Signal Processing 97 (2014) 207–220

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Hierarchical particle filtering for multi-modal data fusion with application to multiple-target tracking ☆

Phani Chavali *, Arye Nehorai

Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, One Brookings Drive, St. Louis, MO 63130, USA

ARTICLE INFO

Article history:
Received 24 January 2013
Received in revised form
14 October 2013
Accepted 15 October 2013
Available online 30 October 2013

Keywords:
Sensor network
Multitarget tracking
Multi-modal sensors
Data fusion
Independent likelihood
Sequential Bayesian filtering

ABSTRACT

We propose a sequential and hierarchical Monte Carlo Bayesian framework for state estimation using multi-modal data. The proposed hierarchical particle filter (HPF) estimates the global filtered posterior density of the unknown state in multiple stages, by partitioning the state space and the measurement space into lower dimensional subspaces. At each stage, we find an estimate of one partition using the measurements from the corresponding partition, and the information from the previous stages. We demonstrate the proposed framework for joint initiation, termination and tracking of multiple targets using multi-modal sensors. Here, the multi-modal data consists of the measurements collected from a radar, an infrared camera and a human scout. We compare the performance of the proposed HPF with the performance of a standard particle filter that uses linear opinion (SPF-LO), independent opinion (SPF-IO), and independent likelihood (SPF-IL) for data fusion. The results show that HPF improves the robustness of the tracking system in handling the initiation and termination of targets and provides a lower mean-squared error (RMSE) in the position estimates of the targets that maintain their tracks. The RMSE in the velocity estimates using the HPF was similar to the RMSE obtained using SPF based methods.

1. Introduction

The last few decades have seen burgeoning growth in the development of inexpensive sensing devices due to the advances in the micro-electro-mechanical system (MEMS) fabrication techniques. With the sensing devices becoming cheap, it has become possible to deploy several kinds of sensors to observe the state of the nature through the measurements collected by the sensors. Such networks ;are called multi-modal sensor networks, and they are deployed for various applications including visual tracking [1], cardiovascular diagnosis [2], 3D image reconstruction [3], temperature inference [4], and target tracking [5]. In multi-modal sensor networks, different quantities associated with the same state are measured using sensors of different kinds. A single sensor is usually unable to provide complete information about the hidden state, and therefore it is necessary to combine the information provided by multiple sensors for obtaining an estimate of the state. When complimentary information from different kinds of sensors is appropriately combined, the performance of the overall system improves significantly compared with the performance of each modality separately.

The problem of combining the diverse and sometimes inconsistent measurements provided by multiple sensors is called data fusion [6–8]. Fusing the data acquired from different sensors has to be done before the estimation process and therefore, it is a critical bottleneck. There are several data fusion algorithms that are proposed in the

☆ This work was supported by the Department of Defense under the AFOSR Grant FA9550-11-1-0210 and the NSF Grants CCF-1014908 and CCF-0963742.

* Corresponding author. Tel.: +1 314 935 7520; fax: +1 314 935 7500.
E-mail addresses: chavalis@ese.wustl.edu (P. Chavali), nehorai@ese.wustl.edu (A. Nehorai).

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.10.015

literature which combine the data at one of the following three levels. At the low level, the raw data obtained from each sensor is combined. This kind of data fusion is used when the employed sensors collect similar types of data. At the intermediate level, the raw data from each sensor is processed to obtain the features of interest, and the features are combined. The problem with this approach is that the process of feature selection is subjective and problem dependent. At the high level or the decision level, the posterior distributions of the unknown state vector, which are computed using the raw data from individual sensors, are combined. Graphical models [9] have been used as suitable candidates for data fusion at the high level.

After the data fusion step, the unknown state is estimated using the information that is collected by various sensors. This is accomplished in a recursive fashion, using a Markov state model to represent the state evolution, and an observation model that relates the measurements obtained by the sensors to the unknown state vector in a stochastic manner. This is called the Bayesian filtering problem [10]. A detailed survey in the area of Bayesian filtering is provided in [11].

Particle filtering [12] is a Monte Carlo approach to solve the recursive Bayesian filtering problem. Particle filters (PF) provide a tractable solution to the state estimation problem in non-linear and non-Gaussian systems, and they provide a framework for fusing the measurements collected by various sensors. A PF approximates the posterior filtered distribution using a set of particles, and associated weights, which are then propagated in time to approximate the filtered distribution in a sequential manner. The weights are updated using the principle of importance sampling [13]. Standard particle filters, based on the principle of importance sampling, suffer from a drawback called the degeneracy phenomenon. After a few iterations, the weights of all but a few particles will be close to zero. As a result of degeneracy, the number of particles contributing to the posterior distribution become significantly less over time, and hence the performance of the filter degrades. In theory, it is impossible to avoid degeneracy, but its effect can be reduced by drawing samples from a *good* proposal distribution and by using resampling techniques. Further, the number of particles required to approximate the posterior density grows exponentially [14] with the dimension of the state vector. Filters using such large number of particles are computationally complex and run into numerical inconsistencies. One of the most common ways to overcome this difficulty is to partition the state vector into several subspaces, and explore each subspace independently [15].

1.1. Formalization of the problem

Given a discrete time system characterized by a state equation and a measurement equation:

$$\boldsymbol{\theta}_t = g(\boldsymbol{\theta}_{t-1}, \boldsymbol{v}_{t-1}) \quad \text{and} \quad \boldsymbol{y}_t = h(\boldsymbol{\theta}_t, \boldsymbol{w}_t), \tag{1}$$

our goal is to estimate the unknown state vector $\boldsymbol{\theta}_t$, $t \in \mathbb{N}$, given the past measurements $\boldsymbol{y}_{1:t-1} = \{\boldsymbol{y}_1, \boldsymbol{y}_2 \ldots \boldsymbol{y}_{t-1}\}$ and the current measurement $\boldsymbol{y}_t$. In the above, $g(\cdot)$, and $h(\cdot)$ are known, but possibly nonlinear functions that characterize the state evolution and the observation model, $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are the stochastic process noise, and the stochastic measurement noise at time t, respectively. The state equation and the measurement equation can be equivalently written in terms of the conditional probability distributions, $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ and $p(\boldsymbol{y}_t|\boldsymbol{\theta}_t)$. The posterior probability distribution of the state vector given the measurements can be computed, under the Bayesian framework, recursively, using the Chapman–Kolmogorov equation and Bayes' theorem [10]:

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) p(\boldsymbol{\theta}_{t-1}|\boldsymbol{y}_{1:t-1})\, \mathrm{d}\boldsymbol{\theta}_{t-1}, \tag{2}$$

and

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = \frac{1}{z} p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t-1}), \tag{3}$$

where z is a normalization constant. Eqs. (2) and (3) are called the time-update equation and the measurement-update equation, respectively. When a multi-modal sensor network is considered, the measurements are collected by N modalities, and we denote the data collected by the nth modality using $\boldsymbol{y}_{n,t}$. $\boldsymbol{y}_t = \{\boldsymbol{y}_{1,t}, \ldots, \boldsymbol{y}_{N,t}\}$ denotes the data collected across all the modalities. Given the state $\boldsymbol{\theta}_t$, we assume that the data collected is conditionally independent across the modalities, i.e., $p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) = \prod_{n=1}^{N} p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_t)$. Our goal is to estimate the global posterior filtered density $p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})$ using particle filtering.

1.2. Related work

Fusion of information from different modalities has been a major research area and there has not been any standard technique that has been widely accepted [8]. In general the global posterior filtered density is obtained by combining the local posterior densities corresponding to different modalities. Three most commonly used methods are [16,17].

1. Linear Opinion Pool:

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = \sum_{n=1}^{N} \pi_n p(\boldsymbol{\theta}_t|\boldsymbol{y}_{n,1:t}), \tag{4}$$

   where each weight π_n represents a subjective measure of the reliability of the information from the nth modality and $\sum_{n=1}^{N} \pi_n = 1$.
2. Independent Opinion Pool:

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = \prod_{n=1}^{N} p(\boldsymbol{\theta}_t|\boldsymbol{y}_{n,1:t}). \tag{5}$$

3. Independent Likelihood Pool:

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{\theta}_t) \prod_{n=1}^{N} p(\boldsymbol{y}_{n,1:t}|\boldsymbol{\theta}_t). \tag{6}$$

In addition, the authors in [18] proposed a mixture kernel-based Bayesian filtering for multi-modal data. In this approach, the local posterior distribution corresponding to each modality is modeled as a mixture of Gaussian

distributions. The global posterior distribution is then represented as a mixture of the local posterior distributions, and this representation is preserved through iterations of the Bayesian filtering. The details of this method can be found in [1,18,19].

1.3. Our contributions

The contributions of this paper are three fold. First, we derive the weight update equations for a particle filter that employs (i) linear opinion pool, (ii) independent opinion pool and (iii) independent likelihood pool for combining the information from the various sensing modalities. In all these three techniques, the information from the multiple sensing modalities is combined only in the measurement-update Eq. (3).

Second, we propose a new particle filtering technique for a special class of models called the hierarchical models. We call this variation of the particle filter, a *hierarchical particle filter* (HPF). Hierarchical models are encountered frequently in practice when the observations about the unknown state are obtained using multi-modal sensors, where each sensor is used to obtain information about a specific aspect of the state vector. We will provide more examples of such models in the subsequent section. In our approach, we partition the state space and the measurement space into lower-dimensional subspaces, and approximate the local posterior distribution corresponding to the first partition, given the measurements from the first partition, using a random particle set and the associated weights. To approximate the subsequent partitions, we generate the particles using the information obtained from the estimation of previous partitions, and update the weights using the measurements from the current partition. Thus, unlike the earlier approaches, we combine information from different modalities in both the time- and the measurement-update steps. We also describe the process of obtaining the particles from the proposal distribution, and updating their corresponding weights. Our proposed filtering method provides two advantages when compared to the other methods that are described in the previous subsection. First, by partitioning the state space into several subspaces, we reduce the effect of the degeneracy phenomenon. Second, since the proposal distribution for each stage is chosen using the information from the previous stages, sampling is more efficient and thereby, the particle weights will have a lower variance.

Third, we consider an example of multiple-target tracking using multi-modal sensors using the proposed hierarchical filtering for the joint initiation, termination and tracking of multiple targets. We employ a multi-modal sensor network comprising a multistatic radar, an infrared camera, and a human scout, to obtain the information about the target scene. The unknown state vector comprises the number of targets, their positions, velocities, and categories. In [20], we presented some preliminary results demonstrating the performance benefits obtained by the proposed hierarchical particle filtering for target tracking application. In the current paper, we describe the detailed state model for this system, and derive the measurement models for the three sensors that are used. We also provide numerical examples to compare the performance obtained using the proposed data fusion method with the performance obtained using other methods that fuse information from multi-modal sensors. The framework provided in this paper is more general than the specific approach taken in [20] to solve the tracking problem, it formulates how the proposed method can be extended to other applications that employ multi-modal sensors.

