Multi-target tracking with credal classification and kinematic data

Samir Hachour, François Delmotte, David Mercier, Eric Lefevre

Multi-target tracking with credal classification and kinematic data

Samir Hachour[a,b], François Delmotte[a,b], David Mercier[a,b], Eric Lefèvre[a,b]

[a] *Univ Lille Nord de France, F-59000 Lille, France*
[b] *UArtois, LGI2A, F-62400, Béthune, France*

Abstract

This article proposes a method to classify multiple maneuvering targets at the same time. This task is a much harder problem than classifying a single target, as sensors do not know how to assign captured measurements to known targets. This article extends previous results scattered in the literature and unifies them in a single global framework with belief functions. Through two examples, it is shown that the full algorithm using belief functions improves results obtained with standard Bayesian classifiers and that it can be applied to a large variety of applications.

Keywords: Multi-target tracking, credal classification, data assignment, targets management.

1. Introduction

The problem of joint multi-target tracking and classification, which is as old as the invention of radars, is a much more complex task than the problem of tracking one target. Indeed it includes a step, called the assignment problem, where sensors have to associate known objects to new captured measurements. A full multi-target tracking solution includes several interlaced components such as the tracking component, the assignment component, the hypothesis rejection (new or disappeared targets...) and finally the classification step. Probability-based solutions already exist in the literature [1, 2, 3, 4, 5, 6, 7].

For about 40 years, other uncertainty models based on non-additive measures have been developed, in particular the Transferable Belief Model [8] based on belief functions [9, 10] which is sometimes referred to as the credal model. Applications of this model can be found for example in classification tasks and decision support systems [11]. Direct comparisons with Bayesian solutions are presented in [12, 13] with discrete variables or in [14] with continuous variables.

Applications of this theory to multi-target tracking and classification problems are scattered through several articles presenting different approaches. In [15, 13] and in

Email addresses: `samir_hachour@ens.univ-artois.fr` (Samir Hachour), `francois.delmotte@univ-artois.fr` (François Delmotte), `david.mercier@univ-artois.fr` (David Mercier), `eric.lefevre@univ-artois.fr` (Eric Lefèvre)

[16] various distances between belief functions are used to tackle the assignment problem. Reference [15] has been recently improved in [17], but these two references only tackles the assignment problem with uncertain measurements and do not cope with the tracking problem. In [14] a solution to track and classify a single dynamical target is proposed, but no extension to multi-target tracking is proposed.

The aim of this article is to gather these scattered results, to unify them in a single and consonant framework based on belief functions, and to propose a solution for multi target tracking using belief functions when no one exists in the recent literature. The proposed solution also includes a step of hypothesis rejection, which means that it manages new and disappeared targets.

The result is a complete solution to multi-target tracking and classification in a cluttered environment. It mimics the standard and well accepted Bayesian solutions, but it extends them, when possible, with belief functions. A short preliminary version of this article was presented in [18].

The well known Interacting Multiple Model (IMM) algorithm is used to track multiple targets. The assignment problem of real measurements with known targets is resolved by the means of a generalized Global Nearest Neighbor algorithm. Target management is ensured by a score function representing the quality of target tracks. Finally, the classification step is realized with the Transferable Belief Model instead of a classical Bayesian solution.

This article is organized as follows. An introduction to belief function theory is presented in Section 2. Section 3 deals with tracking problems. Assignment and hypothesis rejection problems are tackled in Section 4. Bayesian and proposed algorithms are summarized in Section 5. Finally, two application examples are detailed in Section 6. The first one involves an academic example on aircraft classification with constant classes, it allows a comparison with a Bayesian solution. The second example concerns a pedestrian activity recognition, it highlights a first extension of the proposed algorithm to time varying classes.

2. Belief functions

2.1. Main functions

This section introduces basic notions on belief function theory, which was firstly introduced by Dempster in [9] and extended by Shafer [10] and Smets [8]. Knowledge is expressed on a discrete set $C = \{c_1, c_2, ..., c_{nc}\}$ of nc mutually exclusive and exhaustive hypotheses. Frame C is called the *frame of discernment*. A mass $m(A)$ with $A \subseteq C$ is the part of belief supporting A that, due to a lack of information, cannot be given to any strict subset of A [8]. A mass function m (or basic belief assignment) has to satisfy:

$$\sum_{A \subseteq C} m(A) = 1 \ . \tag{1}$$

Throughout this article, 2^C represents all the subsets of C. A set A such that $m(A) > 0$ is called a *focal element* of m.

