Aerospace and Electronic Systems, March 2003

1 Introduction

Tracking of multiple objects simultaneously over time is an important problem in statistics, computer vision, signal processing, as well as other areas. The objects could either be connected so that their parameter spaces and motion models are strongly dependent [5, 28, 30, 32] or moving freely, nearly independent of each other [9, 11, 14, 15, 17, 18, 19, 21, 22, 23, 25, 29, 34].

The difference between tracking a single object and a known number, n, of objects is the object-data association problem [2, 9], resulting in non-linear observation models. Using these observation models, Bayesian iterative methods like particle filtering (Section 3.1) can be applied 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 of different dimensionality. The path we take to address this problem, is to view the set of objects to track as a *random set* [10, 20], 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) [10, 20], 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 along with the distribution over the rest of the state-space (Figure 1c). The FISST formulation of the tracking problem is presented in Section 4.

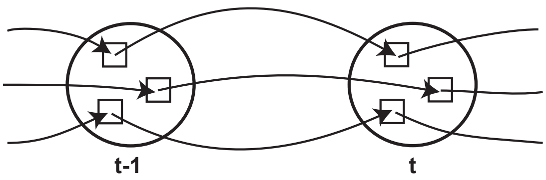
The contribution of this report is a general formulation of the particle filter, extended to approximate the optimal FISST tracking formulation (Section 5). An application [33] to tracking vehicles in terrain has also been developed. A simulated problem of tracking three objects moving non-linearly on a surface is presented in Section 7 to test the performance of the FISST particle filter. Furthermore, possibilities for reducing the computational complexity of the full tracking solution is discussed in Section 6.

Notation. The notation in this report is selected according to the notation of Mahler [20]: Scalar random variables are denoted with uppercase Roman letters, outcomes of such variables with the corresponding lowercase letter. Random sets are given uppercase Greek letters, the corresponding set outcomes uppercase Roman letters. Vector-valued random variables (random vectors) are given bold uppercase Roman letters, while their outcomes are given the corresponding bold lowercase letter. Particle representations of random vectors are given lowercase bold Greek letters. Furthermore, a superscript number after a variable is usually an index, while a superscript number after a variable within parentheses denotes an exponent. Sets are denoted with curly braces $\{\}$, while vectors are denoted with brackets $[\,]$.

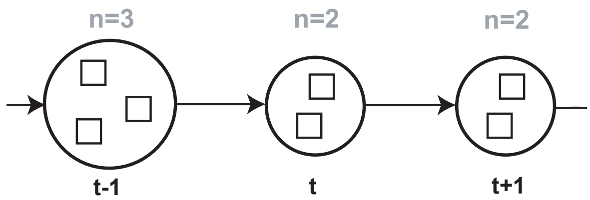
2 Related Work

In many applications of multi-object tracking, the number of objects to track, n, is known beforehand [5, 15, 19, 28, 30, 32]. 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 addressing this problem have been presented, e.g. PDA [2] and JPDA [9].

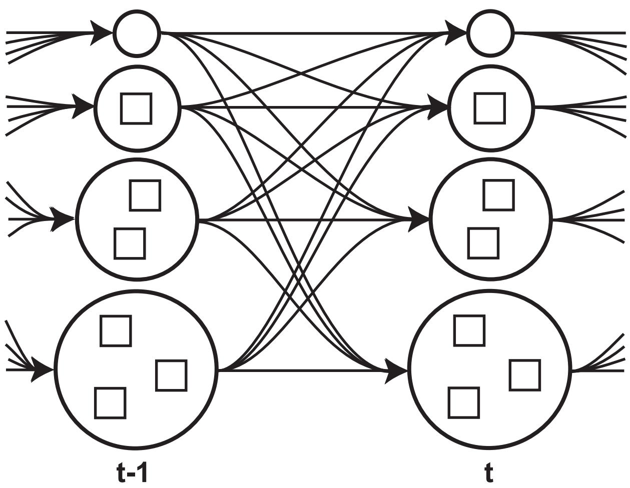
The task of tracking an unknown and changing number N of objects is, from a modeling perspective, an even more challenging one. In principle, the number of objects is then treated as a discrete state variable. However, since each object has its own vector of state variables, the



(a) Tracking with known and constant n



(b) Tracking with separate estimation of n



(c) Tracking with FISST

Figure 1: Illustration of the FISST tracking concept. (a) If the number of objects is known and constant, they can be tracked in parallel, by concatenating the states of all objects. (b) The number of objects in each time step can be estimated deterministically in each time step. Thereafter, the states of each new object is estimated from the previous set of objects. (c) When tracking with a random set, a probability distribution over all hypotheses about object count is maintained. For each object count n, a belief distribution over the states of the n individual objects is also maintained. In a time step, both the distribution over individual objects and the distribution over object count are propagated using knowledge about object motion and the probabilities of appearance and disappearance of objects.

dimensionality of the combined state-space changes with the number of objects.

One approach to address this problem [9, 15] 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 [34] 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 [20] for a vague mathematical foundation.

The particle filters of Isard and MacCormick [17] 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 tracking of a changing number of objects from a changing number of observations. In FISST, 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 [20, 23, 25], however, 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 [8, 13, 24] 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 [8]. IPDA [27] and JIPDA [26] 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 object, 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 object 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

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

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

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

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

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

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

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:

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

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 Filtering

If the shape of the posterior distribution is close to Gaussian, and the functions $h(.)$ and $\phi(.)$ are 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, often with several maxima. In these cases, a Kalman filter is no longer applicable.

Particle filtering, also known as bootstrap filtering [12] or Condensation [16], 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 a general description of sequential Monte Carlo and particle filtering, see [7]. For an overview of the state of the art in applications of particle filters, see [6].

In a particle filter, 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 [12, 16]. 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 $s = 1, ..., \mathcal{N}$. 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 Finite Set Statistics (FISST)

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 the random set formalism to account for this. The set of objects to be tracked can be considered a random set [10, 20]. A random finite set, e.g. Γ, is a finite set whose elements, as well as their number, N^X, are random. In our application, these elements are state (random) vectors for different objects, denoted e.g. $\mathbf{X}^i$ for $i = 1, \ldots, N^X$.

The mathematical framework for handling random sets in a probabilistic manner is called finite-set statistics (FISST) [10, 20]. FISST is a generalization of Bayesian theory to describe statistical properties of random sets.

In a tracking framework, FISST is used to reformulate the problem of tracking an unknown number of objects from a set of observations, to the problem of tracking a "global object" from a "global observation". The formalism is described below. We use this to extend our tracking framework to encompass multiple objects in a mathematically sound way.