1.4. Organization and notations

Organization: The rest of the paper is organized as follows. In Section 2, we describe the hierarchical Bayesian model that we consider in this paper. In Section 3, we describe the particle filtering for the state estimation of non-linear systems and then describe the proposed hierarchical particle filter. We describe the choice of the proposal distributions for sampling each partition of the state vector, and derive the corresponding weight update equations. In Section 4, we provide two numerical examples to demonstrate the performance advantage of the proposed HPF. We use synthetic data sets in these examples. In Section 5, an application of the proposed filtering method is fully addressed. We consider the problem of joint initiation, termination and tracking of multiple targets moving in a region of interest. A multi-modal sensor network comprising a multistatic radar, an infrared camera, and a human scout, is deployed in the surveillance region, to obtain the information about the target scene. The objective of the sensor network is to combine the data obtained by each sensor in order to track the number of targets in the region, their positions, their velocities and their categories. In Sections 5.2 and 5.3, we describe the state model, and the statistical measurement models of the sensors that we use, in Section 5.4, we provide the results for the target tracking, and compare the performance of the tracking system to the performance obtained by using particle filtering with linear opinion, independent opinion, and independent likelihood methods for data fusion. We define four metrics to evaluate the performance of the tracking system. We summarize the paper in Section 6.

Notation: We use the following notations in the paper. We denote vectors by boldface lowercase letters, e.g., $\boldsymbol{a}$, and matrices by boldface uppercase letters, e.g., $\boldsymbol{A}$. For a matrix $\boldsymbol{A}$, we use $\boldsymbol{a}_i$ to represent the ith column of $\boldsymbol{A}$ and $[\boldsymbol{A}]_{ij}$ to represent the element in the ith row and the jth column. $(\boldsymbol{A})^T, (\boldsymbol{A})^H$, and vec($\boldsymbol{A}$) denote the transpose, conjugate transpose and vector form of the matrix $\boldsymbol{A}$, respectively. The ith element of a vector $\boldsymbol{a}$ is denoted by a_i. The Kronecker product of two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, is denoted as $\boldsymbol{A} \otimes \boldsymbol{B}$. $\boldsymbol{I}_M$ and $\boldsymbol{0}_{M\times N}$ denote an identity matrix of order M and a zero matrix of size $M \times N$, respectively. $*$ denotes the convolution operator. $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ and $\mathbb{C}\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ denote a normal distribution and a complex normal distribution in variable $\boldsymbol{x}$, with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, respectively. $\mathbb{R}$ denotes the set of all real numbers and $\mathbb{N}$ denotes the set of all natural numbers.

2. Hierarchical Bayesian models

In this paper, we consider a class of systems in which the state equation and the measurement equation follow a hierarchical structure. The evolution model and the measurement model for such a system can be described by the following equations:

$$\begin{aligned}
&\boldsymbol{\theta}_{1,t} = g_1(\boldsymbol{\theta}_{1,t-1}, \boldsymbol{v}_{1,t-1}); \quad \boldsymbol{y}_{1,t} = h_1(\boldsymbol{\theta}_{1,t}, \boldsymbol{w}_{1,t}),\\
&\boldsymbol{\theta}_{2,t} = g_2(\boldsymbol{\theta}_{2,t-1}, \boldsymbol{\theta}_{1,t}, \boldsymbol{v}_{2,t-1}); \quad \boldsymbol{y}_{2,t} = h_2(\boldsymbol{\theta}_{1,t}, \boldsymbol{\theta}_{2,t}, \boldsymbol{w}_{2,t}),\\
&\vdots \qquad \vdots\\
&\boldsymbol{\theta}_{N,t} = g_N(\boldsymbol{\theta}_{N,t-1}, \boldsymbol{\theta}_{1:N-1,t}, \boldsymbol{v}_{N,t-1}); \quad \boldsymbol{y}_{N,t} = h_N(\boldsymbol{\theta}_{1:N,t}, \boldsymbol{w}_{N,t}).
\end{aligned} \tag{7}$$

In the above, $\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{N,t}$ are the N partitions of the state vector, $\boldsymbol{\theta}_{1:n,t} = \{\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{n,t}\}$, and $\boldsymbol{y}_{1,t}, \ldots, \boldsymbol{y}_{N,t}$ are the partitions of the measurements. Each partition corresponds to the measurements obtained from one modality. Thus, it can be seen that the first modality observes only one partition of the state vector, the second modality observes two partitions and finally the last modality observes the entire state vector. Further, the evolution of each partition of the state vector depends on the previous partitions.

Multi-modal systems are typically characterized by such hierarchical observation and state models, because each sensing modality is selected to measure a specific aspect of the unknown state vector. For example, in surveillance systems that use a combination of audio and video sensors, the audio sensors determine the direction of arrival of the source, whereas the video sensors measure the number of sources in the region, and their positions. The measurements obtained by the audio sensors are, however, a function of the number of the sources and their positions. In the target tracking example that we consider, the human scout is trained to estimate the number of the targets and to recognize their categories, but he is incapable of determining their positions and velocities; the infrared camera sensors can obtain estimate of the target locations, but they cannot determine their velocities; a multistatic radar can measure both the positions and velocities of the targets accurately, but it has limited capabilities in measuring the number of targets.

In some scenarios, each sensor might not be able to sense all the partitions of the state (see Example 2). Hierarchical modeling shown in Eq. (7) above will not describe such models. The proposed method can still be applied to such models, however the performance improvement obtained will not be significant.

3. Hierarchical Bayesian filtering

In this section, we describe the hierarchical approach for obtaining the Bayesian estimate of the state vector using Monte-Carlo sampling. Monte-Carlo methods are based on approximating the posterior distribution using a set of particles. We first describe the particle filter as it was described in [12] and we refer to it as standard particle filter. Next, we describe the proposed hierarchial particle filter that operates by partitioning the state-space and the measurement space into several subspaces. Each partition of the state vector is then estimated in an hierarchical fashion, where the information from one stage is used to estimate the subsequent partitions.

3.1. Standard particle filter

The particle filter computes a discrete weighted approximation to the true posterior density using a set of particles and their associated weights that characterize the posterior probability distribution. For a scenario where all the observations, $\boldsymbol{z}_{1:t} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t\}$, come from a single modality, the posterior distribution can be approximated as

$$p(\boldsymbol{\theta}_{0:t}|\boldsymbol{z}_{1:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \delta(\boldsymbol{\theta}_{0:t} - \boldsymbol{\theta}_{0:t}^{(i)}), \tag{8}$$

where $\{\boldsymbol{\theta}_{0:t}^{(i)}\}_{i=1}^{N_s}$ are the particles (or samples) that characterize the probability distribution $p(\boldsymbol{\theta}_{0:t}|\boldsymbol{z}_{1:t})$, and $\{w_t^{(i)}\}_{i=1}^{N_s}$ are the associated weights. The samples $\{\boldsymbol{\theta}_{0:t}^{(i)}\}_{i=1}^{N_s}$ are drawn from a known proposal distribution, and the weights are derived using the principle of importance sampling [21]. According to this principle, the weights are updated using

$$\tilde{w}_t^{(i)} = \frac{p(\boldsymbol{\theta}_{0:t}^{(i)}|\boldsymbol{z}_{1:t})}{q(\boldsymbol{\theta}_{0:t}^{(i)}|\boldsymbol{z}_{1:t})}, \tag{9}$$

where $q(\boldsymbol{\theta}_{0:t}|\boldsymbol{z}_{1:t})$ is the proposal distribution from which the samples are drawn and $\tilde{w}_t^{(i)}$ is the un-normalized weight of the ith particle at time t. If the observations come from multiple modalities, the posterior distributions should be combined using one of the three methods that we described in Section 1.2. Below, we derive the weight update for each of these cases.

3.1.1. Linear opinion pool

The posterior distribution $p(\boldsymbol{\theta}_{0:t}|\boldsymbol{y}_{1:t})$ is expressed as

$$p(\boldsymbol{\theta}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{n=1}^{N} \pi_n \sum_{i=1}^{N_s} w_{n,t}^{(i)} \delta(\boldsymbol{\theta}_{0:t} - \boldsymbol{\theta}_{0:t}^{(i)}), \tag{10}$$

where $w_{n,t}^{(i)}$ is the weight corresponding to the ith particle, and nth modality. The weights corresponding to each modality are updated using the principle of importance sampling.

$$\begin{aligned}
\tilde{w}_{n,t}^{(i)} &\propto \frac{p(\boldsymbol{\theta}_{0:t}^{(i)}|\boldsymbol{y}_{n,1:t})}{q(\boldsymbol{\theta}_{0:t}^{(i)}|\boldsymbol{y}_{n,1:t})}\\
&\propto \frac{p(\boldsymbol{\theta}_{0:t-1}^{(i)}|\boldsymbol{y}_{n,1:t-1})}{q(\boldsymbol{\theta}_{0:t-1}^{(i)}|\boldsymbol{y}_{n,1:t-1})} \times p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_t^{(i)}) \times \frac{p(\boldsymbol{\theta}_t^{(i)}|\boldsymbol{\theta}_{t-1}^{(i)})}{q(\boldsymbol{\theta}_t^{(i)}|\boldsymbol{\theta}_{0:t-1}^{(i)}, \boldsymbol{y}_{n,1:t})}\\
&\propto w_{n,t-1}^{(i)} \times p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_t^{(i)}) \times \frac{p(\boldsymbol{\theta}_t^{(i)}|\boldsymbol{\theta}_{t-1}^{(i)})}{q(\boldsymbol{\theta}_t^{(i)}|\boldsymbol{\theta}_{0:t-1}^{(i)}, \boldsymbol{y}_{n,1:t})}\\
&\propto w_{n,t-1}^{(i)} \times p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_t^{(i)}),
\end{aligned} \tag{11}$$

where the proposal distribution $q(\boldsymbol{\theta}_{0:t}|\boldsymbol{y}_{n,1:t})$ is chosen to factorize such that

$$q(\boldsymbol{\theta}_{0:t}|\boldsymbol{y}_{n,1:t}) = q(\boldsymbol{\theta}_{0:t-1}|\boldsymbol{y}_{n,1:t-1}) \times q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \boldsymbol{y}_{n,1:t}), \tag{12}$$

and $q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{0:t-1}, \boldsymbol{y}_{n,1:t}) = q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{y}_{n,t})$. Due to this choice, the proposal distribution only depends on $\boldsymbol{\theta}_{t-1}$, and we can discard the samples $\boldsymbol{\theta}_{0:t-1}$, and the observations $\boldsymbol{z}_{1:t-1}$.

The last step of Eq. (11) is obtained by choosing the proposal distribution $q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1},\boldsymbol{y}_{n,t})$ to be the state transitional prior $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$. Note that the proposal distribution is independent of the sensor index n. Thus using the linear opinion pool, we update the weights due to each modality first, and then obtain the overall weights as

$$w_t^{(i)} = \sum_{n=1}^{N} \pi_n w_{n,t}^{(i)}. \tag{13}$$

We refer to this filtering method as standard particle filtering with linear opinion pool (SPF-LO).

3.1.2. Independent opinion pool

Using Eq. (5), the global posterior distribution for the independent opinion pool is written as

$$p(\boldsymbol{\theta}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{i=1}^{N_s} \prod_{n=1}^{N} w_{n,t}^{(i)} \delta(\boldsymbol{\theta}_{0:t} - \boldsymbol{\theta}_{0:t}^{(i)}). \tag{14}$$

Following Eq. (14), it can be seen that the overall weights for standard particle filtering with independent opinion pool (SPF-IO) are given by

$$\tilde{w}_t^{(i)} \propto \prod_{n=1}^{N} \tilde{w}_{n,t}^{(i)}, \tag{15}$$

where $\{\tilde{w}_{n,t}^{(i)}\}_{i=1}^{N_s}$ are the un-normalized weights obtained by following Eq. (11).