In addition to the mass function, two other functions are defined in the following manner. The plausibility function Pl represents the total amount of belief that may be given to a subset A of C with further pieces of evidence:

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B) . \qquad (2)$$

Unlike the plausibility function, the belief function Bel represents the amount of belief that is certain and cannot be reduced:

$$Bel(A) = \sum_{A \supseteq B} m(B) . \qquad (3)$$

These functions are in one-to-one correspondence [10], so they are used indifferently with the same term ***belief function*** when the context is clear.

A belief function whose focal elements are singletons is called a *Bayesian belief function*, it corresponds to a probability distribution and respects the property of additivity:

$$Pl(A \cup B) = Pl(A) + Pl(B) , \qquad (4)$$

with $(A, B) \subseteq 2^C$ and $A \cap B \neq \emptyset$. In general, this relation is false and belief functions are non-additive measures.

A belief function such that $m(C) = 1$ respects $Pl(A) = 1$ for all A subsets of C, $A \neq \emptyset$. Denoted by m_0, it is called the *vacuous belief function* and represents the full ignorance.

2.2. *Fusion rule and discounting*

When more than one mass function is expressed on the same frame of discernment, they can be fused to obtain a single representation. The conjunctive combination used in this work assumes independent and absolutely reliable sources. Let m_1 and m_2 be two mass functions provided by two distinct sources and expressed on the same frame of discernment C, their conjunctive combination is defined as follows:

$$m_{12}(A) = (m_1 \textcircled{\scriptsize $\cap$} m_2)(A) = \sum_{A_1 \cap A_2 = A} m_1(A_1) m_2(A_2) . \qquad (5)$$

Equation (5) is the unnormalized rule, the normalized rule is referred to as Dempster's rule of combination, and is defined by:

$$m_{12}(A) = \frac{\sum_{A_1 \cap A_2 = A} m_1(A_1) m_2(A_2)}{1 - \sum_{A_1 \cap A_2 = \emptyset} m_1(A_1) m_2(A_2)} . \qquad (6)$$

This rule is used in order to update *a priori* beliefs with online measurements, as in the following Section 2.4.

The last example in this article involves the discounting of a source of information. Such a discounting assumes that you can estimate the reliability of a source by a factor $\lambda \in [0, 1]$. If $\lambda = 1$, the source is considered as absolutely reliable, while if $\lambda = 0$, the source must be discarded and replaced by a vacuous belief. Thus the discounting m^λ of a source m is defined by:

$$\begin{cases} m^\lambda(A) = \lambda\, m(A) & \text{if } A \neq C\ , \\ m^\lambda(C) = \lambda\, m(C) + 1 - \lambda & \text{otherwise.} \end{cases} \tag{7}$$

If several sources of information m_i have to be fused, each one with its own reliability λ_i, a classical approach is to first discount all of them, and then to conjunctively fuse them using Equation (5). Several other fusion rules are defined, and for a review, readers can refer to [19]. For example, contextual data can be included also in the fusion process, see [20], and reliability factors can be adapted online [21], although this is not used in this article.

2.3. Decision rule

Several belief function interpretations exist, among them the Upper/Lower Probabilities (ULP), usually called imprecise probabilities, and the Transferable Belief Model (TBM) of Smets. Basically these models are equal when considering static knowledge, but differ when conditioning steps are involved. Readers interested in this topic can refer to [22]. In a few words, within an ULP model, conditioning requires to condition every probability measure $P(A)$ compatible with the bounds defined by $Bel(A) \leq P(A) \leq Pl(A)$. Then, the new bounds $Bel(.|.)$ and $Pl(.|.)$ must be recomputed by checking every conditioned probability measures, and taking the new maximum and minimum, while within the TBM it suffices to condition only the original $Bel(A)$ and $Pl(A)$. Thus, the ULP model is more computationally demanding. In a recent article [23], the ULP model has been advocated in a classification problem. But the authors of this article do not find the provided examples conclusive and the advantage of the ULP model over the TBM remains unclear. Since it is more complex to use, the TBM has been chosen in this article, as in [14] for instance.

