

Chapter 15

Environment Adaption for *Companion-Systems*

Stephan Reuter, Alexander Scheel, Thomas Geier, and Klaus Dietmayer

Abstract One of the key characteristics of a Companion-System is the adaptation of its functionality to the user's preferences and the environment. On the one hand, a dynamic environment model facilitates the adaption of output modalities in human computer interaction (HCI) to the current situation. On the other hand, continuous tracking of users in the proximity of the system allows for resuming a previously interrupted interaction. Thus, an environment perception system based on a robust multi-object tracking algorithm is required to provide these functionalities. In typical Companion-System applications, persons in the proximity are closely spaced, which leads to statistical dependencies in their behavior. The multi-object Bayes filter allows for modeling these statistical dependencies by representing the multi-object state using random finite sets. Based on the social force model and the knowledge base of the companion system, an approach to modeling object interactions is presented. In this work, the interaction model is incorporated into the prediction step of the sequential Monte Carlo (SMC) of the multi-object Bayes filter. Further, an alternative implementation of the multi-object Bayes filter based on labeled random finite sets is outlined.

15.1 Introduction

In addition to adapting the system behavior to the user's preferences and its cognitive state, a *Companion-System* is expected to adapt to the current environment. An intuitive example is the adaptation of the system's input and output modalities. For example, audio output should not be used for providing confidential information if other persons are in the proximity of the *Companion-System*. Further, the presence of other persons typically increases the uncertainty of speech and gesture input.

S. Reuter (✉) • A. Scheel • K. Dietmayer
Institute of Measurement, Control, and Microtechnology, Ulm University,
Albert-Einstein-Allee 41, 89081 Ulm, Germany
e-mail: stephan.reuter@uni-ulm.de; alexander.scheel@uni-ulm.de; klaus.dietmayer@uni-ulm.de

T. Geier
Institute of Artificial Intelligence, Ulm University, James-Franck-Ring, 89081 Ulm, Germany
e-mail: thomas.geier@alumni.uni-ulm.de

To realize these capabilities, the *Companion*-System requires an exact model of its environment including all persons in its proximity.

The environment model is realized using a multi-object tracking system which jointly estimates the number of persons as well as the current state of the individual persons. The continuous tracking of the persons in the proximity of the *Companion*-System additionally facilitates the resumption of previously interrupted interactions. Standard multi-object tracking approaches like the Joint Probabilistic Data Association (JPDA) [1] filter, the Joint Integrated Probabilistic Data Association (JIPDA) filter [16], and Multiple Hypotheses Tracking (MHT) [18] are bottom-up approaches which extend the Kalman filter [7] to facilitate the tracking of multiple objects. During the last decade, approximations of the Multi-Object Bayes filter [11] became very popular in multi-object tracking applications. The representation of the multi-object state using random finite sets (RFSs) naturally represents the uncertainty in the number of objects as well as in their individual states. Hence, a realization of an RFS delivers an estimate for the number of persons in the proximity of the *Companion*-System. Since the number of objects in each realization is fixed, an RFS allows for the incorporation of dependencies between the objects, which is not possible in standard multi-object tracking approaches. Especially in crowded environments, the modeling of statistical dependencies between the objects is required since the presence of other objects physically restricts the possible movements of the considered object. A popular approach to modeling the interactions of persons is the Social Force Model [5], which is widely used for the simulation of evacuation scenarios.

This chapter is outlined as follows: First, the basics of random finite sets and multi-object tracking are introduced. In Sect. 15.3, a sequential Monte Carlo implementation of the multi-object Bayes filter as well as possibilities to integrate object interactions in the filtering algorithm are presented. Finally, an accurate and efficient approximation of the multi-object Bayes filter, the labeled multi-Bernoulli filter, is introduced and the differences with the sequential Monte Carlo implementation of the multi-object Bayes filter are illustrated.

15.2 Random Finite Sets and the Multi-object Bayes Filter

Random vectors are typically used to represent the state of an object in single-object tracking. A commonly used approach to applying random vectors for multi-object tracking is to stack the vectors of the individual objects. However, a drawback of this approach is the missing representation of the uncertainty about the number of objects and the ordering of the stacked vectors. In contrast, a random finite set (RFS)

$$X = \{x^{(1)}, \dots, x^{(n)}\} \quad (15.1)$$

comprises a random number $n \geq 0$ of unordered points whose states are represented by random vectors $x^{(1)}, \dots, x^{(n)}$. Hence, an RFS X implicitly captures the uncer-

tainty in the number of objects of the multi-object state. Similarly to single-object tracking, the state of the individual objects is represented using random vectors. Due to the varying number of objects in the sensor's field of view and the possibility of missed detections and false alarms, the measurement process typically returns a random number of measurements. Further, the values of the measurements are also random. Consequently, an RFS

$$Z = \{z^{(1)}, \dots, z^{(m)}\} \quad (15.2)$$

is well suited to represent the uncertainty of the measurement process, where $z^{(i)}$ denotes a single measurement. Finite set statistics (FISST) facilitates calculations with RFSs using the notion of integration and density in a way which is consistent with point process theory. Hence, FISST provides a mathematically well-founded way to extend the well-known single-object Bayes filter to multi-object tracking applications using RFSs—the multi-object Bayes Filter [11]. By filtering a finite set-valued random variable over time, the estimate obtained by the multi-object Bayes filter captures the uncertainty in number of objects in addition to the state uncertainty of the individual objects. Similarly to the single-object Bayes filter, the multi-object Bayes filter comprises a prediction and an update step, which are outlined in the following. For additional details as well as the derivations, the reader is referred to [11].

In the prediction or time update step, the multi-object posterior density at time k is predicted to the time of the next measurement. In contrast to single-object tracking, where a prediction of the object's state x to the time of the next measurement using a Markov density $f_+(x_+|x)$ is sufficient, the motion models required for multi-object tracking are far more complex. In addition to the state transition of the individual objects, the multi-object motion model is required to handle object appearance and disappearance. In some applications, it may even be necessary to incorporate object spawning, i.e. that an already existing object originates a new object. Since spawning is not relevant for the environment perception of a *Companion*-System, it is neglected in the following.