3.1.3. Independent likelihood pool

Using Eq. (6), the global posterior distribution for the independent likelihood pool is written as

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = p(\boldsymbol{\theta}_t|y_{1:t-1}) \prod_{n=1}^{N} p(\boldsymbol{y}_{n,1:t}|\boldsymbol{\theta}_t). \tag{16}$$

Substituting Eq. (16) into the weight update equation defined by Eq. (9), it can be shown that the overall weights for standard particle filtering independent likelihood pool (SPF-IL) are given as

$$\tilde{w}_t^{(i)} \propto w_{t-1}^{(i)} \prod_{n=1}^{N} p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_t^{(i)}). \tag{17}$$

It should be noted here that for all the above methods, particle filters corresponding to all the sensors have the same set of particles.

3.2. Hierarchical particle filter

In this section, we describe the proposed hierarchical particle filter. The idea behind the hierarchical particle filter is to exploit the model structure in order to improve the efficiency of the sampling, thereby reducing the variance in the particle weights. The filter operates by using the information obtained at each stage to compute the estimates of the partitions of the state vector in the subsequent stages.

Let $\boldsymbol{\theta}_{1,t},\ldots,\boldsymbol{\theta}_{N,t}$ be the N partitions of the state vector. The measurement partitions are labeled as $\boldsymbol{y}_{1,t},\boldsymbol{y}_{2,t},\ldots,\boldsymbol{y}_{N,t}$. We find the posterior distribution of each partition individually, by using the information provided from the estimates of the previous partitions and the measurements associated with the current partition. In order to do this, we simplify the posterior distribution of the nth partition as follows:

$$\begin{aligned} p(\boldsymbol{\theta}_{n,t}|\boldsymbol{y}_{1:n,1:t}) &= \int p(\boldsymbol{\theta}_{n,t}|\boldsymbol{y}_{1:n,1:t},\boldsymbol{\theta}_{1:n-1,t},\boldsymbol{\theta}_{1:n,1:t-1}) \\ &\quad \times p(\boldsymbol{\theta}_{1:n-1,t},\boldsymbol{\theta}_{1:n,1:t-1}|\boldsymbol{y}_{1:n,1:t}) \\ &\quad \times \mathrm{d}\boldsymbol{\theta}_{1:n-1,t}\,\mathrm{d}\boldsymbol{\theta}_{1:n,1:t-1} \\ &\propto \int p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_{1:n,t}) \times p(\boldsymbol{\theta}_{n,t}|\boldsymbol{\theta}_{1:n-1,t},\boldsymbol{\theta}_{n,t-1}) \\ &\quad \times p(\boldsymbol{\theta}_{1:n-1,t},\boldsymbol{\theta}_{1:n,1:t-1}|\boldsymbol{y}_{1:n-1,t},\boldsymbol{y}_{1:n,1:t-1}) \\ &\quad \times \mathrm{d}\boldsymbol{\theta}_{1:n-1,t}\,\mathrm{d}\boldsymbol{\theta}_{1:n,1:t-1} \end{aligned} \tag{18}$$

We use an approximation for the joint posterior distribution $p(\boldsymbol{\theta}_{1:n-1,t},\boldsymbol{\theta}_{1:n,1:t-1}|\boldsymbol{y}_{1:n-1,t},\boldsymbol{y}_{1:n,1:t-1})$:

$$\begin{aligned} &p(\boldsymbol{\theta}_{1:n-1,t},\boldsymbol{\theta}_{1:n,1:t-1}|\boldsymbol{y}_{1:n-1,t},\boldsymbol{y}_{1:n,1:t-1}) \\ &\quad \approx \sum_{i=1}^{N_s} w_{n,t-1}^{(i)} \delta(\boldsymbol{\theta}_{1:n,1:t-1} - \boldsymbol{\theta}_{1:n,1:t-1}^{(i)}) \\ &\qquad \times \prod_{m=1}^{n-1} \sum_{i=1}^{N_s} w_{m,t}^{i} \delta(\boldsymbol{\theta}_{m,t} - \boldsymbol{\theta}_{m,t}^{(i)}). \end{aligned} \tag{19}$$

Substituting the approximation from Eq. (19) into Eq. (18) and rewriting, we get

$$\begin{aligned} p(\boldsymbol{\theta}_{n,t}^{(i)}|\boldsymbol{y}_{1:n-1,t},\boldsymbol{y}_{1:n,1:t-1}) &\approx w_{n,t-1}^{(i)} \prod_{m=1}^{n-1} \\ &\quad \times w_{m,t}^{i_m} p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_{n,t}^{(i)},\boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})}) \\ &\quad \times p(\boldsymbol{\theta}_{n,t}^{(i)}|\boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})},\boldsymbol{\theta}_{n,t-1}^{(i)}), \end{aligned} \tag{20}$$

where $\boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})} = [\boldsymbol{\theta}_{1,t}^{(i_1)},\boldsymbol{\theta}_{2,t}^{(i_2)},\ldots,\boldsymbol{\theta}_{n-1,t}^{(i_{n-1})}]^T$.

Using Eq. (20), the weight update for the nth stage can be obtained as

$$\begin{aligned} \tilde{w}_{n,t}^{(i)} &\propto \frac{p(\boldsymbol{\theta}_n^{(i)}|\boldsymbol{y}_{1:n,1:t})}{q_n(\boldsymbol{\theta}_{n,t}^{(i)}|\boldsymbol{y}_{1:n,1:t})}, \\ &\propto w_{n,t-1}^{(i)} \times \frac{p(\boldsymbol{\theta}_{n,t}^{(i)}|\boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})},\boldsymbol{\theta}_{n,t-1}^{(i)})}{q_n(\boldsymbol{\theta}_{n,t}^{(i)}|\boldsymbol{y}_{1:n,1:t}^{(i)})} \\ &\quad \times p(\boldsymbol{y}_{n,t}|\boldsymbol{\theta}_{n,t}^{(i)},\boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})}) \prod_{m=1}^{n-1} w_{m,t}^{(i_m)}. \end{aligned} \tag{21}$$

3.3. Sampling from the proposal distribution

The choice of the proposal distribution is crucial and specific to the proposed hierarchical model. For optimal efficiency, it is necessary that the distribution propose most probable particles. In this work, we choose the proposal distribution for the nth stage, $n>1$, to be

$$q_n(\boldsymbol{\theta}_{n,t}|\boldsymbol{y}_{1:n,1:t}) = \prod_{m=1}^{n-1} w_{m,t}^{(i_m)} p(\boldsymbol{\theta}_{n,t}^{(i)}|\boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})},\boldsymbol{\theta}_{n,t-1}^{(i)}). \tag{22}$$

We make a comment here that sampling the particles from the above proposal distribution is different than sampling them from the distribution $p(\boldsymbol{\theta}_{1:n,t}|\boldsymbol{\theta}_{1:n,t-1})$, even for hierarchical models that are described in Section 2. If the distribution $p(\boldsymbol{\theta}_{1:n,t}|\boldsymbol{\theta}_{1:n,t-1})$ is used, the entire state vector is sampled jointly. We however, use the proposal distribution described by Eq. (22), in which the samples are obtained in stages. We use the information provided by the weights corresponding to the partitions from earlier stages to obtain the samples for the current stage. Intuitively, this means that we are using the posterior weights

$w_{m,t}^{(i_m)}$, $i_m = 1,\ldots,N_s$, $m = 1,\ldots,n-1$ as extrinsic information for sampling the current partition $\boldsymbol{\theta}_{n,t}$. As a result, although we are sampling the partitions individually, we are still taking into account the dependencies among them. This is in contrast to the traditional sampling schemes that either draw samples of all the partitions in one stage, or partition the state and draw the samples for each partition independently without using the information from the previous stages. In the weight update step, we use the current measurements $\boldsymbol{y}_{n,t}$ of each partition to update the corresponding weights.

In order to draw the samples from the distribution $\prod_{m=1}^{n-1} w_{m,t}^{(i_m)} p(\boldsymbol{\theta}_{n,t}^{(i)} | \boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})}, \boldsymbol{\theta}_{n,t-1}^{(i)})$, we use the following steps:

1. Choose N_s samples of $\boldsymbol{\theta}_{1,t},\ldots,\boldsymbol{\theta}_{n-1,t}$, generated from the earlier $N-1$ stages.
2. Choose, for each partition, the samples $\{\boldsymbol{\theta}_{m,t}^{(i_m)}\}_{i_m=1}^{N_s}$, $m = 1,\ldots,n-1$, with replacement, according to the probability distribution defined by the weights $\{w_{m,t}^{(i)}\}_{i=1}^{N_s}$, $m = 1,\ldots,n-1$.
3. Generate the samples $\{\boldsymbol{\theta}_{n,t}^{(i)}\}_{i=1}^{N_s}$ according to the known state transition probability $p(\boldsymbol{\theta}_{n,t} | \boldsymbol{\theta}_{1:n-1,t}, \boldsymbol{\theta}_{n,t-1})$.

As a result of the second step, the samples with higher weights are selected more frequently. Due to this selection, there is a lower probability of choosing a sample of other partitions that will lead to an incorrect sample of the current partition. Therefore, the weights associated with the samples generated using this approach will have a lower variance. Further, since we are choosing the samples according to the weights $\{w_{m,t}^{(i)}\}_{i=1}^{N_s}$, $m = 1,\ldots,n-1$, the weight update equation simplifies to

$$\tilde{w}_{n,t}^{(i)} \propto w_{n,t-1}^{(i)} \times p(\boldsymbol{y}_{n,t} | \boldsymbol{\theta}_{n,t}^{(i)}, \boldsymbol{\theta}_{1:n-1,t}^{(i_1,\ldots,i_{n-1})}). \tag{23}$$

In order to obtain Eq. (23), we set $w_{m,t}^{(i_m)} = (1/N_s)$ $i_m = 1,\ldots,N_s$, $m = 1,\ldots,n-1$ since the particles are chosen according to these weights already.

In this manner, we obtain a sampling algorithm that operates in stages. We split the unknown state vector into partitions, and sample the individual partitions rather than sampling the overall state jointly. In the first stage of the algorithm, we estimate the first partition of the state vector using the measurements from the first partition using a standard particle filter. The particles are drawn from the proposal distribution $q_1(\boldsymbol{\theta}_{1,t} | \boldsymbol{y}_{1,1:t})$, which is chosen to be $q_1(\boldsymbol{\theta}_{1,t} | \boldsymbol{y}_{1,1:t}) = p(\boldsymbol{\theta}_{1,t} | \boldsymbol{\theta}_{1,t-1})$, the state transition distribution corresponding to the first partition. In the subsequent stages of the algorithm, the posterior weights $\{w_{m,t}^{i_m}\}, i_m = 1,\ldots,N_s$, $m = 1,\ldots,n-1$ are used as extrinsic information to draw the samples and update the corresponding weights of the nth partition of the state vector. Similar approaches have been used in [22,23], in the context of multiple-target tracking where the association between the measurements and targets is unknown. In these works, the state vector corresponding to each target is sampled individually and independently. However, if the partitions are not truly independent (which will be the case since the association between a target and the measurements is usually dependent on the associations between the other targets and the measurements), as N_s increases, the approximation given in Eq. (19), may not become exact. However, the performance obtained by such a sampling is better as it reduces the variance of the weights [22]. The advantage of such independent sampling