The TBM represents and manages knowledge with a two level model. The first one, referred to as the credal level, concerns the representation and the manipulation of the data. It is the place where data are encoded, combined and updated with belief functions without assuming probability measures [12]. Decisions are made when necessary at a second level called the pignistic level, where belief functions are transformed into a probability measures using the pignistic transformation justified in [24] through rationality requirements and basic axioms. Pignistic probability $BetP$ is defined by:

$$BetP(\{c_i\}) = \sum_{c_i \in A} \frac{m(A)}{\mid A \mid (1 - m(\emptyset))}. \tag{8}$$

Let us remark that other decision rules have been introduced for belief functions, among them the maximum of plausibility [25]. Comparing the drawbacks and advantages of all the decision rules is outside the scope of this article, see for instance [24].

2.4. Generalized Bayes Theorem

Bayes theorem enables to compute the *a posteriori* probability from an *a priori* one. With likelihoods $l(c_i|z) = P(z|c_i)$, where z is a measure provided by a sensor, each probability $P(c_i|z)$ can be computed by:

$$P(c_i|z) = \frac{l(c_i|z)P(c_i)}{\sum_{c_j} l(c_j|z)P(c_j)} \ . \tag{9}$$

In a recursive scheme, $P(c_i)$ becomes $P_k(c_i)$ and $P(c_i|z) = P_k(c_i|z)$, where the index k represents a given time step. Initialization, without any further information, is usually carried out with the uniform probability measure.

With the same ideas, Smets introduces a Generalized Bayesian Theorem (GBT) [26] allowing the recursive calculation of mass functions. The GBT requires two steps. Based on likelihoods $l(c_i|z)$, a conditional mass function $m(A|z)$ for each $A \subseteq C$ is firstly calculated as follows:

$$m(A|z) = \prod_{c_i \in A} l(c_i|z) \prod_{c_i \in \bar{A}} (1 - l(c_i|z)) \ . \tag{10}$$

Formulated in a recursive manner, the second step consists in conjunctively merging *a priori* and conditional beliefs with Equation (5):

$$m_k = m_k(.|z) \circledcirc m_{k-1} \ . \tag{11}$$

The *a priori* belief is initialized with the vacuous belief m_0.

GBT applications are numerous [12, 14, 27, 28, 29].

3. Target state and measurement models

3.1. Measurement model

The final goal of the credal algorithms presented in this article is to find each target class, which belongs to C, from a series of measurements. For simplicity reasons, targets measurements z_k^j, with $j = 1, 2, ..., m$ where m represents the number of measurements acquired at time k, are considered linearly dependent on the target state vectors x_k^i, with $i = 1, 2, ..., n$, where n is the number of known targets at time k. Target state vectors evolutions are described by Equation (12) and measurements are taken according to Equation (13).

$$x_k^i = F(c_s) x_{k-1}^i + B u_k^i + w_k^i \ , \tag{12}$$

$$z_k^j = \begin{cases} H x_k^i + v_k & \text{known target originated measurement,} \\ H x_k^{i'} + v_k & \text{new target or noise originated measurement,} \end{cases} \tag{13}$$

where:

- $x_k^i \in \mathbb{R}^p$ is the i^{th} target state vector with $i \in \{1, 2, ..., n\}$ representing the index of the target at time step k.
- $F(c_s)$ is a $p \times p$ state matrix depending on class c_s, with $s \in \{1, 2, ..., nc\}$.

- u_k^i is the i^{th} target deterministic input at time step k and w_k is a state Gaussian noise with covariance matrix Q. The input matrix is denoted B.
- $z_k^j \in \mathbb{R}^q$ is the j^{th} received observation at time step k, with $j \in \{1, 2, ..., m\}$.
- H is a $q \times p$ time invariant observation matrix and v_k is the measurement error, it is considered as a Gaussian noise with zero mean value and covariance matrix R.

The number of targets n is not constant over time and a given measurement z_k^j at time step k can come from a known target i, a new target i' or even from noise.

The state model parameters $< F, B, u, w >$ are given in a general manner in Equation (12). In maneuvering targets case, there will be r different model parameters sets, where r is the number of different evolution modes.