The standard multi-object motion model introduced by Mahler [11] comprises the following assumptions:

- an object survives during the transition to the time of the next measurement with probability $p_S(x)$,
- each object is assumed to move independently of other objects in the scene based on a Markov transition density $f_+(x_+|x)$,
- new-born objects follow a Poisson distributed birth density $\pi_B(\mathbf{X})$ which is statistically independent of the persisting objects.

Based on these assumptions, the multi-object Markov density is given by

$$f_+(\mathbf{X}_+|\mathbf{X}) = \pi_B(\mathbf{X}_+)\pi_+(\emptyset|\mathbf{X}) \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_S(x^{(i)}) \cdot f_+(x_+^{(\theta(i))}|x^{(i)})}{(1 - p_S(x^{(i)})) \cdot \lambda_B p_B(x_+^{(\theta(i))})}. \quad (15.3)$$

Here, the state-dependent survival probability is denoted by $p_S(\cdot)$ and the expected number of new-born objects λ_B as well as the probability density $p_B(\cdot)$ are parameters of the birth model. The sum in (15.3) includes all possible associations $\theta : \{1, \dots, n'\} \rightarrow \{0, 1, \dots, n\}$; the association $\theta(i) = 0$ represents the disappearance of object i and $\theta(i) > 0$ implies the persistence of object i . The probabilities of all objects being new-born and of the disappearance of all objects are given by

$$\pi_B(\mathbf{X}_+) = e^{-\lambda_B} \prod_{i=1}^n \lambda_B p_B(x_+^{(i)}) \quad (15.4)$$

and

$$\pi_+(\emptyset|\mathbf{X}) = \prod_{i=1}^{n'} (1 - p_S(x^{(i)})). \quad (15.5)$$

Observe that the contribution of a state vector $x^{(i)}$ to (15.4) and (15.5) is canceled out for all associations $\theta(i) > 0$ by the denominator of the product in (15.3).

Using the Chapman-Kolmogorov equation and the multi-object Markov density (15.3), the prediction of the multi-object Bayes filter to the time of the next measurement is given by

$$\pi_+(X_+) = \int f(X_+|X)\pi(X)\delta X, \quad (15.6)$$

where $\pi(X)$ is the prior multi-object density. Observe that the Markov assumption used in (15.6) implies that the multi-object posterior density $\pi(X)$ captures the entire information about the multi-object state at a time k . The integral in (15.6) is a set integral which integrates over all possible cardinalities.

The update step of the multi-object Bayes filter is based on a multi-object likelihood function $g(Z|\mathbf{X}_+)$ incorporating the single-object likelihood function $g(z|x_+)$, the field of view (FOV) of the sensor, the detection probability, and the false alarm rate. Here, the single-object likelihood function $g(z|x_+)$ provides the likelihood that a measurement z has been generated by object x_+ based on the spatial distance and the corresponding uncertainties. Further, the state-dependent detection probability incorporates the handling of the sensor's FOV. The standard multi-object measurement model [11] is illustrated by Fig. 15.1 and uses the following assumptions:

- A measurement is generated by at most one object and each object is observed by the sensor according to a single-object spatial likelihood function $g(z|x_+)$,
- Each object gives rise to a measurement according to the state dependent detection probability $p_D(x_+)$ and it is not detected with probability $1 - p_D(x_+)$,
- The sensor delivers Poisson-distributed false alarms with mean number of λ_c measurements. The false alarms follow the spatial distribution $c(z)$ which is

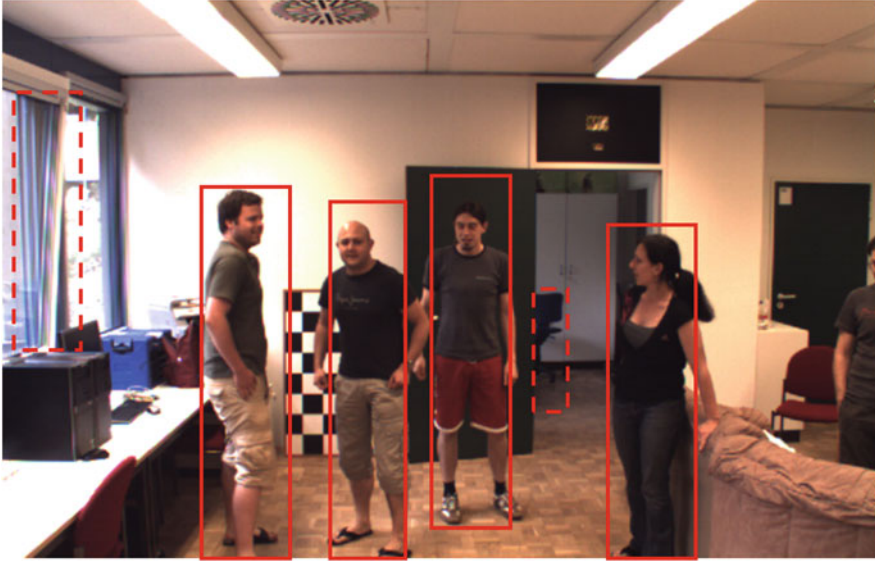


Fig. 15.1 Illustration of the events represented by the multi-object likelihood function: object detections (*red rectangles*), missed detections (no measurement for partially occluded person on the *right side*), false alarms (*red dashed rectangles*)

usually modeled by a uniform distribution over the sensor's FOV. Further, the object detection process and the false alarm process are assumed to be statistically independent and the measurements have to be conditionally independent of the objects' states.

In multi-object tracking, the track-to-measurement association, i.e. which measurement belongs to which target, is ambiguous in most scenarios due to the spatial uncertainty of the objects' states and the measurements. Further, the possibility of missed detections and false alarms additionally increases the ambiguity. To handle these ambiguities, the multi-object likelihood function averages over all possible association hypotheses, which is the best one can do if no prior knowledge about the track-to-measurement association is available. Similarly to the multi-object Markov density, the association hypotheses for n objects and m measurements are represented by $\theta : \{1, \dots, n\} \rightarrow \{0, 1, \dots, m\}$, where the measurement '0' covers possible missed detections of some of the objects. The assumption that a measurement belongs to at most one of the objects is ensured by $\theta(i) = \theta(j) > 0$ if and only if $i = j$, which uniquely assigns a measurement $z_{\theta(i)}$ to an object i . The missed detection of an object is represented by $\theta(i) = 0$.