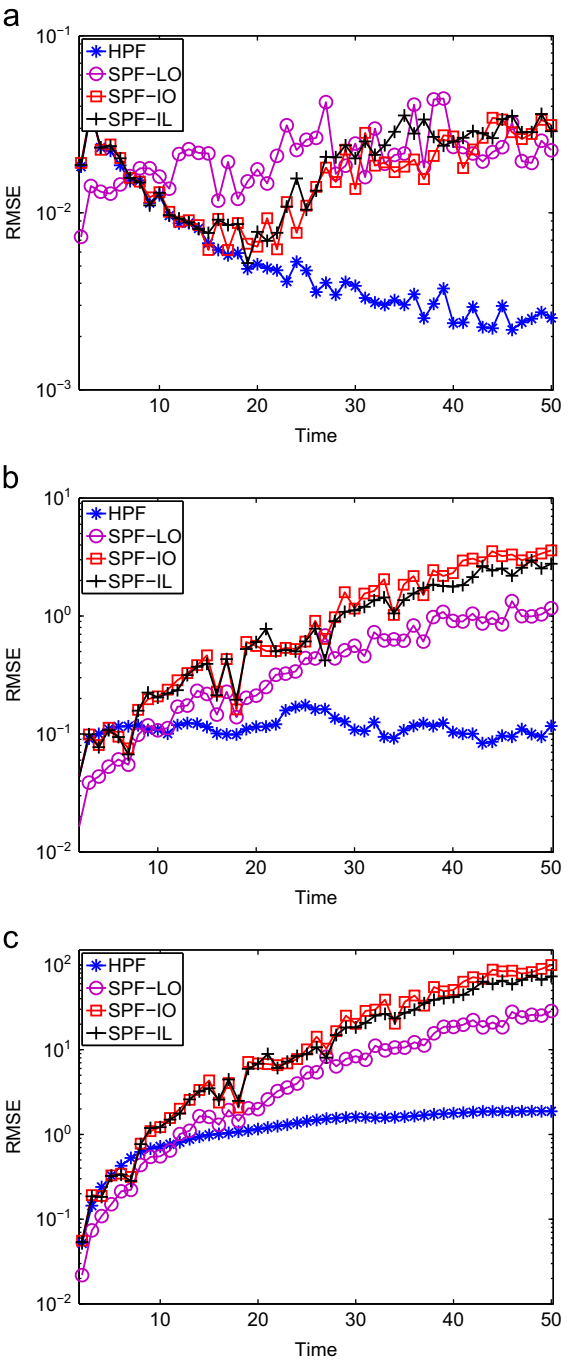

Fig. 1. RMSE for the three partitions for the first example: (a) RMSE – θ_1; (b) RMSE – θ_2; and (c) RMSE – θ_3.

approach is that all the partitions can be simultaneously sampled. In our approach, the sampling is done sequentially, as we use information from one stage to sample the state partitions in subsequent stages.

3.4. Resampling

The next step after updating the weights associated with all the particles is resampling. The idea of resampling is to eliminate the particles that have smaller weights and replace them with new samples with equal weights. Resampling is a crucial step in particle filtering, since after a few iterations, propagating trivial weights becomes computationally expensive. After each iteration, the effective number of particles is computed using [12]:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_s}(w_t^{(i)})^2}, \tag{24}$$

and the particle set is resampled if N_{eff} falls below a predetermined threshold. There are several resampling techniques available in the literature [11]. In this paper, we use the residual systematic resampling technique that is described in [24].

4. Numerical examples

In this section, we present two examples to describe the performance of the proposed hierarchical filtering method. We consider a state vector which is partitioned into three subspaces. The measurements are collected by three modalities. Each partition of the state vector is of unit dimension, and the corresponding measurements are scalars.

4.1. First example

In the first example, the scalar probability distributions governing the state equation and the measurement equations for each partition are given as

$$p(y_{1,t}|\theta_{1,t}) \sim \mathcal{N}\left(y_{1,t}; \theta_{1,t}^2, \sigma_{y_1}^2\right)$$

$$p(\theta_{1,t}|\theta_{1,t-1}) \sim \mathcal{N}\left(\theta_{1,t}; 1+\theta_{1,t-1}, \sigma_{\theta_1}^2\right)$$

$$p(y_{2,t}|\theta_{1,t},\theta_{2,t}) \sim \mathcal{N}\left(y_{2,t};\ \sin(\theta_{1,t}) + \cos(\theta_{2,t}), \sigma_{y_2}^2\right)$$

$$p(\theta_{2,t}|\theta_{2,t-1},\theta_{1,t}) \sim \mathcal{N}\left(\theta_{2,t}; \theta_{2,t-1}+\theta_{1,t}, \sigma_{\theta_2}^2\right)$$

$$p(y_{3,t}|\theta_{1,t},\theta_{2,t},\theta_{3,t}) \sim \mathcal{N}\left(y_{3,t}; \frac{\theta_{1,t}^2+\theta_{2,t}^2+\theta_{3,t}^2}{(\theta_{1,t}+\theta_{2,t}+\theta_{3,t})^2}, \sigma_{y_3}^2\right)$$

$$p(\theta_{3,t}|\theta_{3,t-1},\theta_{2,t},\theta_{1,t}) \sim \mathcal{N}(\theta_{3,t}; \theta_{3,t-1}+\theta_{2,t}+\theta_{1,t}, \sigma_{\theta_3}^2), \tag{25}$$

where $\sigma_{y_1}^2 = \sigma_{y_2}^2 = \sigma_{y_3}^2 = 0.2$, and $\sigma_{\theta_1}^2 = \sigma_{\theta_2}^2 = \sigma_{\theta_3}^2 = 0.1$. We used $N_s = 200$ particles for the simulation and averaged the result for $N_c = 50$ Monte-Carlo iterations. In Fig. 1, we plot the root mean-squared error (RMSE) for the three partitions. It can be seen that the HPF produces a lower RMSE when compared to the RMSE obtained using SPF-LO, SPF-IO and SPF-IL.

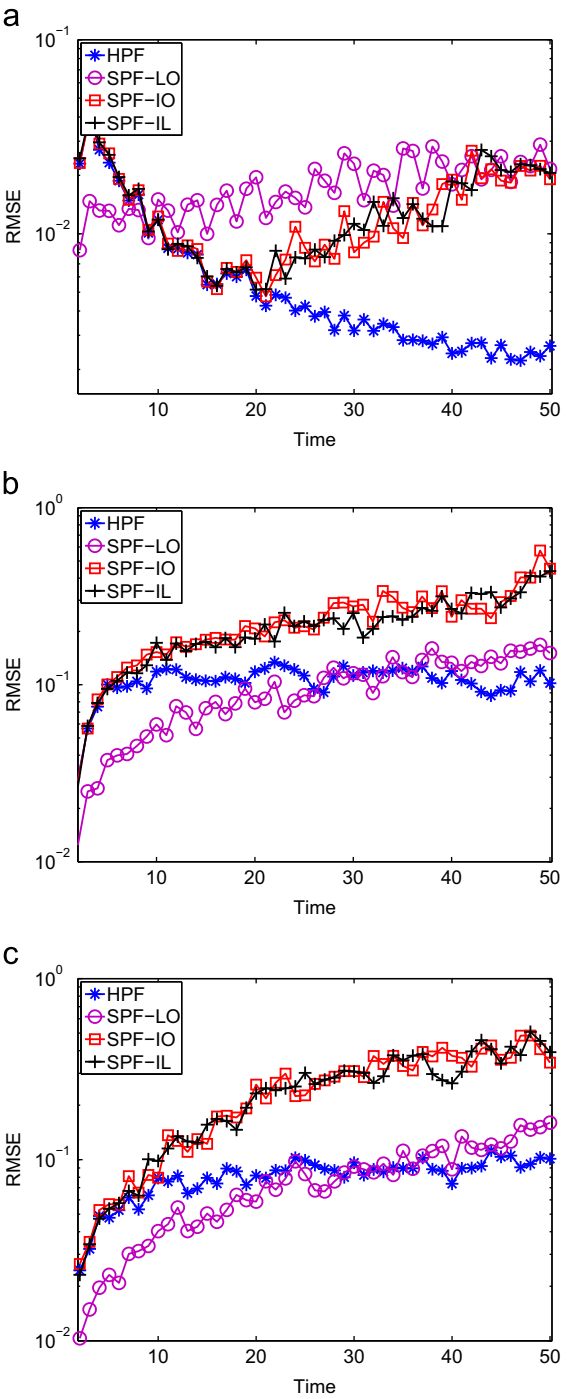

Fig. 2. RMSE for the three partitions for the second example: (a) RMSE – θ_1; (b) RMSE – θ_2; and (c) RMSE – θ_3.

4.2. Second example

In the second example, we analyze the behavior of algorithm when the hierarchy in the state equations is not satisfied. To this end, we modify the probability distributions that govern the state evolution as

$$p(\theta_{1,t}|\theta_{1,t-1}) \sim \mathcal{N}(\theta_{1,t}; 1+\theta_{1,t-1}, \sigma_{\theta_1}^2)$$

$$p(\theta_{2,t}|\theta_{2,t-1},\theta_{1,t}) \sim \mathcal{N}(\theta_{2,t};\theta_{2,t-1},\sigma^2_{\theta_2})$$
$$p(\theta_{3,t}|\theta_{3,t-1},\theta_{2,t},\theta_{1,t}) \sim \mathcal{N}(\theta_{3,t};\theta_{3,t-1},\sigma^2_{\theta_3}), \quad (26)$$

The probability distributions corresponding to the three measurements were the same as the distributions that were used in the first example, and the variances corresponding to the measurement noise and the process noise were the same as the ones used in the first example. In Fig. 2, we plot the RMSE for all the methods for the second example. It can be seen that the RMSE in the estimates of the partitions using the HPF was lower than the RMSE obtained using SPF-IO, and SPF-IL, but it was similar to the RMSE obtained using the SPF-LO.

5. Application to multiple-target tracking

In this section, we apply the proposed algorithm for the problem of multiple target tracking [5,25,26]. The ability to track multiple targets is essential in several military and commercial applications such as air traffic control and battlefield surveillance. This problem poses two major challenges. First, the number of targets is unknown and the tracking system should be capable of automatically determining this number. Second, the state space model and the sensor measurement model are non-linear. To overcome these problems, a particle filter with observations from a multi-modal sensor network is used. Earlier works using multi-modal information for the problem of target tracking include [27–29]. In this paper, we consider a network comprising three kinds of sensors: a multistatic radar with one transmit and three receive antennas, which can measure the backscattered signal from the targets; one infrared camera, which can obtain the top view of the region using arial shots; and an intelligence report provided by a human scout.

5.1. System model

We consider a planar region of the battlefield, $\mathcal{R}$, with an unknown number of moving targets of various types. At time t, we assume that the number of targets is N_t. The targets are indexed as $\{1,\ldots,n_t,\ldots,N_t\}$ with the position and velocity of n_tth target denoted as $\boldsymbol{\rho}_{n_t}=[\rho_{x,n_t},\rho_{y,n_t}]^T \in \mathbb{R}^2$ and $\dot{\boldsymbol{\rho}}_{n_t}=[\dot{\rho}_{x,n_t},\dot{\rho}_{y,n_t}]^T \in \mathbb{R}^2$, respectively. We label the target categories using $\boldsymbol{\alpha}=\{\alpha_1,\ldots,\alpha_{n_t},\ldots,\alpha_{N_t}\}$, where $\alpha_{n_t} \in \mathcal{A}$, with $\mathcal{A}$ being the set of all the target types. The parameters of interest are the number of targets N_t in the scene, the positions $\boldsymbol{\rho}_{n_t}$, the velocities $\dot{\boldsymbol{\rho}}_{n_t}$, and the categories α_{n_t} of each of the targets. The overall state vector at time t is obtained by concatenating all the unknown parameters and we denote it using $\boldsymbol{\theta}_t=[N_t,\boldsymbol{\rho}_1^T,\dot{\boldsymbol{\rho}}_1^T,\alpha_1,\ldots,\boldsymbol{\rho}_{N_t}^T,\dot{\boldsymbol{\rho}}_{N_t}^T,\alpha_{N_t}]^T \in (\mathbb{R}^2\times\mathbb{R}^2\times\mathcal{A})^{N_t}$. Our goal is to use the partial and complementary information obtained from different sensors to track the state vector over a period of time. In particular, we are interested in obtaining an estimate of $\boldsymbol{\theta}_t$ at time t. In the following subsections, we derive the state transition model for the vector $\boldsymbol{\theta}_t$, and develop the statistical measurement models for the sensors used.