The optimal Bayesian estimation of the i^{th} target state and class at time step k requires the calculation of the following probability density function:

$$p(x_k^i, c_s | Z_k^i), \quad i = 1, ..., n \tag{14}$$

where $Z_k^i = \left\{z_1^i, z_2^i, ..., z_k^i\right\}$ represents the cumulated measurement for the target i until the time step k.

3.2. Discussion about the filters used to track

The Gaussian assumption on the state and measurement noises allows the probability density function of n maneuvering targets, represented by Equation (14), to be optimally estimated by Kalman filter based Interacting Multiple Model (IMM) algorithm. Each IMM algorithm is dedicated to track one target, it handles its r possible evolution models. Details on the derivation of the targets motion models can be found in [30, 31, 32]. More information on IMM algorithm can also be found in [30, 33]. Different than the IMM algorithm proposing an analytic estimation of the *a posteriori* probability density in Equation (14), other more recent methods exist. Based on Monte Carlo approximation of the *a posteriori* density, they are called either Probability Hypothesis Density (PHD) [34, 35, 36] filters with different Monte Carlo sampling methods, or also Particle Filter (PF) [37, 38]. The approximation of the *a posteriori* probability density, in these methods, is done by performing a set of N weighted Monte Carlo samples. The approximation is almost optimal when N tends to infinity. The major advantage of Monte Carlo based methods is the capacity to estimate the *a posteriori* probability density in Equation (14), regardless the nature of the variable x (linear or non-linear) and regardless the nature of the noise affecting the targets evolution and measurement models (Gaussian or non-Gaussian). However, these approaches need a high number of samples (particles) N to achieve an acceptable approximation. For example, Caron et *al.* in [39] use $N = 250$ particles to achieve the performance of the same kinematic data based classification presented in [14] for one target classification, and used in the first simulation example of Section 6 for multi-target classification. In their work, Caron et *al.* use the so-called Rao-Blackwellised filter, the particles in this filter are simply Kalman estimators where each Kalman estimator handles a target linear model. This means that for a kinematic data based classification problem, it is better to have a set of different linear models describing targets evolutions instead of unified non-linear one. Regarded this circumstance, Kalman filter based IMM algorithm appears much more efficient in the case of Gaussian noises.

In a simple example of linear signal estimation illustrated in Figure 1, the Kalman filter estimation error averaged on 100 different simulations has a mean value of 5.5721, while the obtained mean error with a particle filter for $N = 150$ is about 7.1466. Knowing that one particle is approximately equivalent to a Kalman filter in terms of computation complexity, this example illustrates the great need on computation resource of the particle filter. The estimation error in this comparison is calculated by: $(x - \hat{x})^2$, where the real state x and the estimated one $\hat{x}$ are scalars.

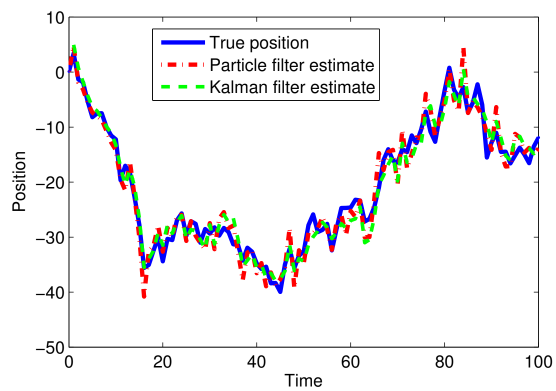

Figure 1: Kalman filter vs Particle filter.

This article is mainly devoted to the classification task. To simplify the problem only linear models are considered. Of course, if non linear models are required to describe the targets dynamic, Kalman filters, therefore IMMs, would be out performed by such particle or PHD filters. This would not impact the classification stage where only likelihoods of targets models are needed.

3.3. Difference between single object and multiple object classifications

This article deals with multiple object classification, where the difference between the number of known targets n and the number of measurements m could be strictly greater than 1.

As illustrated by the following examples, this task is much more complex in a multi-target context as the classifier has to infer the association of captured measurements z_k with already known targets.

When only one object has to be classified, the Bayesian approach provides directly the solution using Bayes rule (Equation (9)) as it can be observed in Figure 2 with P_{k-1} representing the *a priori* probabilities and P_k the *a posteriori* ones.