Using the associations θ , the multi-object likelihood function covering missed detections and false alarms is given by

$$g(\mathbf{Z}|\mathbf{X}_+) = \pi_c(\mathbf{Z})\pi(\emptyset|\mathbf{X}_+) \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_D(x_+^{(i)}) \cdot g(z_{\theta(i)}|x_+^{(i)})}{(1 - p_D(x_+^{(i)})) \cdot \lambda_c c(z_{\theta(i)})}, \quad (15.7)$$

where $p_D(\cdot)$ denotes the state-dependent detection probability and $g(\cdot|\cdot)$ is the single-object likelihood function representing the likelihood of a measurement $z_{\theta(i)}$ given an object with predicted state $x_+^{(i)}$. The expected number of false alarms λ_c and the spatial distribution $c(\cdot)$ are the parameters of the Poisson clutter process. The factor

$$\pi(\emptyset|\mathbf{X}_+) = \prod_{i=1}^n (1 - p_D(x_+^{(i)})) \quad (15.8)$$

denotes the probability that none of the objects has been detected by the sensor at the current time step and

$$\pi_c(\mathbf{Z}) = e^{-\lambda_c} \prod_{z \in \mathbf{Z}} \lambda_c c(z) \quad (15.9)$$

is the probability that all measurements $z \in \mathbf{Z}$ are originated by the clutter process.

Using (15.7), the multi-object posterior density after integrating the current set of measurements is calculated using the multi-object Bayes filter update,

$$\pi(X|\mathbf{Z}) = \frac{g(\mathbf{Z}|X)\pi(X)}{\int g(\mathbf{Z}|X)\pi(X)\delta X}. \quad (15.10)$$

The recursive update of the multi-object posterior density is consequently realized by applying (15.6) and (15.10) each time a new measurement is obtained. Similarly to the single-object Bayes filter, an analytical implementation of the multi-object Bayes filter is not possible in general. However, the multi-object Bayes filter facilitates an approximation using sequential Monte-Carlo (SMC) methods as well as a closed-form implementation using δ -generalized labeled multi-Bernoulli (δ -GLMB) RFSs which are presented in detail in Sects. 15.3 and 15.4.

Further approximations of the multi-object Bayes filter, which will not be discussed in this chapter, are the probability hypothesis density (PHD) filter [9], the cardinalized probability hypothesis density (CPHD) filter [10], and the cardinality balanced multi-target multi-Bernoulli (CB-MeMber) filter [32]. While the PHD and CPHD filters approximate the multi-object posterior by the first statistical moment (and the cardinality distribution in case of the CPHD filter), the CB-MeMber filter approximates the posterior using a Multi-Bernoulli distribution. Further details as well as implementations of these filters using Gaussian mixture (GM) and SMC methods are given in [10, 28, 30–33].

15.3 SMC Implementation of the Multi-object Bayes Filter and Modeling of Object Interactions

In typical applications of *Companion-Systems*, a high number of humans in the proximity is expected. Obviously, the movement of the individual persons is restricted in these scenarios by the presence of other persons, leading to statistical dependencies in their movements. In the following, the sequential Monte-Carlo (SMC) implementation of the multi-object Bayes filter incorporating object interactions is presented. For further details, the reader is referred to [19, 20, 22] as well as [8, 11, 26, 30].

In the SMC implementation of the single-object Bayer filter, vector-valued particles $x^{(i)} \in \mathbb{R}^n$ are typically used to approximate the spatial distribution $p(x)$. For the sequential Monte-Carlo multi-object Bayes (SMC-MOB) filter, each multi-object particle has to be a sample of a random finite set and is consequently given by a set of state vectors

$$\mathbf{X}^{(i)} \triangleq \{x^{(1)}, \dots, x^{(n)}\}, \quad (15.11)$$

where the number of objects n as well as the state vectors $x^{(j)}$ are random. In the following, the state vectors $x^{(j)}$ of the multi-object particle $\mathbf{X}^{(i)}$ are conveniently called “particles”. Using the ν multi-object particles, the multi-object probability density is approximated by

$$\pi(\mathbf{X}) \cong \sum_{i=1}^{\nu} w^{(i)} \cdot \delta_{\mathbf{X}^{(i)}}(\mathbf{X}). \quad (15.12)$$

15.3.1 Prediction

The prediction step of the SMC-MOB filter has to predict each multi-object particle according to the multi-object Markov density (15.3), incorporating the motion of persisting objects as well as object appearance and disappearance. Consequently, the prediction of a multi-object particle is obtained by the union of the set of surviving particles $\mathbf{X}_{+,S}^{(i)}$ and the set of new-born particles $\mathbf{X}_B^{(i)}$:

$$\mathbf{X}_+^{(i)} = \mathbf{X}_{+,S}^{(i)} \cup \mathbf{X}_B^{(i)}. \quad (15.13)$$

The set of persisting particles of a multi-object particle $\mathbf{X}^{(i)} = \{x^{(1)}, \dots, x^{(n)}\}$ is obtained by a multi-Bernoulli distribution using the persistence probability $p_S(x^{(j)})$ as a parameter. Since a multi-Bernoulli distribution is the union of several independent Bernoulli distributions, the persistence of each object is assumed to be statistically independent of other objects. Thus, the persistence of a subset

$\{x^{(1)}, \dots, x^{(n')}\}$ has a probability of

$$\pi \left(\{x^{(1)}, \dots, x^{(n')}\} | X^{(i)} \right) = \prod_{x \in X^{(i)}} (1 - p_S(x)) \cdot \prod_{\tilde{x} \in \{x^{(1)}, \dots, x^{(n')}\}} \frac{p_S(\tilde{x})}{1 - p_S(\tilde{x})}. \quad (15.14)$$

Instead of drawing the persisting particles directly using (15.14), the independence of the M Bernoulli distributions within the multi-Bernoulli distribution facilitates sampling the persistence of each particle independently. This can be realized by drawing a uniformly distributed random number $\zeta^{(j)}$ for each particle in $X^{(i)}$. Consequently, the set of persisting particles follows

$$X_S^{(i)} = \{x : \zeta^{(j)} < p_S(x^{(j)}) \forall j = 1, \dots, |X^{(i)}|\}. \quad (15.15)$$