The FISST equivalent of a probability $p_{\mathbf{X}}(S) = P(\mathbf{X} \in S)$ for a random vector $\mathbf{X}$, where S is a subset of the state space, is an entity called belief function $\beta_\Gamma(S) = P(\Gamma \subseteq S)$ where $\Gamma = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^{N^X}\}$ is a random set.[1] A certain outcome of Γ is denoted $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{n^X}\}$.

The core of FISST is the concepts of set derivative and set integral, taking into account the random nature of the size of Γ.

4.1 Set Derivative and Set Integral

From a distribution function $p_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{X} \leq \mathbf{x})$ for a d-dimensional random vector $\mathbf{X}$, a pdf can be derived via differentiation:

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{(\partial)^d p_{\mathbf{X}}}{\partial x^1 \ldots \partial x^d}(\mathbf{x}) \; . \tag{5}$$

In a similar manner, a multi-object probability density function can be derived from the belief function $\beta_\Gamma(S) = P(\Gamma \subseteq S)$ via *set differentiation* [10, 20]:

$$f_\Gamma(X) = \frac{\delta \beta_\Gamma}{\delta X}(\emptyset) = \frac{(\delta)^{n^X} \beta_\Gamma}{\delta\{\mathbf{x}^1\} \ldots \delta\{\mathbf{x}^{n^X}\}}(\emptyset) \; . \tag{6}$$

[1] The belief β is not a heuristically defined measure. In Section 4.1 the relation between belief and probability is discussed.

From a pdf, a probability distribution function can be computed as

$$p_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}\ . \tag{7}$$

Likewise, a belief function is obtained by set integration over a multi-object probability density function [10, 20]:

$$\beta_\Gamma(S) = \int_S f_\Gamma(X)\, \delta X = \sum_{n^X=0}^{\infty} \frac{1}{n^X!} \int_{(S)^{n^X}} f_\Gamma(\{\mathbf{x}^1, \ldots, \mathbf{x}^{n^X}\})\, d\mathbf{x}^1 \ldots d\mathbf{x}^{n^X} \tag{8}$$

where the first term in the sum should be interpreted as $f_\Gamma(\emptyset)$. The factor $1/n^X!$ relates to the set notation; in general, $f(\{\mathbf{x}^1, \ldots, \mathbf{x}^{n^X}\}) = n^X! f(\mathbf{x}^1, \ldots, \mathbf{x}^{n^X})$ since the vector $[\mathbf{x}^1, \ldots, \mathbf{x}^{n^X}]$ is ordered, in opposite to the set $\{\mathbf{x}^1, \ldots, \mathbf{x}^{n^X}\}$. For a more formal derivation of the relations above, see [10].

The density $f_\Gamma(X)$ encompasses densities over Γ for each specific value n^X of the number of objects N^X, as well as a discrete density function over the number of objects N^X:

$$f_\Gamma(X) = \begin{cases} f_\Gamma(\emptyset) & \text{if } n^X = 0 \\ f_\Gamma(\{\mathbf{x}\}) & \text{if } n^X = 1 \\ f_\Gamma(\{\mathbf{x}^1, \mathbf{x}^2\}) & \text{if } n^X = 2 \\ \text{etc.} & \end{cases} \tag{9}$$

In the case of zero objects, it collapses to $f_\Gamma(\emptyset) = P(N^X = 0)$. For $n^X > 0$, the discrete marginal distribution over object count N^X is

$$p_{N^X}(n^X) = P(N^X = n^X) = \frac{1}{n^X!} \int_{(\Theta)^{n^X}} f_\Gamma(\{\mathbf{x}^1, \ldots, \mathbf{x}^{n^X}\})\, d\mathbf{x}^1 \ldots d\mathbf{x}^{n^X} \tag{10}$$

where Θ is the total state space of one object. This is the probability that there are exactly n^X objects in the system.

Hence, the set integral over the whole state space is

$$\sum_{n^X=0}^{\infty} p_{N^X}(n^X) = \sum_{n^X=0}^{\infty} \frac{1}{n^X!} \int_{(\Theta)^{n^X}} f_\Gamma(\{\mathbf{x}^1, \ldots, \mathbf{x}^{n^X}\})\, d\mathbf{x}^1 \ldots d\mathbf{x}^{n^X} = 1\ . \tag{11}$$

4.2 Tracking of Multiple Objects Using FISST

FISST has been used (e.g. [20]) to formulate a framework for tracking of an unknown number of objects from a set of observations. Here, this framework is described. Furthermore, 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 $Z_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^{n_t^Z}\}$.

Using FISST formalism [10, 20], it can be shown that the set equivalents of the prior and posterior densities in Eqs (1) and (4) become

$$f_{\Gamma_t \,|\, \Sigma_{1:t-1}}(X_t \,|\, Z_{1:t-1}) = \int f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{1:t-1}}(X_t \,|\, X_{t-1}, Z_{1:t-1}) \; f_{\Gamma_{t-1} \,|\, \Sigma_{1:t-1}}(X_{t-1} \,|\, Z_{1:t-1}) \; \delta X_{t-1} \;, \tag{12}$$

$$f_{\Gamma_t \,|\, \Sigma_{1:t}}(X_t \,|\, Z_{1:t}) \propto f_{\Sigma_t \,|\, \Gamma_t}(Z_t \,|\, X_t) \; f_{\Gamma_t \,|\, \Sigma_{1:t-1}}(X_t \,|\, Z_{1:t-1}) \;. \tag{13}$$

We will now discuss these equations in more detail.

Prediction. The belief density function in the left-hand side of Eq (12) is denoted prior distribution. Using the definition of set integral in Eq (8), the prior can be expanded as

$$\begin{aligned} f_{\Gamma_t \,|\, \Sigma_{1:t-1}}(X_t \,|\, Z_{1:t-1}) = \sum_{n_{t-1}^X = 0}^{\infty} \frac{1}{n_{t-1}^X!} \int & f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{1:t-1}}(X_t \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{1:t-1}) \\ & f_{\Gamma_{t-1} \,|\, \Sigma_{1:t-1}}(\{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\} \,|\, Z_{1:t-1}) \; d\mathbf{x}_{t-1}^1 \ldots d\mathbf{x}_{t-1}^{n_{t-1}^X} \;. \end{aligned} \tag{14}$$

In analog to Eq (1), the motion model is used to generate, for each value n_{t-1}^X, the conditional density $f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{1:t-1}}(X_t \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{1:t-1})$. The motion model for a single object is defined as in Eq (2):

$$\mathbf{X}_t = \phi(\mathbf{X}_{t-1}, \mathbf{W}_t) \tag{15}$$

i.e. with no dependence on the observations, only on the state at $t-1$. As pointed out in Section 3, this object motion model defines the conditional pdf $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$, which is a special case of $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$ in Eq (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, and the probability of birth p_B. The maximum number of objects that can be born in a single time step is denoted n^B.