We would like to emphasize here that in this paper, we do not consider the problems of the target labeling and measurement association ambiguity. We assume that the tracker can estimate the target labels, and the association vectors at each time instant. This assumption allows for a greatly simplified model and notations which helps the reader understand the intuition behind the hierarchical modeling. The generalization of the problem to the case where the target labels and measurement associations are unknown is left to the readers. Reference [30] provides a detailed modeling required to address the problem of target labeling and [31] describes the Monte Carlo modeling used to address the measurement association ambiguities.

5.2. State-space model

We assume, for simplicity, that there can be at most one birth or one death of the targets at each state transition, and we represent the probabilities of the death and the birth of the targets using p_d and p_b, respectively. Hence, we have

$$p(N_{t+1}|N_t=n_t)=\begin{cases} p_b & \text{if } N_{t+1}=n_t+1,\\ n_t p_d(1-p_d)^{n_t-1} & \text{if } N_{t+1}=n_t-1,\\ 1-p_b-n_t p_d(1-p_d)^{n_t-1} & \text{if } N_{t+1}=n_t. \end{cases} \quad (27)$$

Let $\boldsymbol{\alpha}_t=\{\alpha_{t,1},\alpha_{t,2},\ldots,\alpha_{t,N_t}\}$ denote the categories of each of the N_t targets, and assume that the number of possible categories is finite, i.e., $\text{card}(\mathcal{A})=M<\infty$. Let $\alpha^* \in \mathcal{A}$ be the category of the new target that appears at time $t+1$. The probability distribution for the target categories $\boldsymbol{\alpha}_{t+1}$ at time $t+1$ given $\boldsymbol{\alpha}_t$ and N_{t+1} can be written as

$$p\left(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t,N_{t+1}\right)=\begin{cases} \dfrac{1}{M} & \text{if } N_{t+1}=N_t+1,\boldsymbol{\alpha}_{t+1}=\boldsymbol{\alpha}_t\cup\alpha^*,\\ \dfrac{1}{N_t} & \text{if } N_{t+1}=N_t-1,\boldsymbol{\alpha}_{t+1}=\boldsymbol{\alpha}_t-\alpha_{t,n_t} \quad \text{for any } n_t,\\ \dfrac{p_d p_b}{N_t M} & \text{if } N_{t+1}=N_t,\boldsymbol{\alpha}_{t+1}=\boldsymbol{\alpha}_t-\alpha_{t,n_t}\cup\alpha^*,\\ 1-\dfrac{p_d p_b}{N_t M} & \text{if } N_{t+1}=N_t,\boldsymbol{\alpha}_{t+1}=\boldsymbol{\alpha}_t. \end{cases} \quad (28)$$

Let χ_t denote the set of all the targets present in the scene at time t. We now define the state transitions of the targets that are present at both times t and $t+1$. For such a target, $n_t \in \{\chi_{t+1}\cap\chi_t\}$, define a vector of its position and velocity

as $\boldsymbol{\xi}_{t+1,n_t}=[\boldsymbol{\rho}_{t+1,n_t}^T,\dot{\boldsymbol{\rho}}_{t+1,n_t}^T]^T$. Then, given $\boldsymbol{\xi}_{t,n_t}$, we have

$$\boldsymbol{\xi}_{t+1,n_t}=\boldsymbol{F}_{n_t}\boldsymbol{\xi}_{t,n_t}+\boldsymbol{v}_{t,n_t}, \tag{29}$$

where $\boldsymbol{F}_{n_t}$ is the state transition matrix and $\boldsymbol{v}_{t,n_t}$ is the process noise. We assume that the targets follow linear trajectories, and hence the state transition matrix is given as

$$\boldsymbol{F}_{n_t}=\begin{bmatrix}1&0&\Delta t&0\\0&1&0&\Delta t\\0&0&1&0\\0&0&0&1\end{bmatrix}, \tag{30}$$

where Δt is the system sampling time. The process noise, $\boldsymbol{v}_{t,n_t}$, is assumed to be Gaussian distributed, with a zero mean and a covariance matrix given by [25]

$$\boldsymbol{\Sigma}_{\mathrm{v},n_t}=\varepsilon_{n_t}\begin{bmatrix}\frac{1}{3}\Delta t^3&0&\frac{1}{2}\Delta t^2&0\\0&\frac{1}{3}\Delta t^3&0&\frac{1}{2}\Delta t^2\\\frac{1}{2}\Delta t^2&0&\Delta t&0\\0&\frac{1}{2}\Delta t^2&0&\Delta t\end{bmatrix}, \tag{31}$$

where ε_{n_t} is the intensity of the process noise for the n_tth target. Hence, we have

$$p(\boldsymbol{\xi}_{t+1,n_t}|\boldsymbol{\xi}_{t,n_t})=\mathcal{N}(\boldsymbol{\xi}_{t+1,n_t};\boldsymbol{F}_{n_t}\boldsymbol{\xi}_{t,n_t},\boldsymbol{\Sigma}_{\mathrm{v},n_t}). \tag{32}$$

The state transition of the vector $\boldsymbol{\theta}_t$ can be obtained by using the chain rule as

$$\begin{aligned}p(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)=&\,p(\boldsymbol{\xi}_{t+1}^*)p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t,N_{t+1})\\&\times p(N_{t+1}|N_t)\prod_{n_t\in\chi_{t+1}\cap\chi_t}p(\boldsymbol{\xi}_{t+1,n_t}|\boldsymbol{\xi}_{t,n_t}),\end{aligned} \tag{33}$$

where $p(\boldsymbol{\xi}_{t+1}^*)$ is the probability density of the position and velocity vector of the new target initiated at time $t+1$. We assume that targets belonging to any of the categories can enter the scene with an equal probability. Further, for simplicity, we assume that all the targets belonging to a particular category will enter the scene at the same location.

5.3. Measurement model

In this subsection, we describe the sensor measurement models for the multistatic radar, the infrared camera and the human scout. At time t, let $\boldsymbol{y}_t=\{\boldsymbol{y}_{1,t},\boldsymbol{y}_{2,t},\boldsymbol{y}_{3,t}\}$ be the measurements obtained from the sensors. $\boldsymbol{y}_{1,t}$ corresponds to the measurement obtained from a multistatic radar system, $\boldsymbol{y}_{2,t}$ corresponds to the measurement obtained from an infrared camera, and $\boldsymbol{y}_{3,t}$ corresponds to the measurement obtained by the human scout.

Multistatic radar: We transmit a coherent train of multiple pulses with a pulse repetition period of t_{p} seconds from the transmit antenna:

$$s(t')=\sum_{l=0}^{L-1}a_l(t'-lt_{\mathrm{p}}), \tag{34}$$

where $a_l(t')$ is the transmitted signal in the lth pulse. The discrete version of the signal $a_l(t')$ is assumed to be of length G and it is denoted as $\boldsymbol{a}_l$. Let τ_{p,n_t} be the total time taken for the signal to travel from the transmit antenna to the n_tth target and back to the pth receive antenna, and ν_{p,n_t} be the Doppler frequency shift due to the n_tth target. The parameters τ_{p,n_t} and ν_{p,n_t} depend on the position and velocity of the n_tth target and the positions of the transmit and the receive antennas. We have

$$\tau_{p,n_t}=\frac{1}{c}\{R_{n_t}+R_{p,n_t}\}, \tag{35}$$

and

$$\nu_{p,n_t}=\frac{f_c}{c}\{\dot{R}_{n_t}+\dot{R}_{p,n_t}\}, \tag{36}$$

where c is the speed of propagation, f_{c} is the carrier frequency, R_{n_t} is the range from the transmit antenna to the n_t^{th} target, R_{p,n_t} is the range from the pth receive antenna to the n_tth target, and $\dot{R}_{n_t}$ and $\dot{R}_{p,n_t}$ are the corresponding range rates. Without the loss of generality, we assume that the transmitter is located at the origin of the coordinate system. The location of the pth receive antenna is denoted by $(x_{\mathrm{Rx},p},y_{\mathrm{Rx},p})$. We then have

$$\begin{aligned}&R_{n_t}=\sqrt{\rho_{\mathrm{x},n_t}^2+\rho_{\mathrm{y},n_t}^2},\\&R_{p,n_t}=\sqrt{(x_{\mathrm{Rx},p}-\rho_{\mathrm{x},n_t})^2+(y_{\mathrm{Rx},p}-\rho_{\mathrm{y},n_t})^2},\\&\dot{R}_{n_t}=\frac{\dot{\rho}_{\mathrm{x},n_t}\rho_{\mathrm{x},n_t}+\dot{\rho}_{\mathrm{y},n_t}\rho_{\mathrm{y},n_t}}{R_{n_t}},\quad\text{and}\\&\dot{R}_{p,n_t}=\frac{\dot{\rho}_{\mathrm{x},n_t}(x_{\mathrm{Rx},p}-\rho_{\mathrm{x},n_t})+\dot{\rho}_{\mathrm{y},n_t}(y_{\mathrm{Rx},p}-\rho_{\mathrm{y},n_t})}{R_{p,n_t}}.\end{aligned} \tag{37}$$

The received signal at the pth receive antenna, due to the signal bouncing off the n_tth target, can then be expressed as

$$y_{1,p}(t')=\beta_{p,n_t}\sqrt{\gamma_1\zeta_{p,n_t}}s(t'-\tau_{p,n_t})e^{j2\pi\nu_{p,n_t}t'}+w_{1,p}(t'), \tag{38}$$

where β_{p,n_t} is the target RCS which is a function of the target category, γ_1 is the transmit signal energy, $\zeta_{p,n_t}\propto 1/R_{n_t}^2R_{p,n_t}^2$ represents the path loss effects, and $w_{1,p}(t')$ is the additive receiver noise. We assume that the mapping from the target category to the target RCS is known and one-to-one. The noise is assumed to be circularly symmetric, complex, white, and follow a Gaussian distribution. The corresponding discrete-time signal can be obtained by sampling the received signal and considering only N_{p} samples. It can be expressed as

$$\begin{aligned}y_{1,p}(nt_{\mathrm{s}})=&\,\beta_{p,n_t}\sqrt{\gamma_1\zeta_{p,n_t}}\sum_{l=0}^{L-1}a_l(nt_{\mathrm{s}}-lt_{\mathrm{p}}-\tilde{\tau}_{p,n_t})e^{j2\pi\nu_{p,n_t}lt_{\mathrm{p}}}\\&+w_p(nt_{\mathrm{s}}).\end{aligned} \tag{39}$$

Here $\tilde{\tau}_{p,n_t}=f_s\tau_{p,n_t}$ is the delay in the discrete domain. Since $a(t')$ is a narrow pulse, in arriving at Eq. (39), within each $a(t')$, we have approximated the term $e^{j2\pi\nu_{p,n_t}t'}$ as $e^{j2\pi\nu_{p,n_t}lt_{\mathrm{p}}}$ (a constant). Simplifying Eq. (39), we get

$$\boldsymbol{y}_{1,p,n_t}=\beta_{p,n_t}\sqrt{\gamma_1\zeta_{p,n_t}}\underbrace{(\boldsymbol{\Upsilon}(p,n_t)\otimes\boldsymbol{\Gamma}(p,n_t))\boldsymbol{s}}_{\boldsymbol{\phi}_{p,n_t}}+\boldsymbol{w}_{1,p}, \tag{40}$$

where

- $\boldsymbol{y}_{1,p,n_t}$ is a $LN_{\mathrm{p}}\times 1$ received signal vector at the pth antenna due to the signal bouncing off the n_tth target.
- $\boldsymbol{\Upsilon}(p,n_t)$ is an $L\times L$ Doppler modulation matrix defined as $\mathrm{diag}\{1,e^{j2\pi\nu_{p,n_t}t_{\mathrm{p}}},\ldots,e^{j2\pi\nu_{p,n_t}(L-1)t_{\mathrm{p}}}\}$.
- $\boldsymbol{\Gamma}(p,n_t)$ is a $N_{\mathrm{p}}\times G$ time shift matrix defined as $[\boldsymbol{0}_{\tilde{\tau}_{p,n_t}\times G};\ \boldsymbol{I}_G;\ \boldsymbol{0}_{N_{\mathrm{s}}-G-\tilde{\tau}_{p,n_t}\times G}]$.