Hence, a particle only persists if its state-dependent survival probability is greater than the drawn random number. Finally, the $j = 1, \dots, n'$ persisting particles have to be predicted to the time of the next measurement using a single-object Markov transition density,

$$x_+^{(j)} \sim f_+ (\cdot | x^{(j)}), \quad (15.16)$$

in order to obtain the predicted set of surviving particles:

$$X_{+,S}^{(i)} = \{x_+^{(1)}, \dots, x_+^{(n')}\}. \quad (15.17)$$

The birth process is utilized to obtain the set of new-born particles $X_B^{(i)}$. Therefore, the number of appearing objects n_B is sampled from a Poisson-distributed cardinality distribution $\rho_B(n)$ with an expectation value of λ_B . The state of the new-born particles is obtained by sampling from the spatial distribution p_B of new-born objects:

$$x_+^{(j)} \sim p_B(\cdot) \forall j = 1, \dots, n_B. \quad (15.18)$$

15.3.2 Update

In the update step of the SMC-MOB, the weight of each multi-object particle has to be updated using the multi-object likelihood function (15.7). The usage of a hypotheses tree [12–14] facilitates an intuitive representation of all valid association hypotheses. An example of a hypotheses tree is illustrated by Fig. 15.2 for a scenario with two objects and two measurements. A complete association hypothesis for a multi-object particle corresponds to the path from the root of the tree to a leaf. Since different cardinalities are represented by additional hypotheses trees and the assignment of measurements to the clutter source is realized by the

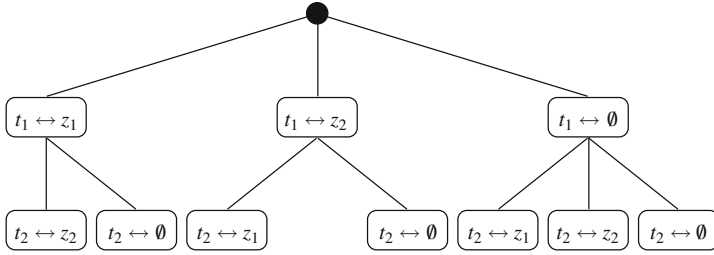


Fig. 15.2 Hypotheses tree for a scenario with two objects, t_1 and t_2 , and two measurements, z_1 and z_2 . Each node represents an association of the object t_i to measurement z_j (i.e., $\theta(i) = j$) or the missed detection \emptyset (i.e., $\theta(i) = 0$)

factor $\pi_c(Z)$, the hypotheses tree for the SMC-MOB filter is less complex than the one for the joint integrated probabilistic data association (JIPDA) algorithm in [12, 14]. The likelihood of an association $\theta(i)$ corresponds to an edge of the hypotheses tree. The value of each summand of (15.7) is calculated by multiplying the edge likelihoods from the root of the tree to the corresponding leaf. The likelihood of the measurement Z for the multi-object particle $X_+^{(i)}$ is obtained by accumulating the likelihoods for all paths and a subsequent multiplication with the clutter factor $\pi_c(Z)$ and the missed detection factor $\pi(\emptyset|X_+^{(i)})$. Similarly to [12, 14], the likelihoods for all track-to-measurement assignments are calculated a priori and stored in a lookup table. The implementation of the hypotheses tree using recursion is straightforward.

The update step of the SMC-MOB filter does not affect the state of the multi-object particles, i.e. the posterior multi-object particles are identical to the predicted ones:

$$X^{(i)} \triangleq X_+^{(i)}. \quad (15.19)$$

However, the weights of the multi-object particles are updated using the multi-object likelihood function:

$$w^{(i)} \triangleq \frac{\pi(Z|X_+^{(i)})}{\sum_{e=1}^v \pi(Z|X_+^{(e)})}. \quad (15.20)$$

Here, the denominator is a normalizing constant which ensures that the weights still sum up to 1 after the update.

After several measurement updates, the weights typically tend to concentrate on one or only a few multi-object particles since the prediction step increases the variance of the particles and the update does not decrease the variance. Hence, standard resampling approaches used for SMC implementation of the single-object Bayes have to be applied [25, 27].

15.3.3 Modeling Object Interactions

The standard multi-object motion model given by (15.7) assumes that the motion of each object depends only on its current state and the assumed motion model, i.e. the objects are considered to be statistically independent. Especially in scenarios with closely spaced objects, this assumption leads to physically impossible multi-object states after prediction. In the context of a *Companion-System*, typical examples of these impossible states are multi-object particles

In order to avoid invalid multi-object states, an appropriate model for human motion is required. In [6], Henderson observed correlations between fluid dynamics and human motion. However, this approach facilitates only a macroscopic formulation which, e.g., delivers the mean velocity of a group of people. In contrast, the Social Force Model proposed by Helbing and Molnar [5] uses a microscopic model to represent human motion where changes of the individual behaviors due to the current environment are modeled by attractive and repellent force vectors. Repellent forces are typically used to avoid collisions with other persons as well as static obstacles. Attractive forces are used to model the destination of a person. Further, the model is based on the knowledge that each person tries to reach its destination on the shortest path while moving with its desired velocity.

In addition to scenarios with closely spaced objects, the incorporation of object interaction is also recommended for scenarios with occlusions or in the case of low measurement rates. In Sect. 15.3.3.1, an approach based on the incorporation of physical constraints is proposed which avoids collisions of the persons and may be realized without any additional information. Further, Sect. 15.3.3.2 outlines the possibilities to improve the tracking results by using the information available in the *Companion-Systems* knowledge base.

15.3.3.1 Set-Based Weight Adaption

In order to obtain only valid predicted multi-object states, an incorporation of object inter-dependencies in the transition densities is required. Since the computation of such transition densities is computationally demanding, the proposed method predicts all objects within a multi-object particle independently, and a subsequent weight adaption of the multi-object particles is applied to remove invalid ones.