Objects are born according to a birth model. One option is to generate objects from a uniform distribution over the object state space as $\mathbf{X}_t \sim \mathrm{U}(\Theta)$. However, 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 observation $\mathbf{Z}_{t-1}$ at the previous time instance:

$$\mathbf{X}_t = \phi(h_{\mathbf{X}_t}^{-1}(\mathbf{Z}_{t-1}, \mathbf{V}_{t-1}), \mathbf{W}_t) \tag{16}$$

where $\mathbf{V}_{t-1}$ is the observation noise at $t-1$. Here, we assume that this is possible. This enables us to explore the state space more efficiently. It also means that the maximum number of born objects $n^B = n_{t-1}^Z$ since each observation can be associated with at most one object (see the next paragraph). The birth model defines the pdf $f_{\mathbf{X}_t \,|\, \mathbf{Z}_{t-1}}(\mathbf{x}_t \,|\, \mathbf{z}_{t-1})$ which also is a special case of the propagation pdf $f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}, \mathbf{Z}_{1:t-1}}(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$.

To account for births and deaths, the motion model for a set of objects is defined as

$$\Gamma_t = T_t^1 \cup \cdots \cup T_t^{N_{t-1}^X} \cup B_t^1 \cup \cdots \cup B_t^{N_{t-1}^Z} \ . \tag{17}$$

The random set T_t^i becomes $\emptyset$ (an object that died/disappeared) with probability p_D, and $\{\phi(\mathbf{X}_{t-1}^i, \mathbf{W}_t^i)\}$ with probability $(1-p_D)$. The set B_t^i becomes $\{\phi(h_{\mathbf{X}_t}^{-1}(\mathbf{Z}_{t-1}^i, \mathbf{V}_{t-1}^i), \mathbf{W}_t^{i+N_{t-1}^X})\}$ (a newly born object) with probability p_B, and $\emptyset$ with probability $(1-p_B)$. Here, $\mathbf{W}_t^{i+N_{t-1}^X}$ are i.i.d. random vectors as are $\mathbf{V}_t^i$.

Thus, the motion model can, from a set X_{t-1} of a certain cardinality, generate sets X_t of both lower and higher cardinality according to certain transition probabilities dependent on p_B and p_D. There is a probability > 0 of transition between level n_{t-1}^X and all levels $n_t^X \in [0, n_{t-1}^X + n_{t-1}^Z]$.

Using the model of birth, death, and motion in Eq (17), the motion distribution in Eq (14) can be expressed as a combination of the individual motion models. For a specified cardinality n_{t-1}^X of Γ_{t-1}, the motion distribution in Eq (14) is, for each cardinality level n_t^X, equal [10] to

$$f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{t-1}}(X_t \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{t-1}) = \\ \sum_{k=\max(0, n_t^X - n_{t-1}^X)}^{\min(n_{t-1}^Z, n_t^X)} P_m(k) \begin{pmatrix} n_t^X \\ k \end{pmatrix} \sum_{l_1 \neq \cdots \neq l_k \in [1, n_{t-1}^Z]} \ \sum_{l_{k+1} \neq \cdots \neq l_{n_t^X} \in [1, n_{t-1}^X]} F_m(\mathbf{l}) \tag{18}$$

where

$$P_m(k) = (p_B)^k (1-p_B)^{n_{t-1}^Z - k} (p_D)^{n_{t-1}^X - (n_t^X - k)} (1-p_D)^{n_t^X - k} \ , \tag{19}$$

$$F_m(\mathbf{l}) = f_{\mathbf{X}_t \,|\, \mathbf{Z}_{t-1}}(\mathbf{x}_t^1 \,|\, \mathbf{z}_{t-1}^{l_1}) \ldots f_{\mathbf{X}_t \,|\, \mathbf{Z}_{t-1}}(\mathbf{x}_t^k \,|\, \mathbf{z}_{t-1}^{l_k}) \\ f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t^{k+1} \,|\, \mathbf{x}_{t-1}^{l_{k+1}}) \ldots f_{\mathbf{X}_t \,|\, \mathbf{X}_{t-1}}(\mathbf{x}_t^{n_t^X} \,|\, \mathbf{x}_{t-1}^{l_{n_t^X}}) \ . \tag{20}$$

$P_m(k)$ 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. 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 $\left(\begin{array}{c} n_t^X \\ k \end{array} \right)$, 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.

In the case of $n_t^X = 0$ objects, Eq (18) collapses to $f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{t-1}}(\emptyset \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{t-1}) = (1 - p_B)^{n_{t-1}^Z} (p_D)^{n_{t-1}^X}$.

Observation. The observation model is used to generate the likelihood density $f_{\Sigma_t \,|\, \Gamma_t}(Z_t \,|\, X_t)$. As in the single object case in Eq (3), an observation is generated from exactly one object, according to the model[2]

$$\mathbf{Z}_t = h(\mathbf{X}_t, \mathbf{V}_t) \ . \tag{21}$$

At a certain time step, there can be both missing observations and clutter (i.e. spurious observations that do not correspond to real objects). 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 $f_{\mathbf{Z}_t}$. (For many types of sensors, $\Theta = \Theta_o$.) Each sensor sends zero or one observation. Similarly to the motion model, the generative model for a random set of observations is then

$$\Sigma_t = T_t^1 \cup \cdots \cup T_t^{N_t^X} \cup C_t^1 \cup \cdots \cup C_t^{n^S} \ ,$$

[2]This is of course dependent of the type of sensor that generates the observations. In this paper, we have an application in mind in which the observations are delivered from n^S spatially separated sensors that observe one or zero objects in each time step. The sensors could for example be human observers, and the objects vehicles. However, in a vision based tracking application the observation would be an image, in which all objects appear. In that case, we will always have exactly one observation, which would be generated according to a model $\mathbf{Z}_t = h(\Gamma_t, \mathbf{V}_t)$. The objects can be visible or non-visible in the observation, due to occlusion.

$$|\Sigma_t| = N_t^Z \tag{22}$$

where n^S is the number of sensors.

The random set T_t^i becomes $\emptyset$ (a missing observation, false negative) with probability p_{FN}, and $\{h(\mathbf{X}_t^i, \mathbf{V}_t^i)\}$ with probability $(1 - p_{FN})$. The set C_t^i becomes $\{\mathbf{C}_t^i\}$ (clutter, false positive) with probability p_{FP}, and $\emptyset$ with probability $(1 - p_{FP})$. Here, $\mathbf{V}_t^i$ are i.i.d. random vectors, as are $\mathbf{C}_t^i$. The clutter observations are distributed according to $\mathbf{C}_t^i \sim U(\Theta_o)$.