- $\boldsymbol{s}$ is a $LG\times 1$ column vector obtained by stacking the transmitted signal in each pulse i.e., $\boldsymbol{s}=[\boldsymbol{a}_0^T,\boldsymbol{a}_1^T,\ldots,\boldsymbol{a}_{L-1}^T]^T$.
- $\boldsymbol{w}_{1,p}$ is a $LN_s\times 1$ complex additive white Gaussian noise at the pth receiver with a zero mean and a covariance matrix $\boldsymbol{\Sigma}_{1,p}=\sigma_{1,p}^2\boldsymbol{I}_{LN_p}$.

The received signal due to all the targets can now be expressed in a matrix form as

$$\boldsymbol{y}_{1,p,t}=\sum_{n_t=1}^{N_t}\beta_{p,n_t}\sqrt{\gamma_1\zeta_{pqm}}(\boldsymbol{\Upsilon}(p,n_t)\otimes\boldsymbol{\Gamma}(p,n_t))\boldsymbol{s}+\boldsymbol{w}_{1,p}=\boldsymbol{\Phi}_{p,t}\boldsymbol{\beta}_{p,t}+\boldsymbol{w}_{1,p}, \tag{41}$$

where

- $\boldsymbol{y}_{1,p,t}$ is a $LN_p\times 1$ received signal vector at the pth antenna due to all the targets.
- $\boldsymbol{\Phi}_{p,t}$ is a $LN_p\times N_t$ matrix defined as $\boldsymbol{\Phi}_{p,t}=[\boldsymbol{\phi}_{p,1},\ldots,\boldsymbol{\phi}_{p,N_t}]$.
- $\boldsymbol{\beta}_{p,t}$ is a $N_t\times 1$ vector defined as $\boldsymbol{\beta}_t=[\beta_{p,1},\ldots,\beta_{p,N_t}]$.

Hence we have

$$p(\boldsymbol{y}_{1,p,t}|\boldsymbol{\theta}_t)=\mathbb{CN}(\boldsymbol{y}_{1,p,t};\boldsymbol{\Phi}_{p,t}\boldsymbol{\beta}_{p,t},\boldsymbol{\Sigma}_{1,p})\quad\text{for } p=1,2,3. \tag{42}$$

Since the measurements obtained at the receive antennas are independent of each other, we have

$$p(\boldsymbol{y}_{1,t}|\boldsymbol{\theta}_t)=\mathbb{CN}(\boldsymbol{y}_{1,t};\boldsymbol{\Phi}_t\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_1), \tag{43}$$

where $\boldsymbol{y}_{1,t}=[\boldsymbol{y}_{1,1,t}^T,\boldsymbol{y}_{1,2,t}^T,\boldsymbol{y}_{1,3,t}^T]^T$, $\boldsymbol{\Phi}_t=\text{blkdiag}\{\boldsymbol{\Phi}_{1,t},\boldsymbol{\Phi}_{2,t},\boldsymbol{\Phi}_{3,t}\}$, $\boldsymbol{\beta}_t=[\boldsymbol{\beta}_{1,t}^T,\boldsymbol{\beta}_{2,t}^T,\boldsymbol{\beta}_{3,t}^T]^T$ and $\boldsymbol{\Sigma}_1=\text{blkdiag}\{\boldsymbol{\Sigma}_{1,1},\boldsymbol{\Sigma}_{1,2},\boldsymbol{\Sigma}_{1,3}\}$.

Infrared camera: We use a measurement model similar to the one proposed in [32,33] for the infrared camera. The output of an infrared camera, which is an $R\times C$ matrix of pixel values, is modeled as a noisy version of an ideal image $\boldsymbol{I}_0$ convolved with the point-spread function of the camera. For simplicity, we assume that the point-spread function to be a delta function and the ideal image to be of the form:

$$\boldsymbol{I}_0(\boldsymbol{z})=\sum_{n_t=1}^{N_t}T_{n_t}\delta(\boldsymbol{z}-\boldsymbol{\rho}_{n_t}), \tag{44}$$

where $\boldsymbol{z}$ is a two-dimensional pixel location in the image obtained by the camera, $\boldsymbol{\rho}_{n_t}$ is the two-dimensional pixel location of the n_tth target in the image obtained by the camera, which depends on the actual location of the n_tth target, and T_{n_t} is a constant that depends on the target category. Following [32], we express the likelihood of the output of the camera as

$$p(\boldsymbol{y}_{2,t}|\boldsymbol{\theta}_t)=\mathcal{N}(\boldsymbol{y}_{2,t};\text{vec}(\gamma_2\boldsymbol{I}_0),\boldsymbol{\Sigma}_2), \tag{45}$$

where $\boldsymbol{y}_{2,t}$ is a vector form of the matrix of pixel values that correspond to the output of the camera, γ_2 is a constant that depends on the quality of the camera, and $\boldsymbol{\Sigma}_2=\boldsymbol{\Sigma}_{2,1}\otimes\boldsymbol{\Sigma}_{2,2}$ where $\boldsymbol{\Sigma}_{2,1}$ and $\boldsymbol{\Sigma}_{2,2}$ are the covariance matrices of the measurement noise along the rows and the columns, respectively.

Human scout: The measurement report given by the scout is an M-dimensional vector with its mth entry representing the number of targets of type m. The total number of targets at time t, as counted by the scout, is given as

$$N_{3,t}=\sum_{m=1}^{M}y_{3,t,m}, \tag{46}$$

where $y_{3,t,m}$ is the mth entry of the vector $\boldsymbol{y}_{3,t}$. In the above, we use a subscript, $N_{3,t}$, to denote the number of targets counted by the scout in order to differentiate it from the actual number of targets in the scene which is denoted by N_t. Let $N_{\max}$ be an upper bound on $N_{3,t}$. We obtain the probability mass distribution for the number of targets counted by scout, $N_{3,t}$, denoted $g(N_{3,t})$, by evaluating a Gaussian density[1] with mean N_t, the actual number of targets, and variance $\sigma_{3,t}^2$ followed by normalizing:

$$g(N_{3,t}=k)=\frac{\mathcal{N}(k;N_t,\sigma_{3,t}^2)}{\sum_{i=1}^{N_{\max}}\mathcal{N}(i;N_t,\sigma_{3,t}^2)},\quad N_t>0 \tag{47}$$

The variance $\sigma_{3,t}^2$ is chosen to be N_t/γ_3, where the parameter γ_3 is proportional to the level of the training that the scout undergoes and the quality of the equipment that he uses. The variance also depends on the actual number of the targets in the scene. The greater the number of targets, the higher the probability that the scout incorrectly counts them. A similar probability mass function was used in [33] for the number of targets counted by a human scout. Let p_c denote the probability that the scout counts at least one target incorrectly. We model the probability p_c to be inversely proportional to γ_3, i.e., $p_c=b/\gamma_3$, where b is a known constant chosen such that $p_c\in[0,1]$. From the expression for p_c, it can be seen that well-trained scouts and scouts with better equipment have a lower probability of incorrectly identifying the targets. We model the scout's measurements $\boldsymbol{y}_{3,t}$ to follow a multinomial distribution, whenever at least one target is identified incorrectly. The likelihood for the scout's measurement report can then be expressed as

$$p(\boldsymbol{y}_{3,t}|\boldsymbol{\theta}_t)=\begin{cases}p_c\sum_{k=1}^{N_{\max}}g(k)\dfrac{k!}{y_{3,t,1}!\ldots y_{3,t,M}!}q_{1,t}^{y_{3,t,1}}\ldots q_{M,t}^{y_{3,t,M}} & \text{if at least one target is incorrectly identified,}\\ (1-p_c)+p_c\left[g(N_{3,t})\dfrac{N_{3,t}!}{y_{3,t,1}!\ldots y_{3,t,M}!}q_{1,t}^{y_{3,t,1}}\ldots q_{M,t}^{y_{3,t,M}}\right] & \text{otherwise.}\end{cases} \tag{48}$$

The probabilities $q_{1,t},q_{2,t},\ldots,q_{M,t}$ are obtained based on the actual values of the state vector $\boldsymbol{\theta}_t$. We first evaluate an

[1] Note here that we are computing a discrete probability distribution by sampling a continuous Gaussian density. The resulting probability mass function for the number of targets counted by the scout is still a discrete distribution.

$N_t \times M$ matrix $\boldsymbol{Q}_t$, such that

$$[\boldsymbol{Q}_t]_{n_t,m} = \begin{cases} 1-q_c & \text{if the } n_t\text{th target is of type } m, \\ \dfrac{q_c}{M-1} & \text{otherwise} \end{cases}, \quad (49)$$

where $0 \le q_c \le 1$ is a known constant. We then normalize $\boldsymbol{Q}_t$ to obtain the probability vector $q_{m,t}$ as $q_{m,t} = (1/N_t) \sum_{n_t=1}^{N_t} [\boldsymbol{Q}_t]_{n_t,m}$. We assumed in this work that the samples from all the sensors arrive at the receiver at the same time and that the sensors are time-synchronized.

Filtering algorithm: In this example, we divide the state-space into three partitions. The first partition $\boldsymbol{\theta}_{1,t}$ comprises the target number and the target categories, i.e., $\boldsymbol{\theta}_{1,t} = [N_t, \boldsymbol{\alpha}_t^T]^T$. The second and the third partitions comprise the target positions and velocities, respectively, i.e., $\boldsymbol{\theta}_{2,t} = [\boldsymbol{\rho}_{1,t}^T, \ldots, \boldsymbol{\rho}_{N_t,t}^T]^T$ and $\boldsymbol{\theta}_{3,t} = [\dot{\boldsymbol{\rho}}_{1,t}^T, \ldots, \dot{\boldsymbol{\rho}}_{N_t,t}^T]^T$. It can be seen that the evolution of the second and third partitions depends on the evolution of the first partition. The measurements collected by the scout depend only on the first partition, the measurements collected by the infrared camera are a function of both first and the second partitions, and the measurements collected by the radar are function of all the three partitions. Thus the measurement and the state functions satisfy the hierarchical model that is described Eq. (7). In order to obtain the state estimate, we use the HPF that is described in Section 3.2.