The weight adaption is based on the repellent forces used in the social force model, which are modeled using exponential functions [17]. In the case of circular objects with radius r_p , the likelihood that a multi-object particle comprises two objects s and t follows

$$\Lambda_d(x^{(s)}, x^{(t)}) = \begin{cases} 0 & \text{if } d(x^{(s)}, x^{(t)}) < 2r_p \\ 1 - \exp\left(-\frac{(d(x^{(s)}, x^{(t)}) - 2r_p)^2}{2\sigma_d^2}\right) & \text{otherwise,} \end{cases} \quad (15.21)$$

where $d(x^{(s)}, x^{(t)})$ denotes their Euclidean distance. Obviously, a likelihood of 0 is assigned if two objects are overlapping and the exponential function facilitates a smooth transition of the likelihood function for all distances up to the preferred inter-object distance. Afterwards, the weight of the multi-object particle $X^{(i)}$ is adapted using the minimum likelihood of all possible pairings (s, t) :

$$\tilde{w}_+^{(i)} = \min_{s=1, \dots, |X^{(i)}|} \left(\min_{t=1, \dots, |X^{(i)}|, t \neq s} (\Lambda_d(x^{(s)}, x^{(t)})) \right) \cdot w_+^{(i)}. \quad (15.22)$$

The weight of a multi-object particle is set to 0 if any two of its objects are colliding. In contrast, the weight of the multi-object particle is unchanged if the distances between all of its objects are higher than the preferred distance.

15.3.3.2 Integration of Destinations Using Knowledge Base

A *Companion*-System comprises a multitude of different components, each of which can potentially produce and/or consume information (see Fig. 15.3). As decision making and inference across different modules has to be kept consistent, a central probabilistic *knowledge base* (KB) is tasked with maintaining a global filtered probabilistic belief state X_{KB} . Consistency requires that the local belief state X_C of a component C correspond to the marginalization of X_{KB} over the variables not included in C . To achieve this global synchronization, probabilistic state information may flow bidirectionally between the KB and the interfacing modules. By maintaining a globally consistent belief state, the knowledge base provides mutual abstraction between all interfacing components, such that every module has to deal only with its local view of the global state. Such joint treatment of belief across components fosters synergistic effects, which may also improve the state prediction of multi-object trackers. Information originating from high-level components can be used to improve track continuity—in particular in situations

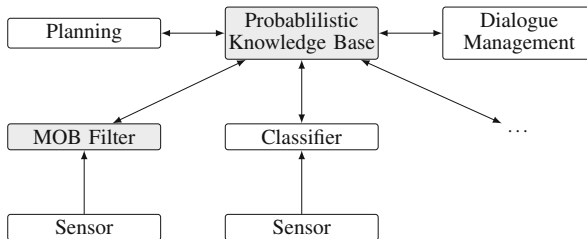


Fig. 15.3 Architecture of a prototypical *Companion*-System. The central knowledge base maintains a filtered belief state. It integrates between lower-level sensor processing modules, like the multi-object tracker and further classifier, and high-level functionality, including decision making/planning and routines of the user interface

featuring occlusions or low measurement rates, where the associations of the tracks between the individual time steps are ambiguous.

Basically every correlation between the global belief state X_{KB} and the true location of a certain object/user can be used to improve the association quality. We can identify several sources of potentially useful information, although a large part of the model is application-dependent. As human users are supposed to interact with the system, we can harvest this interaction to gain hints about their true location. A registered touch event at a stationary device gives a strong indication that a user instead of a non-user is standing in front of the device. Further, knowledge about screen content increases the chance that a user will be moving toward the screen to read the information, even if no touch contact takes place. Besides information originating from the dialog management, we can also use knowledge obtained from planning. The planning component maintains a future course of action for the user [2] to follow. If some actions are known to be connected to a certain location, we can exploit this knowledge to improve the tracking accuracy. An exemplary situation happens when the system issues a job to a printer, and the user is supposed to fetch the produced document. Then this knowledge, in combination with knowledge of the location of the printer, can be used to disambiguate which observed object corresponds to the current user.

To exploit such hints on the true user location, one can follow two approaches. One approach consists of the multi-object tracking algorithm maintaining the track labels only in cases of high confidence, trying to avoid any wrong associations. This results in many spawned tracks belonging to the same object in challenging situations. Then a track to person association can be maintained by the knowledge base using available background information. The feasibility of this approach has been demonstrated successfully in an experimental setting [3, 4] using a probabilistic model formulated with Markov Logic [24]. One major disadvantage could be identified in the requirement of discretizing the user position, because of the limitations of the modeling language. The second approach is to improve the performance of the multi-object tracking algorithm itself by integrating the hints about current and future positions of human users. The (typically imprecise) knowledge about future destinations of the user may be used within the Social Force Model to improve the predicted state of the users, which is expected to significantly improve the performance in case of long term occlusions.

15.3.4 Real-Time Implementation

An implementation of the SMC-MOB filter requires a very large number of multi-object particles to obtain a sufficient approximation of the multi-object posterior. The reason for this is that the dimension of the space of the multi-object particles is given by the dimension of the state vector times the number of objects in the scene. The prediction and update steps of the SMC-MOB filter facilitate a massively parallel implementation since the calculations for each multi-object particle do not

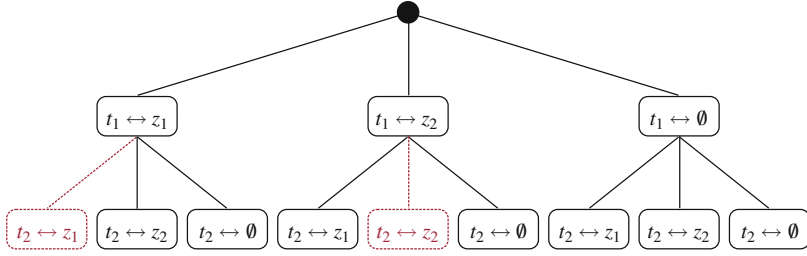


Fig. 15.4 Approximate multi-object likelihood function: compared to Fig. 15.2, two additional nodes (marked by red dashed lines) have been added which facilitate the assignment of one measurement to multiple tracks

depend on any other multi-object particle. Consequently, graphics processing units (GPUs) are well suited for the implementation of the SMC-MOB filter.