As illustrated in Figure 3, with more than one target to track and classify, Bayes rule can no more be applied directly and an association step is needed to assign new measurements to known targets.

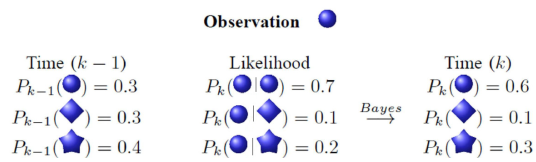

Figure 2: Illustration of a single target classification task with a Bayesian approach.

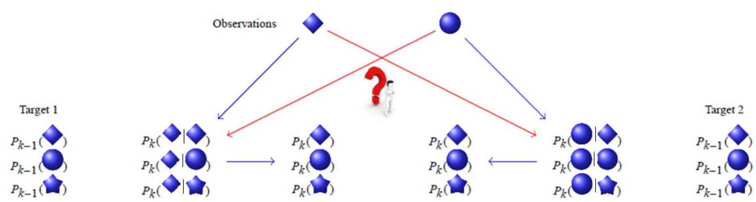

Figure 3: Illustration of the assignment question in a 2-target classification task.

A solution to the assignment problem is presented in Section 4.

Remark. As readers can notice it, states and measurements equations presented in this section are based on real numbers. This is in contrast with the work of Smets and Ristic [14] on single dynamical target classification. Indeed in their approach, the authors have decided to use belief functions at every step of the classification chain. So, they have introduced *continuous* belief functions to describe targets states. Generalized Kalman filters were used to handle continuous belief functions at that stage. This is not the case here, since classical Kalman Filter based IMMs are used. But, as readers will be able to check it, results proposed in simulation Section 6 behave similarly as the ones of Smets and Ristic. The major improvement over the Bayesian classification seems to lie in the use of belief functions at the classification stage and in particular the use of the Generalized Bayesian Theorem. The crucial importance of this component when trying to classify objects from measurements was also studied in [12].

4. Assignment problem with targets appearances and disappearances management

Assignment problem and targets management is an intermediate step between the prediction and the update steps of the IMMs. It is supplied by a set of predicted measurements and their corresponding covariance matrices, respectively, $\hat{z}^i$ and S^i, with $i = \{1, 2, ..., n\}$, of the n already known targets. For each target i, the quantities $\hat{z}^i$ and S^i are aggregations of the corresponding IMM models predicted quantities, namely $\hat{z}^i_l$ and S^i_l, where $l = 1, ..., r$ and r is the number of models in the IMM, see [30, pages 225-226] for detailed information.

Targets predicted measurements $\hat{z}^i$, with $i = \{1, 2, ..., n\}$, are compared with a set of m real measurements z^j, $j = \{1, 2, ..., m\}$ received at time step k, then an assignment problem is resolved in such a way to answer the following questions:

- *How to assign the m received real measurements z^j to the n already predicted measurements $\hat{z}^i$?*
- *How to manage targets appearances, reappearances and disappearances?*

The answers that should be given to the above questions are:

- Targets that have received an observation are updated following the IMMs update process.
- Targets that have not received any observation are considered as non-detected.
- The non-assigned measurements are used to initialize new targets.

The following paragraphs discuss the main existing assignment solutions in literature, and the adopted solution to obtain the above expected answers.

4.1. Overview of assignment solutions for multi-target tracking

The problem of false assignments in multi-target tracking and recognition is treated by researchers from various application domains. Namely, in the domain of artificial vision where the problem becomes less difficult since complementary information (texture, color, etc.) can help to distinguish the kinematic observations of the objects and get less conflicting assignment problem. Readers can refer for example to [40] where appearance information (color, texture) are added to distinguish tracked people in a video, this increases the robustness to false assignments. The same method benefits of the low calculation complexity of the Hungarian algorithm used to resolve the assignment problem, which makes it feasible online. Other works in artificial vision using complementary information (color and sound characterizing the targets) to distinguish the targets observations can be found in [41, 42].

In this work, the only available data are the kinematic ones. In this context, several categories of approaches have been developed principally with probability measures, deterministic approaches and Monte Carlo sampling based approaches.