In general, it is observed that partitioning the samples of each target individually results in a better filtering performance [31,34]. The proposed hierarchical filtering can be extended to operate in this manner by further dividing the second and third partitions $\boldsymbol{\theta}_{2,t}$ and $\boldsymbol{\theta}_{3,t}$.

Table 1
The initial positions and velocities of all the target classes.

Target class	Initial position		Initial velocity	
	Fixed	Varying	Fixed	Varying
1	(5,5)	(5,5)	(10,10)	(10,0)
2	(10,35)	(10,0)	(5,0)	(0,10)
3	(20,20)	(20,20)	(3,3)	(3,3)
4	(35,35)	(12,15)	(−10,−10)	(0,−10)
5	(0,30)	(8,12)	(5,−10)	(10,−5)

5.4. Results and discussion

In this section, we use numerical examples to demonstrate the performance of the tracking system when the proposed hierarchical filtering method is used for target tracking. In order to quantify the performance of the multiple target tracking system, we define four performance metrics. We describe the simulation setup first and then discuss the examples.

Target parameters: We consider surveillance of a region for a period of 20 tracking intervals. The duration of each tracking interval was 0.1 s ($\Delta t = 0.1$ s). We consider

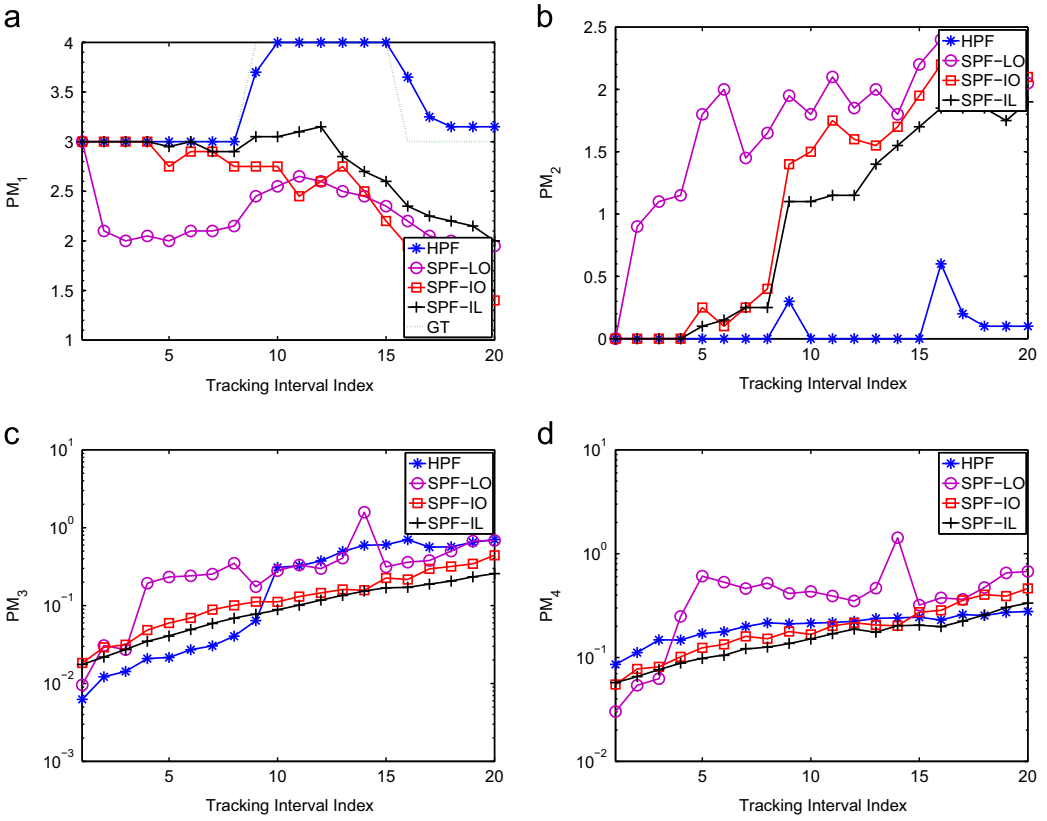

Fig. 3. Performance comparison for varying number of targets. (a) Average number of targets. (b) Average number of targets incorrectly identified. (c) RMSE in range. (d) RMSE in velocity.

tracking under two scenarios. In the first scenario, during the first 0.8 s, which corresponds to 8 intervals, there were 3 targets in the scene, during the next 0.8 s, i.e., between the 9th and the 15th interval there were 4 targets, and thereafter there were 3 targets again. The number of target classes was chosen to be 5 ($M=5$), and the initial positions and initial velocities of the targets belonging to various classes were chosen as shown in the Table 1. During the entire tracking period, the target categories were chosen as

$$\boldsymbol{\alpha}_t = \begin{cases} \{1,2,4\}, & t = 1,2,\ldots,8 \\ \{1,2,4,5\}, & t = 9,10,\ldots,15 \\ \{1,2,5\}, & t = 16,\ldots,20 \end{cases}$$

In the second scenario, there were three targets during the entire duration, i.e. $\boldsymbol{\alpha}_t = \{1,2,4\}$ for $t = 1,\ldots,20$. The initial positions and velocities of these targets were again chosen according to Table 1. The probabilities of the birth and the death of the targets were chosen to be 0.01, i.e., $p_d = p_b = 0.01$, respectively. The constants T_{n_t} in the measurement model of the infrared camera (see Eq. (44)) were chosen to be $T_1 = 4.2, T_2 = 8, T_3 = 9, T_4 = 10.4, T_5 = 13.6$, respectively.

Signal and sensor parameters: In each pulse, we transmit a spread-spectrum waveform [35,36] with 16 ($G=16$) chips from the radar antenna. The total bandwidth was 100 MHz ($B = 100$) and the carrier frequency, f_c, of the transmitted waveforms was 1 GHz. We used four ($L=4$) pulses in each tracking interval. The radar receive antennas were located at $(x_{\mathrm{Rx},1}, y_{\mathrm{Rx},1}) = (0,0), (x_{\mathrm{Rx},2}, y_{\mathrm{Rx},2}) = (20,0)$, $(x_{\mathrm{Rx},3}, y_{\mathrm{Rx},3}) = (40,0)$, respectively, and the variance of the measurement noise at each receiver was $\sigma^2_{1,p} = 1 \times 10^{-3}, p = 1,2,3$. The covariance matrices of the measurement noise at the infrared camera were chosen to be $\boldsymbol{\Sigma}_{2,1} = \sigma_2^2 \boldsymbol{I}_R, \boldsymbol{\Sigma}_{2,2} = \sigma_2^2 \boldsymbol{I}_C$, with $\sigma_2^2 = 1 \times 10^{-2}$. The constants b and q_c for the human scout were chosen to be 0.2 and 0.05, respectively. We assumed that all the targets are observable by all the sensors.

We evaluated the performance of the system using four metrics: the average number of targets detected in the scene, the average number of targets identified incorrectly, the root mean-squared error in the position of correctly identified targets, and the root mean-squared error in the velocity of correctly identified targets. Let the estimate of the state vector at time t in the ith Monte-Carlo iteration be $\hat{\boldsymbol{\theta}}_{t,i} = [\hat{N}_t, \hat{\boldsymbol{\rho}}_1^T, \hat{\boldsymbol{\rho}}_1^T, \hat{\alpha}_1, \ldots, \hat{\boldsymbol{\rho}}_{N_t}^T, \hat{\boldsymbol{\rho}}_{N_t}^T, \hat{\alpha}_{N_t}]_i^T$. The four performance metrics are then defined as

$$\mathrm{PM}_1 = \frac{1}{N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \hat{N}_{t,i},$$

$$\mathrm{PM}_2 = \frac{1}{N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \mathrm{card}\left((\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}}_{t,i}) \bigcup (\hat{\boldsymbol{\alpha}}_{t,i} - \boldsymbol{\alpha}_t)\right),$$

$$\mathrm{PM}_3 = \frac{1}{|B_t| N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \sum_{n_t \in B_t} \sqrt{(\rho_{\mathrm{x},n_t} - \hat{\rho}_{\mathrm{x},n_t})^2 + (\rho_{\mathrm{y},n_t} - \hat{\rho}_{\mathrm{y},n_t})^2},$$

$$\mathrm{PM}_4 = \frac{1}{|B_t| N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \sum_{n_t \in B_t} \sqrt{(\dot{\rho}_{\mathrm{x},n_t} - \hat{\dot{\rho}}_{\mathrm{x},n_t})^2 + (\dot{\rho}_{\mathrm{y},n_t} - \hat{\dot{\rho}}_{\mathrm{y},n_t})^2},$$

where B_t is the set of correctly identified targets.

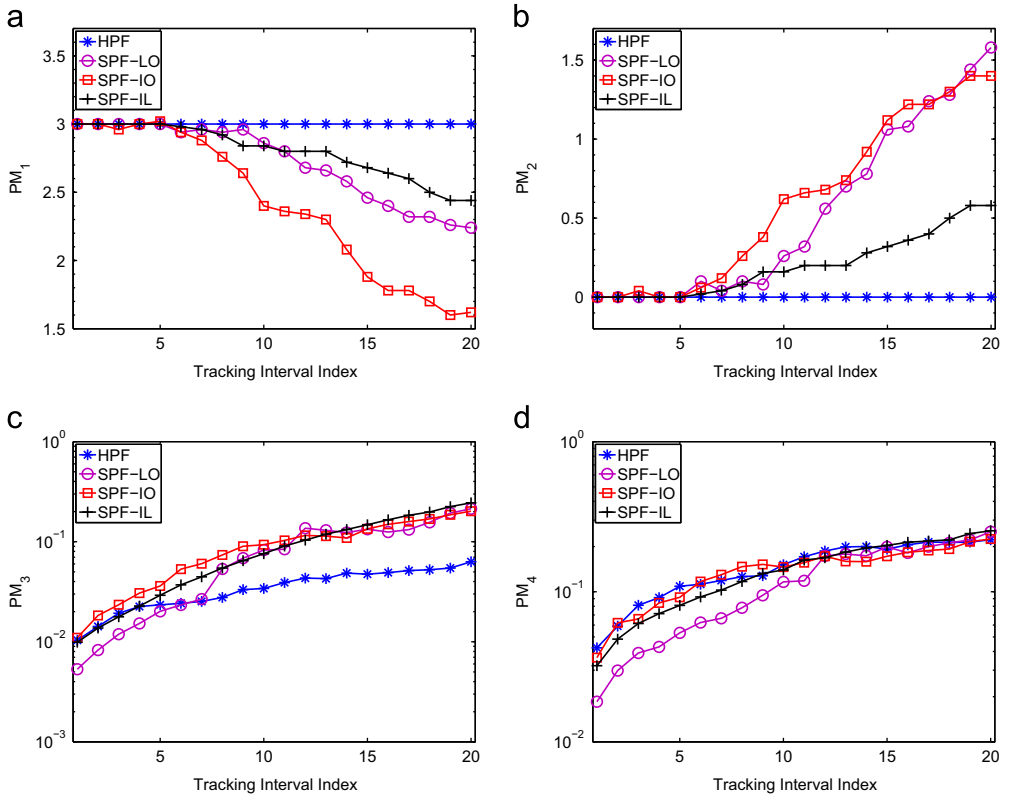

Fig. 4. Performance comparison for fixed number of targets. (a) Average number of targets. (b) Average number of targets incorrectly identified. (c) RMSE in range. (d) RMSE in velocity.