Due to the combinatorial complexity and the restrictions concerning recursive functions on GPUs, an exact computation of the multi-object likelihood function is only feasible for a limited number of tracks and measurements. The reason for the complexity is the assumption that a measurement is created by at most one object. Neglecting this assumption, the multi-object likelihood function simplifies to [19]

$$\tilde{g}(Z|X_+) = \pi_C(Z)\pi(\emptyset|X_+) \cdot \prod_{i=1}^n \left(1 + \sum_{j=1}^m \frac{p_D(x_+^{(i)}) \cdot g(z_j|x_+^{(i)})}{(1 - p_D(x_+^{(i)})) \lambda_{cC}(z_j)} \right), \quad (15.23)$$

i.e. the multi-object likelihood may be calculated using two for loops and the computational complexity reduces to $\mathcal{O}(mn)$, where n is the number of objects and m is the number of measurements. The corresponding hypotheses tree for two measurements and two tracks is depicted by Fig. 15.4. Obviously, the approximation leads to two additional nodes in the tree (marked by dashed lines) and the approximation error is negligible if each measurement has a significant likelihood for at most one object, i.e. the association hypotheses due to the two additional nodes have an insignificant contribution to the multi-object likelihood function. Modeling object interactions as presented in Sect. 15.3.3; the approximation errors are negligible if the extent of the objects is significantly larger than the standard deviation of the measurement noise.

In [21, 22], it is shown that the proposed approximation of the multi-object likelihood function facilitates a real-time capable implementation of the SMC-MOB filter using a GPU. With a total number of 25,000 multi-object particles, an Nvidia Tesla C2075 GPU processes the prediction step, the update step and the track extraction in less than 40 ms for a scenario with up to seven objects.

15.4 Labeled Multi-Bernoulli Filter

The SMC-MOB filter introduced in the previous section requires a huge number of multi-object particles since the dimension of the sample space increases linearly in the number of objects. Since the required number of multi-object particles for a sufficient state representation grows exponentially with the state dimension, the maximum number of objects in the scene is limited due to the available computational resources. Hence, alternative approaches are required to handle large numbers of objects.

In [29], Vo and Vo showed that the class of δ -generalized labeled multi-Bernoulli (δ -GLMB) RFSs is closed under the prediction and update equations of the multi-object Bayes filter¹ for the standard multi-object motion model as well as the standard multi-object likelihood. Hence, the δ -GLMB filter [29] facilitates an analytical implementation of the multi-object Bayes filter. Similarly to the SMC-MOB filter, the number of components required within the δ -GLMB filter is combinatorial. The labeled multi-Bernoulli (LMB) filter [19, 23] approximates the δ -GLMB distribution by an LMB distribution which facilitates the tracking of a huge number of objects due to the application of principled approximations.

15.4.1 Labeled Random Finite Sets

The SMC-MOB filter as well as the PHD, CPHD, and CB-MeMber filters require a (typically heuristic) post-processing to extract object tracks out of the estimated multi-object probability density. The underlying idea of the class of labeled RFSs is to augment the state by track labels. Thus, filtering a labeled RFS over time delivers a joint estimate of the number of tracks, their individual positions as well as their trajectories.

In a labeled RFS, each object's state $x \in \mathbb{X}$ is augmented by a label $\ell \in \mathbb{L}$, where \mathbb{L} is a discrete label space (e.g., the set of positive integers \mathbb{N}). Consequently, a labeled multi-object state is represented by the set $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$ on the space $\mathbb{X} \times \mathbb{L}$, where the labeled state vectors are abbreviated using $\mathbf{x} = (x, \ell)$. In multi-object tracking applications, it is required that the object labels are distinct, i.e. a label ℓ may be assigned to at most one object in each realization. In order to ensure distinct labels within each realization of a labeled RFS \mathbf{X} , the distinct label indicator [29]

$$\Delta(\mathbf{X}) = \delta_{|\mathbf{X}|}(|\mathcal{L}(\mathbf{X})|) \quad (15.24)$$

¹Observe that the number of components of the δ -GLMB distribution increases during these steps.

requires the cardinality of a realization \mathbf{X} to be equal to the number of distinct track labels $|\mathcal{L}(\mathbf{X})|$, where the set of track labels is given by the projection

$$\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \quad (15.25)$$

with $\mathcal{L}(\mathbf{x}) = \mathcal{L}((x, \ell)) = \ell$.

15.4.1.1 Labeled Multi-Bernoulli Random Finite Set

The representation of the uncertainty about object existence is intuitively realized using a Bernoulli RFS \mathbf{X} . With the existence probability r , the Bernoulli RFS is given by a singleton. Consequently, the RFS \mathbf{X} corresponds to the empty set with probability $1 - r$. The probability density of a Bernoulli RFS follows [11, pp. 368]

$$\pi(\mathbf{X}) = \begin{cases} 1 - r, & \text{if } \mathbf{X} = \emptyset, \\ r \cdot p(x), & \text{if } \mathbf{X} = \{x\}, \end{cases} \quad (15.26)$$

where $p(x)$ is the spatial distribution of the object on the space \mathbb{X} . Obviously, the cardinality distribution follows a Bernoulli distribution with parameter r . A multi-Bernoulli distribution \mathbf{X} [11] is the union of M independent Bernoulli RFSs $\mathbf{X}^{(i)}$, i.e. $\mathbf{X} = \cup_{i=1}^M \mathbf{X}^{(i)}$.

By interpreting the component indices of the multi-Bernoulli distribution as track labels, the LMB RFS [29] is obtained which is completely defined by the parameter set

$$\boldsymbol{\pi}(\mathbf{X}) = \{(r^{(\ell)}, p^{(\ell)})\}_{\ell \in \mathbb{L}}. \quad (15.27)$$

Using the multi-object exponential notation, an LMB RFS is expressed by

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) w(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}}, \quad (15.28)$$

where $h^{\mathbf{X}} = \prod_{x \in \mathbf{X}} h(x)$ and $h^{\emptyset} = 1$. The weights of the realizations are given by the multi-Bernoulli distribution

$$w(L) = \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}}, \quad (15.29)$$

and the spatial distributions are $p(x, \ell) = p^{(\ell)}(x)$. An example for an LMB RFS is illustrated by the upper part of Fig. 15.5.