This definition yields the likelihood density

$$f_{\Sigma_t \,|\, \Gamma_t}(Z_t \,|\, X_t) = \sum_{k=0}^{\min(n_t^Z, n_t^X)} P_o(k) (f_{\mathbf{Z}_t})^{n_t^Z - k} \sum_{l_1 \neq \cdots \neq l_k \in [1, n_{t-1}^Z]} \; \sum_{m_1 \neq \cdots \neq m_k \in [1, n_t^X]} F_o(\mathbf{l}, \mathbf{m}) \tag{23}$$

where

$$P_o(k) = (p_{FN})^{n_t^X - k} (1 - p_{FN})^k (1 - p_{FP})^{n^S - (n_t^Z - k)} (p_{FP})^{n_t^Z - k} \;, \tag{24}$$

$$F_o(\mathbf{l}, \mathbf{m}) = f_{\mathbf{Z}_t \,|\, \mathbf{X}_t}(\mathbf{z}_t^{l_1} \,|\, \mathbf{x}_t^{m_1}) \ldots f_{\mathbf{Z}_t \,|\, \mathbf{X}_t}(\mathbf{z}_t^{l_k} \,|\, \mathbf{x}_t^{m_k}) \;. \tag{25}$$

$P_o(k)$ 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.

4.3 Extracting Expected Object States from the Belief Distribution

In the single target case, the result of the tracking at time t is often considered to be the expected value of the object state. This is computed as

$$E[\mathbf{X}_t] = \int \mathbf{x}_t f_{\mathbf{X}_t | \mathbf{Z}_{1:t}}(\mathbf{x}_t | \mathbf{z}_{1:t}) \, d\mathbf{x}_t \;. \tag{26}$$

The corresponding function in FISST is the probability hypothesis density (PHD) [10, 23]:

$$D_{\mathbf{X}_t | \Sigma_{1:t}}(\mathbf{x}_t | Z_{1:t}) = \int f_{\Gamma_t | \Sigma_{1:t}}(\{\mathbf{x}_t\} \cup Y | Z_{1:t}) \, \delta Y \;. \tag{27}$$

This entity can be interpreted as a distribution over Θ, which has the property that, for any subset $S \subseteq \Theta$, the expected number of objects in S at time t can be expressed as

$$E[|\Gamma_t \cap S|] = \int_S D_{\mathbf{X}_t|\Sigma_{1:t}}(\mathbf{x}_t|Z_{1:t}) \, d\mathbf{x}_t \ . \tag{28}$$

Hence, the expected number of objects in Θ can be computed as

$$E[N_t^X] = \int_\Theta D_{\mathbf{X}_t|\Sigma_{1:t}}(\mathbf{x}_t|Z_{1:t}) \, d\mathbf{x}_t \ . \tag{29}$$

Given that the targets are separated (on a certain scale) in Θ, the estimated object states can be detected as $E[N_t^X]$ peaks in the distribution $D_{\mathbf{X}_t|\Sigma_{1:t}}(\mathbf{x}_t|Z_{1:t})$. Thus, it is important to develop a good peak-detector. One option is to fit $E[N_t^X]$ Gaussian distributions to the distribution D using the EM algorithm [4] and take the mean of each Gaussian as the estimated state of that object.

5 The FISST Particle Filter

We will now describe the particle filter implementation of Eqs (13), (14), (18) and (23). To enable comparison between the FISST particle filter and a particle filter in its original form, the description in Section 5.2 follows the structure of Section 3.1.

5.1 Representing a Random Set with Particles

First, some general comments about the particle representation of a particle filter is given.

In the original formulation of a particle filter, a set of particles represents a pdf (with integral 1). We have chosen to represent a multi-object density function (which has an integral ≤ 1) by a set of particles whose weights sum to 1, and a number $p \leq 1$ representing the integral over the belief distribution, i.e. the “scaling” of the distribution.

For each level n^X in the random set representation, the probability or "scaling" $p_{N^X}(n^X)$ (Eq (10)) is maintained. Furthermore, each level $n^X > 0$ is represented by a separate set of particles. If the particle filter at level 1 has dimensionality d, then the filter at level 2 has dimensionality $2d$ and so on.

5.2 Particle Filter Implementation of the Tracking

Consider the filter for level n_t^X, i.e. the filter conditioned on the existence of exactly n_t^X objects. The posterior density $\frac{1}{n_t^X!} f_{\Gamma_t \mid \Sigma_{1:t}}(\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t^X}\} \mid Z_{1:t})$ at time t is represented by the integral (i.e. probability for n_t^X objects given the data, see Eq (10)) $\Pi_t^{n_t^X} = p_{N_t^X}(n_t^X)$, 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 an hypothesis s about the state of object i at level n_t^X. One time step proceeds as follows:

Prediction. For each level n_t^X at time t, the transition probabilities 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 $t-1$ (see Eq (18)).

For each new level n_t^X, particles are sampled from each old level n_{t-1}^X with transition probability > 0, and propagated in time by sampling from the motion distributions in the inner sum in Eq (18) (time propagation for a particle representing a single object is described in Section 3.1). For each term in the inner sum, $\mathcal{N}$ single-object particles (of dimensionality d) are sampled from each distribution in the product. The particles are then concatenated to form composite particles (of dimensionality $n_t^X d$). Each composite particle s is given a prior weight ϖ_t^{s,n_t^X} equal to the transition probability from level n_{t-1}^X to n_t^X, times the probability $\Pi_{t-1}^{n_{t-1}^X}$. The propagated particles with attached prior weights, $\{([\tilde{\boldsymbol{\xi}}_t^{1,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{1,n_t^X}], \varpi_t^{1,n_t^X}), \ldots, ([\tilde{\boldsymbol{\xi}}_t^{a\mathcal{N},1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{a\mathcal{N},n_t^X}], \varpi_t^{a\mathcal{N},n_t^X})\}$, at all levels

n_t^X, where a is a positive integer[3], represent the prior distribution $f_{\Gamma_t \,|\, \Sigma_{1:t-1}}(\{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t^X}\} \,|\, Z_{1:t-1})$.

Observation. At each level $n_t^X > 0$, the likelihood (Eq (23)) of each composite particle with prior weight, $([\tilde{\boldsymbol{\xi}}_t^{s,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{s,n_t^X}], \varpi_t^{s,n_t^X})$, is evaluated. The particles are assigned posterior weights $\pi_t^{s,n_t^X} = \varpi_t^{s,n_t^X} f_{\Sigma_t \,|\, \Gamma_t}(Z_t \,|\, [\tilde{\boldsymbol{\xi}}_t^{s,1}, \ldots, \tilde{\boldsymbol{\xi}}_t^{s,n_t^X}])$.

For each level n_t^X, the probability $\Pi_t^{n_t^X}$ is computed as the sum over all posterior 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 posterior 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 its 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 pdf $f_{\Gamma_t \,|\, \Sigma_{1:t}}(X_t \,|\, Z_{1:t})$ at time t.