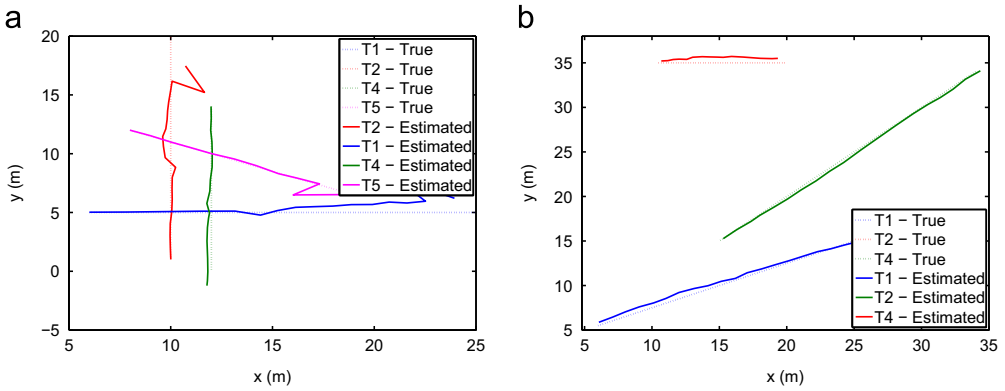

Fig. 5. Estimated vs. actual target trajectories. (a) Scenario 1. (b) Scenario 2.

Table 2
The average computational time in seconds for various algorithms.

No. of particles	HPF	SPF-LO	SPF-IO	SPF-IL
500	4.23	1.02	0.98	0.89
1000	9.36	2.26	1.96	1.85
2000	18.76	4.19	3.74	2.98

In Fig. 3, we plot these metrics using the four methods, for varying number of targets. In the first method, we used the proposed HPF. In the second, third, and fourth methods, we used SPF-LO, SPF-IO, SPF-IL, where the weights were updated following Eqs. (13), (15), and (17), respectively. We used $N_s = 1000$ particles for the simulations and the results were averaged over $N_{mc} = 50$ Monte Carlo iterations.

In Fig. 3(a), we plot the actual number of targets (ground truth), labeled as GT along with estimates of the number of targets obtained using other methods. It can be seen from this figure that the system was able to accurately estimate the number of targets using the HPF approach. On the other hand, using SPF-LO, SPF-IO, and SPF-IL resulted in incorrect estimation of number of targets. It can also been seen that using HPF resulted in higher correct identifications of the target categories compared to the other methods. However, from Fig. 3(d), it can be seen that the RMSE in the range and velocity, per target, for the correctly identified targets, using HPF was comparable to the RMSE obtained using the SPF based methods.

In Fig. 4, we plot the performance metrics for a fixed number of targets. It can be seen using HPF resulted in an accurate estimate of the target number, higher correct identifications, and lower RMSE in range when compared to the performance obtained using SPF based methods. Among the other methods, SPF-IL performed better, followed by SPF-LO and SPF-IO. The RMSE in the velocity estimate using HPF was comparable to the RMSE using SPF based methods. This is due to the fact that the velocity and the range evolve independent of each other. We plot the actual target trajectories for the two scenarios and the estimated trajectories (using a single Monte-Carlo run) using the proposed HPF in Fig. 5.

In Table 2, we list the average computational time taken by each of these algorithms for the case of varying number of targets. We considered the average running time for the case of $N_s = 500, 1000$ and 2000 particles. The simulations were run on an 8-core processor using parallel matlab sessions. HPF requires a sampling and a weight update step in each stage of the filtering process, and as a result it takes longer time compared to other SPF based methods. An n-stage HPF requires n sample generation and weight update steps, whereas an SPF based method requires only one sampling and weight update step.

6. Summary

We considered the problem of recursive Bayesian estimation in hierarchical Bayesian models that use multi-modal data. We proposed a new filter, called the hierarchical particle filter (HPF) to obtain the state estimate. HPF finds the estimate of the state vector in several stages. At each stage, the filter uses the posterior weights obtained in the last stages as extrinsic information, and computes the estimate of the state vector in the current stage. The data obtained from multiple sensing modalities is combined in both the time-update and the measurement-update step of the recursive Bayesian filtering. We use the proposed filtering method for the joint initiation, termination and tracking of multiple targets using multi-modal sensors. We used a sensor network comprising three different types of sensors: a radar, an infrared camera and a human scout. We compared the tracking performance with standard particle filtering (SPF) using linear opinion (SPF-LO), independent opinion (SPF-IO) and independent likelihood (SPF-IL) for the data fusion. The results demonstrated that HPF was able to accurately identify the number of targets, produce higher correct identifications, and lower root mean-squared error (RMSE) in range estimates when compared to the performance obtained using the SPF-LO, SPF-IO, and SPF-IL. The RMSE in the velocity estimates was similar for all the methods. In the future, we will use HPF for other applications such as object tracking using audio and video sensors.

References

[1] B. Han, S.W. Joo, L.S. Davis, Multi-camera tracking with adaptive resource allocation, Int. J. Comput. Vis. 91 (2011) 45–58.
[2] E. Kenneth, A. Rajendra, N. Kannathal, C.M. Lim, Data fusion of multimodal cardiovascular signals, in: 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE, Shangai, China, 2005, pp. 4689–4682.
[3] D. Chetverikov, Z. Jankó, 3d Reconstruction by Multimodal Data Fusion, ERCIM News, 2010 (80).
[4] A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein, W. Hong, Model-driven data acquisition in sensor networks, in: Proceedings of the 13th International Conference on Very Large Data Bases, vol. 30, VLDB Endowment, 2004, pp. 588–599.
[5] Y. Bar-Shalom, P. Willet, X. Tian, Tracking and Data Fusion: A Handbook of Algorithms, 3rd Edition, 2011.
[6] P.K. Varshney, Multisensor data fusion, Electr. Commun. Eng. J. 9 (6) (1997) 245–253.
[7] O. Punska, Bayesian Approaches to Multi-Sensor Data Fusion (M.Phil thesis), University of Cambridge, August 1999.
[8] D.L. Hall, J. Llinas, An introduction to multisensor data fusion, Proc. IEEE 85 (1) (1997) 6–23.
[9] S. Bottone, C. Stanek, Probabilistic graphical models and their application in data fusion, in: Proceedings of SPIE, vol. 6566, 2007.
[10] S.M. Kay, Fundamentals of Statistical Signal Processing, Estimation Theory, vol. 1, Prentice Hall, 1993.
[11] Z. Chen, Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond, Technical Report, Adaptive Syst. Lab., McMaster University, 2003.
[12] M.S. Arulampalam, S. Maskell, N.J. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[13] D.B. Rubin, A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creation a few imputations when fractions of missing information are modest: the SIR algorithm, J. Am. Stat. Assoc. 52 (1987) 543–546.
[14] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, in: Proceedings of Radar and Signal Processing, vol. 140, IEEE, 1993, pp. 107–113.
[15] M.F. Bugallo, P. Djurić, Complex systems and particle filtering, in: Fourty Second Asilomar Conference on Signals, Systems and Computers, IEEE, Asilomar, CA, 2008, pp. 1183–1187.
[16] M. Stone, The opinion pool, Ann. Stat. 32 (1961) 1339–1342.
[17] J. Manyika, H. Durrant-Whyte, Data Fusion and Sensor Management: a Decentralized Information-Theoretic Approach, Ellis Horwood, New-York, 1994.
[18] B. Han, S. Joo, L. Davis, Probabilistic fusion tracking using mixture kernel-based Bayesian filtering, in: International Conference on Computer Vision (ICCV), 2007, pp. 1–8.
[19] B. Han, Y. Zhu, D. Comaniciu, L.S. Davis, Visual tracking by continuous density propagation in sequential Bayesian filtering framework, IEEE Trans. Pattern Anal. Mach. Intell. 31 (5) (2009) 919–930.
[20] P. Chavali, A. Nehorai, Hierarchical particle filtering for target tracking in multi-modal sensor networks, in: Proceedings of Seventh Sensor Array and Multichannel Signal Processing (SAM) Workshop, Hoboken, NJ, 2012.
[21] A. Doucet, On sequential Monte-Carlo methods for Bayesian filtering, Technical Report, Dept. of Engr., Univ. Cambridge, UK, 1998.
[22] W. Yi, M. Morelande, L. Kong, J. Yang, A computationally efficient particle filter for multitarget tracking using an independence approximation, IEEE Trans. Signal Process. 61 (4) (2013) 843–856, http://dx.doi.org/10.1109/TSP.2012.2229999.
[23] M. Orton, W. Fitzgerald, A Bayesian approach to tracking multiple targets using sensor arrays and particle filters, IEEE Trans. Signal Process. 50 (2) (2002) 216–223, http://dx.doi.org/10.1109/78.978377.
[24] M. Bolic, P.M. Djuric, S. Hong, New resampling algorithms for particle filters, in: Proceedings of International Conference on Acoustics Speech and Signal Processing, vol. 2 of 2, IEEE, Hong Kong, 2003, pp. 589–592.
[25] Y. Bar-Shalom, X.-R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation, Wiley, New York, 2001.
[26] S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, vol. 685, Artech House Noorwood, MA, 1999.
[27] R. Chellapa, Site model construction for exploitation of EO and SAR images, in: O. Firschein (Ed.), RADIUS: IU for Imagery Intelligence, Morgan Kaufman, 1997, pp. 185–208.
[28] A. Theil, L. Kester, S.P. van den Broek, P. van Dorp, R. van Sweeden, FRESENEL program: fusion of radar and electro-optical signals for surveillance on land, in: I. Kadar (Ed.), Society of Photo-Optical Instrumentation Engineers Conference Series, vol. 4380, 2001, pp. 453–461.
[29] J. Zhang, A. Papandreou-Suppappola, M. Rangaswamy, Multi-target tracking using multi-modal sensing withwaveform configuration, in: International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, USA, 2010, pp. 3890–3893.
[30] A. García-Fernàndez, Detection and Tracking of Multiple targets Using Wireless Sensor Networks (Ph.D. thesis), Universidad Politécnica de Madrid, Madrid, Spain, 2011.
[31] J. Vermaak, S. Godsill, P. Perez, Monte-Carlo filtering for multi-target tracking and data association, IEEE Trans. Aerosp. Electr. Syst. 41 (1) (2005) 309–332, http://dx.doi.org/10.1109/TAES.2005.1413764.
[32] D. Snyder, A. Hammoud, R. White, Image recovery from data acquired with a charge-coupled-device camera, J. Opt. Soc. Am. 10 (5) (1993) 1014–1023.
[33] M. Smith, Bayesian Sensor Fusion: A Framework for Using Multi-Modal Sensors to Estimate Target Locations and Identities in a Battlefield Scene (Ph.D. thesis), Department of Statistics, Florida State University, 2003.
[34] P. Chavali, A. Nehorai, Concurrent particle filtering and data association using game theory for tracking multiple maneuvering targets, IEEE Trans. on Signal Processing 61 (2013) 4934–4948.
[35] Z.S. Dobrosavljevic, M.L. Dukic, A method of spread spectrum radar polyphase code design by nonlinear programming, Eur. Trans. Telecommun. 7 (3) (1996) 239–242.
[36] Y. Wang, X. Li, Y. Wang, Novel spread-spectrum radar waveform, Proceedings of SPIE, Radar Sensor Technologies 3066 (1997) 186–193.