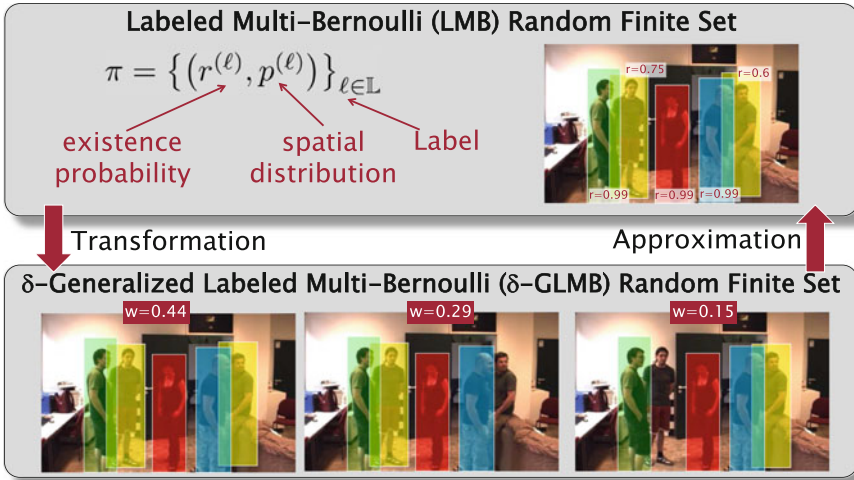


Fig. 15.5 Representation of the multi-object state using LMB and δ -GLMB RFSs. An LMB RFS can be equivalently rewritten in δ -GLMB form. In contrast, a δ -GLMB RFS can only be approximated using an LMB RFS

15.4.1.2 δ -Generalized Labeled Multi-Bernoulli Random Finite Set

An LMB RFS facilitates exactly one realization for a given set of track labels due to the assumption of statistical independence of the tracks. In contrast, a δ -GLMB RFS provides the possibility of several realizations for each set I of track labels. The distribution of a δ -GLMB RFS is given by

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \mathcal{E}} w^{(I, \xi)} \delta_I(\mathcal{L}(\mathbf{X})) [p^{(I, \xi)}]^\mathbf{X}, \quad (15.30)$$

where ξ denotes the history of track-to-measurement associations. Thus, the δ -GLMB RFS is able to represent the ambiguity in the track-to-measurement association during the filter update using several components or hypotheses for each set of track labels.

The difference between an LMB RFS and a δ -GLMB RFS is depicted by Fig. 15.5 (observe that only a subset of all hypotheses of the δ -GLMB RFS is shown). While the LMB RFS requires the tracks to be statistically independent, the δ -GLMB RFS facilitates the representation of statistical dependencies. Since an LMB RFS is a special case of a δ -GLMB RFS, it can be transformed into the corresponding δ -GLMB representation. In contrast, a δ -GLMB RFS can only be approximated by an LMB RFS.

15.4.2 Implementation of the Labeled Multi-Bernoulli Filter

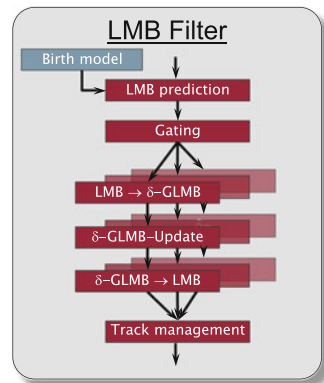
The labeled multi-Bernoulli (LMB) filter is based on the representation of the multi-object state using an LMB RFS. A complete cycle of the LMB filter is conceptually illustrated by Fig. 15.6. In the following, the main ideas behind the LMB filter are outlined for an implementation using GMs. For additional details, the derivation of the filter, and SMC implementations, refer to [19, 23].

In the prediction step, the Bernoulli distribution of each track ℓ is predicted independently. First, the spatial distribution of the track is predicted using the well-known Kalman filter equations. In the case of slightly non-linear motion models, the corresponding equations of the extended Kalman filter (EKF) or unscented Kalman filter (UKF) have to be applied. The prediction of the track's existence probability is realized by multiplying the posterior existence probability with the survival probability p_s . Finally, the tracks of the birth distribution have to be appended to the predicted LMB RFS.

To reduce the computational complexity of the filter update, the predicted LMB density is partitioned using a grouping procedure. The grouping procedure returns groups of closely spaced objects and their associated measurements where the groups can be assumed to be statistically independent in case of sufficiently large gating values. Thus, the filter update can be applied to each group independently, which significantly reduces the computational load [23].

The grouping procedure enables parallel processing of each group during filter update. The update of each group is performed as follows: After transforming the LMB RFS of each group to δ -GLMB form, the full δ -GLMB update [29] is applied, which results in several hypotheses for each set of track labels due to the ambiguity of the track-to-measurement association. The hypotheses are again given by the tree in Fig. 15.2, where each path from the root to a leaf represents a single association hypothesis. In order to reduce computational load, only the k best association hypotheses are evaluated for large groups using Murty's algorithm [15]. After calculating the updated hypotheses, the posterior δ -GLMB density of

Fig. 15.6 LMB filter schematic



each group is approximated by an LMB RFS. The approximation matches the first moment of the δ -GLMB density, i.e. the spatial distribution and the mean value of the cardinality distribution of the approximation are equivalent while the cardinality distribution itself differs. Finally, the LMB RFSs of the groups are merged and the subsequent track management module is extracting track estimates and pruning tracks with very small existence probabilities.

15.5 Conclusion

This chapter presented two multi-object tracking algorithms based on random finite sets, which are suitable to track all humans in the proximity of a *Companion-System*. The SMC-MOB filter facilitates the integration of object-interactions as well as the information of a knowledge-base in the filtering algorithm. In contrast, the LMB filter requires significantly smaller computational resources and is capable of tracking even huge numbers of objects. The table in Fig. 15.7 summarizes the differences of the two filters and illustrates that the choice for the most convenient tracking algorithm strongly depends on the scenarios that should be handled.

The presented multi-object tracking algorithms facilitate an adaption of the *Companion-System*'s behavior to the current environment. Examples for the adaption are given in the context of the demonstration scenario 3 (see Chap. 25), e.g., activation

	LMB	SMC-MOB (approx.)
Approximation Type	- parameters of labeled multi-Bernoulli distribution	- multi-object particles
Approximations	- approximation of posterior using LMB RFS - truncation of update hypotheses using Murty's algorithm	- exact representation requires huge amount of particles - likelihood approximation
Complexity (prediction/update)	- linear / cubic	- linear (one per multi-object particle) / linear
Pros	- integrated track labeling - reduced complexity due to grouping - tracking huge number of objects	- modeling of object interactions
Cons	- track coalescence possible	- number of particles grows exponentially in number of objects

Fig. 15.7 Comparison of LMB and SMC-MOB filters

of the system or including the group size into the purchase process. Further, the continuous tracking of the user provides the possibility to resume previously interrupted interactions. Other possibilities include the adaption of the input and output modalities to the current situation.