6 Time Complexity and Approximations

In its general formulation, the FISST particle filter is computationally heavy. In order to track a large number of objects, simplifications have to be introduced. Below, three approaches to this are discussed, although not included in the current implementation of the FISST particle filter.

[3]The integer a depends on how many terms there are in the sums in Eq (18) since each term contributes with $\mathcal{N}$ particles.

6.1 Simplifying the Filters for the Individual Objects

A natural method to speed up the filter in its entirety is to make the propagation of each individual object hypothesis faster. Mahler [21] suggests using a FISST Kalman filter, in which the marginal distribution over each object on each object count level is approximated with a Gaussian distribution, and where the motion models and observation models of each object are linear.

A FISST Kalman filter will naturally require less computational resources than a FISST particle filter. However, in tracking situations where the motion or observation models are non-linear [33], this type of filter will fail to correctly estimate the posterior distribution.

6.2 Estimation of the Random Set Distribution from the PHD

Under certain conditions (high signal/noise ratio and independently moving objects) the full random set distribution can be reconstructed from the PHD (Section 4.3). A method for tracking using only the PHD [22, 23] has lower time complexity than a tracking method that represents the entire random set structure.

Sidenbladh [31] present a particle implementation of such a PHD filter. Tests show the approximate filter to require about a tenth of the computational effort from a particle filter representing the full random set for five objects.

6.3 Pruning

An obvious method to lower the computational complexity is pruning of unlikely hypotheses (e.g. [29]). In the FISST filter described in Section 4.2 there are two opportunities for pruning.

When the probabilities of object birth p_B and death p_D are small, the integral over the belief distribution $f_{\Gamma_t \,|\, \Gamma_{t-1}, \Sigma_{t-1}}(X_t \,|\, \{\mathbf{x}_{t-1}^1, \ldots, \mathbf{x}_{t-1}^{n_{t-1}^X}\}, Z_{t-1})$ in Eq (18) is very close to 0 for most values of $n_t^X - n_{t-1}^X$. Thus, in reality, only a few terms contribute to the sum over n_{t-1}^X in Eq (14). The prior distribution will be well approximated by a sum over just those few terms.

Eq (18) presents in itself another opportunity for pruning in estimation of $f_{\Gamma_t \mid \Gamma_{t-1}, \Sigma_{t-1}}$, if p_B and p_D are small: In the sum over k, the terms with highest k will be scaled down by the constant $P_m(k) \left(\begin{array}{c} n_t^X \\ k \end{array} \right)$. Thus, these terms can sometimes be disregarded in the propagation.

Similarly, the likelihood computation (Eq (23)) gives opportunities for pruning, if p_{FN} or p_{FP} are small. The constant $P_o(k)(f_{\mathbf{Z}_t})^{n_t^Z - k}$ will be very close to zero for some values of k, regardless of the individual observation likelihoods $f_{\mathbf{Z}_t \mid \mathbf{X}_t}(\mathbf{z}_t^{l_i} \mid \mathbf{x}_t^{m_i})$. These terms can then be disregarded from the sum over k.

7 Experimental Results

To test the concept of FISST particle filtering, experiments were performed on a simulated scenario with three objects moving on a plane of size 100×100 length units. The simulation is 100 time steps long. Objects number 1 (denoted with a solid line in the figures and graphs) and 2 (dashed line) appear at time step 1, while object number 3 (dotted line) appears in time step 26. Object 2 disappears in time step 75.

7.1 Parameter Settings

State space. The state vector for one object used for tracking is $\mathbf{x}_t = [\mathbf{p}_t, \mathbf{v}_t]$ where $\mathbf{p}_t$ is position (in length units) and $\mathbf{v}_t$ velocity (in length units/time step). The random set of objects is in every time step limited according to $N_t^X \leq 4$ objects for computational reasons. Furthermore the number of sensors that can generate observations is $n_t^s = 4$. The observations in each time step of the simulation are generated according to the same observation model (Eq (22)) as that used in the likelihood estimation, with the same parameter settings.

Motion model. In the simulation, the objects move in straight paths, making 90 degrees turns on 3 occasions over a total of 250 time steps for all objects. To model that behavior, the non-linear motion model $\phi(\mathbf{X}_{t-1}, \mathbf{W}_t)$ is designed to make 90 degrees turns with a probability $p_{turn} = 3/250$. The motion noise $\mathbf{W}_t$ comprises this turn factor, as well as a normally distributed noise term with standard deviation $\boldsymbol{\sigma}_W = [0.25, 0.25, 0.05, 0.05]$ units.[4]

Birth model. We assume the birth rate p_B and death rate p_D of objects to be invariant with regard 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. The mean number of steps between observations is $\frac{1}{1-p_{FN}}$. Therefore, we set:

$$p_B = K^{1-p_{FN}} \ , \tag{30}$$

$$p_D = K \ . \tag{31}$$

The constant $K = 10^{-3}$ is set empirically.[5]

[4]Depending on the application, the motion, birth and death models should be learned or modeled from real data. The models could e.g. be position, type or time dependent. Here, the effect of changing the dynamics is not tested.

[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.

Observation model. Observations $\mathbf{Z}_t$ are of the same form as the state space for one object $\mathbf{X}_t$ (i.e. the observation space $\Theta_o = \Theta$, see Section 4.2), which means that $h(.)$ in Eq (3) is simplified to the linear function

$$\mathbf{Z}_t = \mathbf{X}_t + \mathbf{V}_t \ . \tag{32}$$

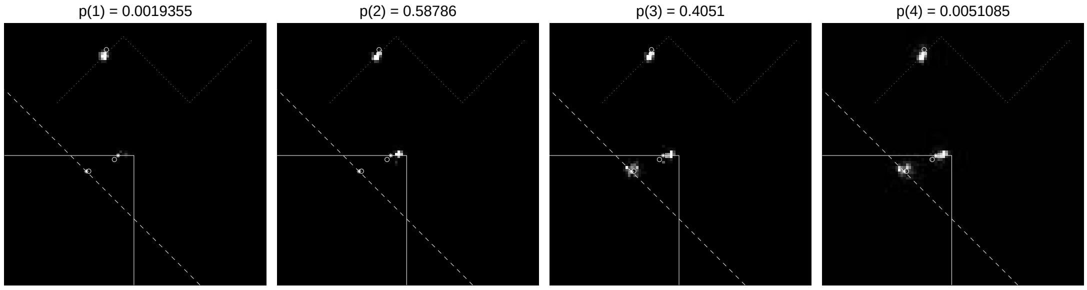
The observation noise $\mathbf{V}_t$ is normally distributed with standard deviation $\boldsymbol{\sigma}_V$, which is varied in the experiment in Section 7.3. The probabilities of missing observations, p_{FN}, and spurious observations, p_{FP}, are also varied in the experiment.