Acknowledgements This work was done within the Transregional Collaborative Research Centre SFB/TRR 62 “*Companion-Technology for Cognitive Technical Systems*” funded by the German Research Foundation (DFG).

References

1. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic, New York (1988)
2. Biundo, S., Bercher, P., Geier, T., Müller, F., Schattenberg, B.: Advanced user assistance based on AI planning. *Cogn. Syst. Res.* **12**(3-4), 219–236 (2011). doi:10.1016/j.cogsys.2010.12.005. <http://www.sciencedirect.com/science/article/B6W6C-51TGG9K-1/2/4f8891da447f7be776d553396911a589>. Special Issue on Complex Cognition
3. Geier, T., Biundo, S., Reuter, S., Dietmayer, K.: Track-person association using a first-order probabilistic model. In: IEEE 24th International Conference on Tools with Artificial Intelligence, vol. 1, pp. 844–851 (2012). doi:10.1109/ICTAI.2012.118
4. Geier, T., Reuter, S., Dietmayer, K., Biundo, S.: Goal-based person tracking using a first-order probabilistic model. In: Proceedings of the Ninth UAI Bayesian Modeling Applications Workshop (UAI-AW 2012) (2012)
5. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. *Phys. Rev. E* **51**, 4282–4286 (1995). doi:10.1103/PhysRevE.51.4282
6. Henderson, L.F.: The statistics of crowd fluids. *Nature* **229**, 381–383 (1971)
7. Kalman, R.E.: A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.* **82**(Series D), 35–45 (1960)
8. Ma, W.K., Vo, B.N., Singh, S., Baddeley, A.: Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach. *IEEE Trans. Signal Process.* **54**, 3291–3304 (2006)
9. Mahler, R.: Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.* **39**(4), 1152–1178 (2003). doi:10.1109/TAES.2003.1261119
10. Mahler, R.: PHD filters of higher order in target number. *IEEE Trans. Aerosp. Electron. Syst.* **43**(4), 1523–1543 (2007). doi:10.1109/TAES.2007.4441756
11. Mahler, R.: Statistical Multisource-Multitarget Information Fusion. Artech House Inc, Norwood (2007)
12. Mählich, M.: Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten. Ph.D. thesis, Ulm University (2009)
13. Maskell, S.: Sequentially structured Bayesian solutions. Ph.D. thesis, Cambridge University (2004)
14. Munz, M., Mählich, M., Dietmayer, K.: Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems. *IEEE Intell. Transp. Syst. Mag.* **2**(1), 6–17 (2010). doi:10.1109/MITS.2010.937293
15. Murty, K.: An algorithm for ranking all the assignments in order of increasing cost. *Oper. Res.* **16**, 682–687 (1968)
16. Musicki, D., Evans, R.: Joint integrated probabilistic data association: JIPDA. *IEEE Trans. Aerosp. Electron. Syst.* **40**(3), 1093–1099 (2004). doi:10.1109/TAES.2004.1337482
17. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You’ll never walk alone: Modeling social behavior for multi-target tracking. In: Proceedings of the 12th IEEE International Conference on Computer Vision, pp. 261–268 (2009). doi:10.1109/ICCV.2009.5459260

18. Reid, D.: An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* **24**(6), 843–854 (1979). doi:10.1109/TAC.1979.1102177
19. Reuter, S.: Multi-object tracking using random finite sets. Ph.D. thesis, Ulm University (2014)
20. Reuter, S., Dietmayer, K.: Pedestrian tracking using random finite sets. In: Proceedings of the 14th International Conference on Information Fusion, pp. 1–8 (2011)
21. Reuter, S., Dietmayer, K., Handrich, S.: Real-time implementation of a random finite set particle filter. In: Proceedings of the 6th Workshop Sensor Data Fusion: Trends, Solutions, Applications (2011). <http://www.user.tu-berlin.de/komm/CD/paper/100149.pdf>
22. Reuter, S., Wilking, B., Wiest, J., Munz, M., Dietmayer, K.: Real-time multi-object tracking using random finite sets. *IEEE Trans. Aerosp. Electron. Syst.* **49**(4), 2666–2678 (2013). doi:10.1109/TAES.2014.6619956
23. Reuter, S., Vo, B.T., Vo, B.N., Dietmayer, K.: The labeled multi-Bernoulli filter. *IEEE Trans. Signal Process.* **62**(12), 3246–3260 (2014)
24. Richardson, M., Domingos, P.: Markov logic networks. *Mach. Learn.* **62**(1–2), 107–136 (2006). doi:10.1007/s10994-006-5833-1. <http://www.springerlink.com/index/10.1007/s10994-006-5833-1>
25. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House Inc., Norwood (2004)
26. Sidenbladh, H., Wirkander, S.L.: Tracking random sets of vehicles in terrain. In: Conference on Computer Vision and Pattern Recognition Workshop, p. 98 (2003). doi:10.1109/CVPRW.2003.10097
27. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, Cambridge (2005)
28. Vo, B.N., Ma, W.K.: The Gaussian mixture probability hypothesis density filter. *IEEE Trans. Signal Process.* **54**(11), 4091–4104 (2006). doi:10.1109/TSP.2006.881190
29. Vo, B.T., Vo, B.N.: Labeled random finite sets and multi-object conjugate priors. *IEEE Trans. Signal Process.* **61**(13), 3460–3475 (2013)
30. Vo, B.N., Singh, S., Doucet, A.: Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Trans. Aerosp. Electron. Syst.* **41**(4), 1224–1245 (2005). doi:10.1109/TAES.2005.1561884
31. Vo, B.T., Vo, B.N., Cantoni, A.: Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Trans. Signal Process.* **55**(7), 3553–3567 (2007). doi:10.1109/TSP.2007.894241
32. Vo, B.T., Vo, B.N., Cantoni, A.: The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Process.* **57**(2), 409–423 (2009). doi:10.1109/TSP.2008.2007924
33. Zajic, T., Mahler, R.: A particle-systems implementation of the PHD multitarget-tracking filter. In: Signal Processing, Sensor Fusion, and Target Recognition XII, SPIE, vol. 5096, pp. 291–299, Bellingham, WA (2003)