7.2 PHD Representation

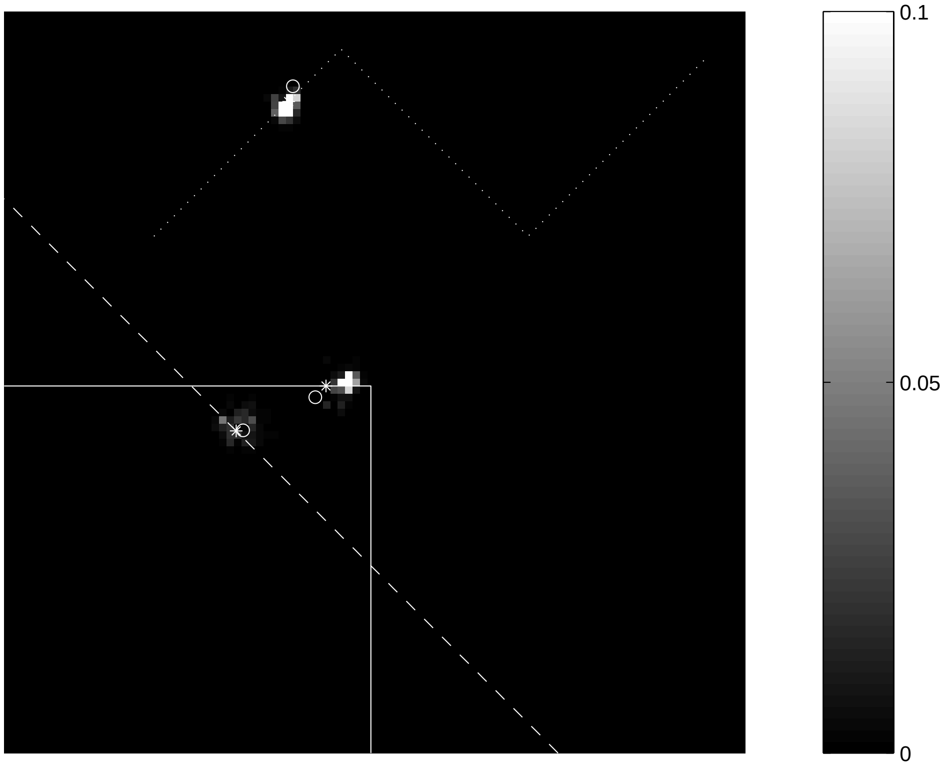
We now visualize the computation of the PHD (Section 4.3) from a random set, at time step 44 of the tracking session. The probability of missing observations is $p_{FN} = 0.5$ while the probability for clutter is $p_{FN} = 0.05$, and the observation noise is $\boldsymbol{\sigma}_V = [2.5, 2.5, 0.5, 0.5]$ length units. This corresponds to the settings of scenario (c) in Figure 3. For visibility reasons, only the positions are taken into regard. In reality, the PHD would be four-dimensional, since there are four dimensions in the state space for one object.

For each level n_t^X, the PHD conditioned on N_t^X is computed from the belief distribution on that level. This is done by computing, for each of the objects $i \in [1, n_t^X]$, the marginal belief distribution for that object, and then taking the sum over all marginal distributions. A marginal distribution over the position of object i is here generated as a two-dimensional histogram in which the $\mathcal{N}$ particles are collected according to their value of object i:s position. The histogram for each marginal distribution is normalized to sum to 1. When summing the distributions for all objects, total conditional PHD histogram will have sum n_t^X.

Figures 2a-d show the PHD-s conditioned on level 1 to 4 respectively. White denotes high



(a) Level $n_t^X = 1$ (b) Level $n_t^X = 2$ (c) Level $n_t^X = 3$ (d) Level $n_t^X = 4$



(e) Total PHD

Figure 2: PHD at time step 44 in the simulation, $P_FN = 0.5$, $P_FP = 0.05$. (a)-(d): PHD-s conditioned on different hypotheses of object number n_t^X (e): The total PHD. The true object positions are denoted with a white star, while the observations are shown as a white circle. Object "solid" and "dashed" starts at time 0, while object "dotted" appears at time 25. Object "dashed" disappears at time 75. All three objects move with constant velocity along the tracks. The particles are collected into histograms that approximate the marginal belief density over object position. The PHD is gray-level coded according to the bar in (e).

belief, while black denotes belief 0 in the PHD. Gray-level coding is shown in Figure 2e. The probability for each level n_t^X is written above the image. We can see that at level 1 (2a), mainly one of the objects (the "dotted" one) is represented by a maximum, while all three are covered at higher levels. However, the hypothesis $N_t^X = 1$ fits badly with the observations, which gives a low probability $p_{N_t^X}(1)$.

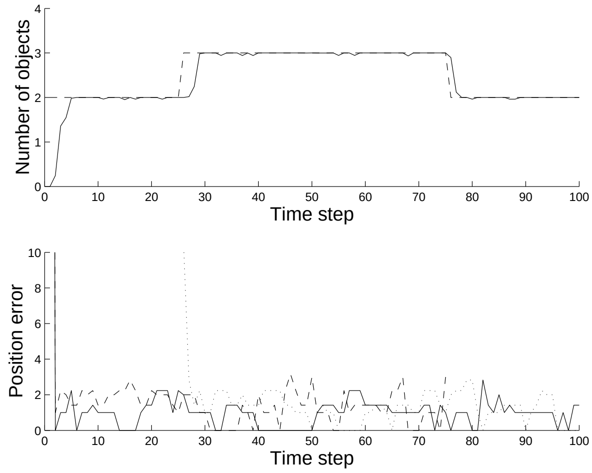
The PHD (Figure 2e) is given by the sum of the conditional PHD-s, weighted by the level probabilities $p_{N_t^X}(n_t^x)$. In this case, levels 0, 1 and 4 have very low probabilities, which means that the PHD is mainly a combination of the conditional PHD-s at levels 2 and 3. The result is a fairly accurate estimation of the position of the three objects. However, the number of objects is slightly under-estimated: $E(N_t^X) = 2.41$.

7.3 Varying Observation Parameters

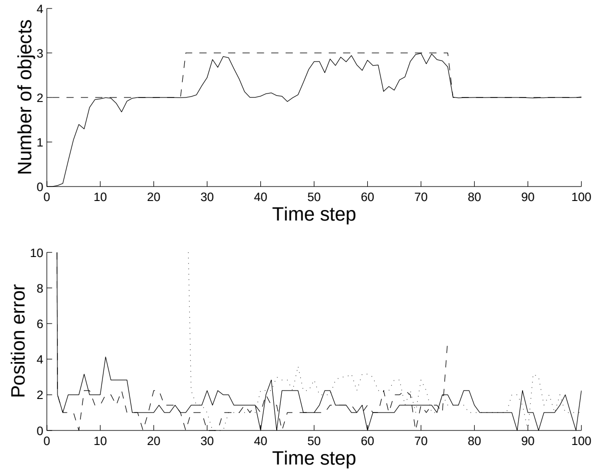
Using the settings in Section 7.1, the standard deviation $\boldsymbol{\sigma}_V$ of the observation noise, and the probability of missing observations p_{FN} and clutter p_{FP} were now varied to test the limits of the FISST particle filter. The results are shown in Figures 3 and 4.[6] The tracking 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 (Sections 4.3, 7.2) histogram.

When p_{FN} is low (Figures 3a-b, 4a-b), the number of objects N_t^X (upper graph) is estimated with high accuracy. However, the system becomes more "suspicious" against observations when p_{FP} is higher, which means that N_t^X is more often under-estimated during a few time steps than

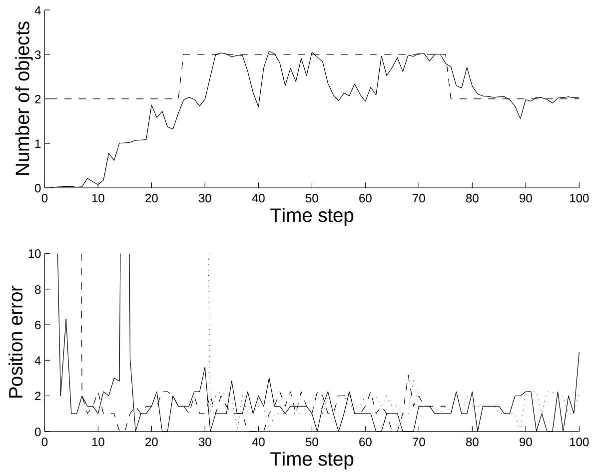
[6]MPEG Movies of the eight tracking examples in Figures 3 and 4 can be found online at `http://www.foi.se/fusion/mpg/AES03/`. For, e.g., Figure 3a, the movie `videoFigure3(a).mpg` shows the PHD histogram over time (blue – 0, red – 0.1). The object trajectories are indicated by a solid, a dashed, and a dotted white line. Ground truth object positions are indicated by white stars, observations by white circles.



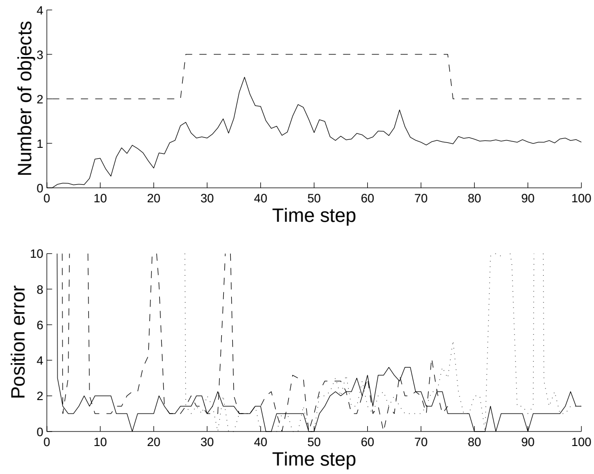
(a) Few missing observations, low clutter



(b) Few missing observations, high clutter

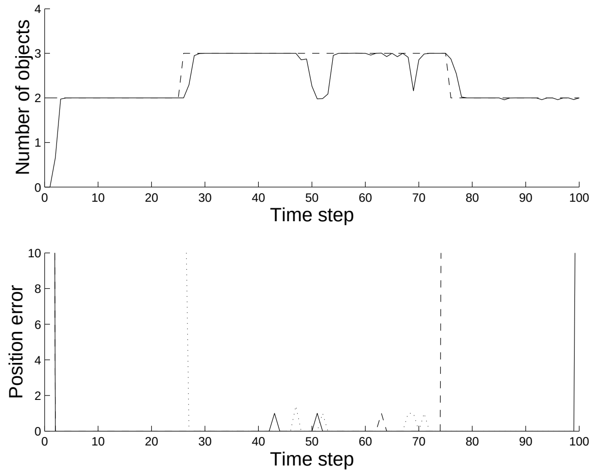


(c) Many missing observations, low clutter

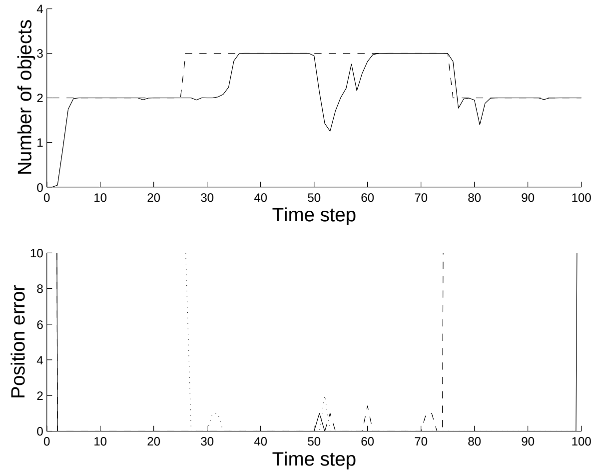


(d) Many missing observations, high clutter

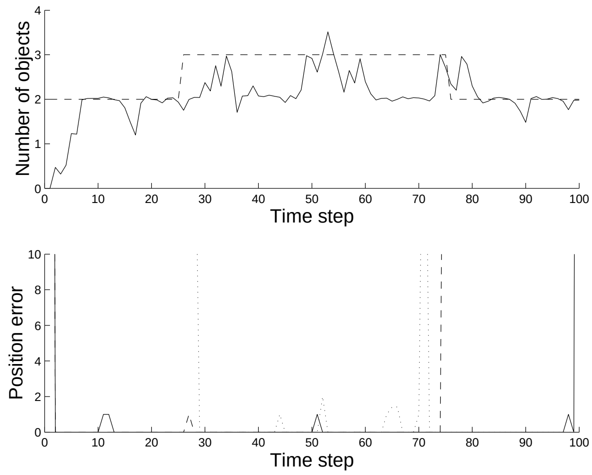
Figure 3: Tracking errors, $\boldsymbol{\sigma}_V = [2.5, 2.5, 0.5, 0.5]$ length units. The higher graph in each subfigure shows estimated (solid line) number of objects, compared to the true (dashed line) number. The lower graph in each subfigure shows position errors for the three objects. Solid, dashed and dotted lines denote different objects. The dotted object appears in time step 25, while the dashed object disappears in time step 75. Position error is measured as the Euclidean distance from the true object position to the nearest of the three largest local maxima in the estimated pdf. The different subfigures show results for varying probability of missing observation p_{FN}, and clutter p_{FP}. (a) $p_{FN} = 0.05$, $p_{FP} = 0.05$. (b) $p_{FN} = 0.05$, $p_{FP} = 0.5$. (c) $p_{FN} = 0.5$, $p_{FP} = 0.05$. (d) $p_{FN} = 0.5$, $p_{FP} = 0.5$.



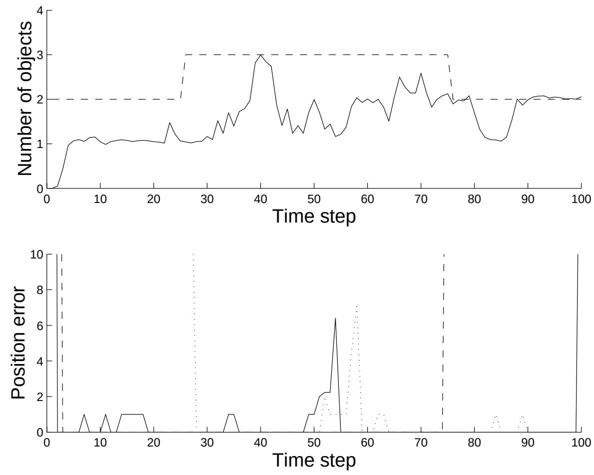
(a) Few missing observations, low clutter



(b) Few missing observations, high clutter



(c) Many missing observations, low clutter



(d) Many missing observations, high clutter

Figure 4: Tracking errors, $\boldsymbol{\sigma}_V = [0.02, 0.02, 0.05, 0.05]$ length units. The different subfigures show results for varying probability of missing observation p_{FN}, and clutter p_{FP}. (a) $p_{FN} = 0.05$, $p_{FP} = 0.05$. (b) $p_{FN} = 0.05$, $p_{FP} = 0.5$. (c) $p_{FN} = 0.5$, $p_{FP} = 0.05$. (d) $p_{FN} = 0.5$, $p_{FP} = 0.5$.

with low p_{FP}.

For high p_{FN} (Figures 3c-d, 4c-d), the estimate of N_t^X is less precise and varies more over time, than for the tracking examples with low p_{FN}. This is simply because there are less observations of the object in each time window, which means that less information enters the system. This affects the position error (lower graph in each subfigure) as well, although less dramatically.

The position errors are mainly affected by the standard deviation $\boldsymbol{\sigma}_V$, a little by p_{FN}. In Figure 3, $\boldsymbol{\sigma}_V$ is ten times higher than in Figure 4. Consequently, the position errors are higher in Figure 3 for all values of p_{FN} and p_{FP}.

In Figure 3d, the tracking fails since the number of objects is under-estimated. The reason is that the parameters $\boldsymbol{\sigma}_V$, p_{FN} and p_{FP} are all high, which means that the observation signal to noise ratio is low. In such observation conditions, one would need motion, birth and death models that better model the true behavior of the tracked objects. However, when $\boldsymbol{\sigma}_V$ is lowered (Figure 4d) the system is able to estimate the number of objects better, and the errors are again low.

To summarize, more precise observations give tracking with higher precision (i.e. lower position error), and a low percentage of missing observations gives a more precise estimate of the number of objects. Spurious observations do not affect the tracking to any great extent.

It should be pointed out that the system (correctly according to Occam's principle) strives to explain the observations with as few objects as possible. Therefore, the number of objects tends to be under-estimated.

It is also important to note, although this is not tested in this experiment, that the performance of the tracking depends on the motion, birth and death models, and on how well the distributions are approximated – in the particle filter case, how many particles are used.

8 Conclusions

In this paper, a FISST particle filter, approximating the optimal FISST tracking formulation [20], was presented. The set of tracked objects were viewed as a random set, which enabled us to treat the number of objects, N_t^X, as a stochastic variable. Thus, both N_t^X and the states of the individual objects were estimated in an integrated manner in each time-step of the tracking.

The filter was implemented in its most general form. However, simplifications to lower the computational complexity can be introduced. This was discussed in Section 6.

The FISST particle filter tracking was tested on a scenario with a varying number of multiple objects moving non-linearly. It showed to be robust to both missing observations, spurious observations and observation noise.

8.1 Future Work

The current filter can be augmented in several directions. We here discuss three such extensions.

Objects of different types. Discrete variables, such as type, can be included in the state-space. In contrast to position and velocity, type is a static property of an object. This means that there is no motion model; a particle of a certain type should not change type when it is propagated or resampled. Otherwise, it is treated as a variable in the state-space.

State-dependent birth, death and motion models. In the example in Section 7, the probabilities for birth and death are constant with respect to the state. However, in many applications, these probabilities vary both in space and time. Similarly for motion, if the application is tracking of vehicles in terrain, the motion model should include the movability in different areas of terrain.

Models that better resemble the true conditions will lead to a more efficient tracking.

Approximations. In order to lower the time complexity of the tracking, relevant approximations should be introduced. Some approaches were discussed in Section 6. By starting with the general formulation and introducing approximations depending on the tracking conditions specific to the application, one can model the effects of that approximation and when the tracker is bound to fail. On the other hand, if a heuristic system for tracking is constructed, designed to fit a specific application, one has no means to systematically determine the limits of that tracker.

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. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39(1):1–38, 1977.

[5] J. Deutscher, A. Blake, and I. Reid. Articulated motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, volume 2, pages 126–133, 2000.

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

[7] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[8] 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.

[9] 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.

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

[11] N. Gordon. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):353–358, 1997.

[12] 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.

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

[14] C. Hue, J-P. Le Cadre, and P. Pérez. The (MR)MTPF: Particle filters to track multiple targets using multiple receivers. In *International Conference on Information Fusion*, volume 2, pages FrC3:33–40, 2001.

[15] 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.

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

[17] 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.

[18] K. Kastella. Joint multitarget probabilities for detection and tracking. In *SPIE Conference on Acquisition, Tracking and Pointing*, volume 3086, pages 122–128, 1997.

[19] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

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

[21] R. Mahler. The search for tractable Bayesian multitarget filters. In *SPIE Conference on Signal and Data Processing of Small Targets*, volume 4048, pages 310–320, 2000.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

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

[28] B. Pitman and D. Tenne. Tracking a convoy of ground vehicles. Technical report, University of Buffalo, Buffalo, NY, USA, 2002.

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

[30] D. J. Salmond and N. J. Gordon. Group and extended object tracking. In *SPIE Conference on Signal and Data Processing of Small Targets*, volume 3809, pages 284–296, 1999.

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

[32] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *European Conference on Computer Vision, ECCV*, volume 2, pages 702–718, 2000.

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

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