Probabilistic approaches update each target with a weighted sum of the observations falling within its neighborhood. The weights represent the *a posteriori* probabilities that the observations are originated from the considered targets. Examples are Joint Probability Data Association (JPDA), Integrated Probability Data Association (IPDA) [43, 44, 45], etc. These approaches consider as a new target the observation having a low probability to be assigned to all the existing targets, and consider as a non-detected target, the one going to be updated with a low weighted sum of observations. Multi-Hypotheses Tracking (MHT) algorithm [5] has a different principle, it is a multi-scan approach that holds off the assignment decision until having more clear and non-conflicting data. Considered as the best approach, it is also the most computationally demanding. MHT, JPDA and IPDA computational complexities increase rapidly with the number of targets and observations. Their performances are known to be weak in a dense targets environment.

Deterministic approaches, like Nearest Neighbor (NN) which updates each target with its nearest neighbor and Global Nearest Neighbor (GNN) [3, 46] take the optimal solution which minimizes the global distance between the targets and observations.

Credal versions of the GNN algorithm are proposed in [15] and [17] but have not yet been tested in a tracking context where sensors noises are permanently changing over time. Deterministic approaches are known to present some weakness when it comes to track nearby or crossing targets (see Section 6.2). However, they are well suited for real time applications, given their simplicity, low computational complexity and their efficiency to handle appearances and disappearances, even in a dense targets environments.

More recently, Markov Chain Monte Carlo (MCMC) based methods [47, 48, 49] have been proposed to resolve the assignment problem. They are sub-optimal methods that do not carry on the statistical distribution of the variables. Their performances tend to the optimal one, when the number of performed Monte Carlo samples tends to infinity. The problem of false assignments still remains even in MCMC based methods. For example, the approach proposed in [47] presents a low rate of false assignments for multiple interacting ants tracking, thanks to the use of a targets interacting model and extended information such as ants orientations before and after crossing. MCMC methods are preferred for their general aspect (no assumption on variables statistics) but they are known to be computationally very complex.

To avoid the assignment problem in multi-target tracking, other conceptions of the problem have emerged. For example, the work presented in [50] presents a so-called ”One State Filter” method. It gathers all state vectors of the tracked targets in one extended state vector, modify accordingly all measurements models and then a single unified estimator can be designed, be it an Extended Kalman Filter (EKF) or an Unscented Kalman Filter (UKF). This way, explicit data association methods (GNN, JPDA, MHT, etc..) can be disregarded. Although very interesting, this method suffers from a major issue, as recognized by the authors: currently this approach can handle two targets only, and it seems difficult to generalize it to more. A second issue, as mentioned in [50], is that identities of crossing or nearby targets are lost. This means that in a context of kinematic data based classification approach, like the one proposed here, this approach can not be applied, because each target is classified based on its own explicit observations.

4.2. Adopted assignment solution: GNN algorithm

In a few recent references [13, 30], the standard GNN algorithm is presented based on a square matrix representing the distances between the n predicted states and the m measurements with $n = m$.

To use the standard Munkres solution with a varying number of targets (which means that possibly $n \neq m$), a more general GNN algorithm is considered in this article. Its resolution is performed by the generalized Munkres algorithm [51] in such a way to handle rectangular assignment matrices. The Generalized GNN algorithm is composed of two main steps: the generalized affectation matrix calculation and its resolution.

A general formulation of the assignment matrix is given in Table 1. In this table, $[D_{j,i}] \in R^{m\times n}$ represents a special case of Mahalanobis distance (normalized Euclidean distance), which follows a χ^2 distribution with a degree of freedom q (dimension of the measurement vector) [52]. The distance is calculated as follows:

$$D_{j,i} = (z^j - \hat{z}^i)^T (S^i)^{-1} (z^j - \hat{z}^i), \tag{15}$$

with $j = \{1, 2, ..., m\}$, $i = \{1, 2, ..., n\}$ where S^i represents target i expected measurement error covariance matrix. It is related to the covariance matrices of the Kalman filters in the corresponding IMM. Threshold T in Table 1 is drawn from the χ^2 table based on an *a priori* probability that the measurement corresponds to a new target with the degree of freedom q. Initials NT stands for New Target.

♠♠♠ Give equation to deduce or compute T: it will be great to avoid ambiguities

Real measurements	n known targets				m possible new targets			
	$\hat{z}^1$	$\hat{z}^2$	$\ldots$	$\hat{z}^n$	NT^1	NT^2	$\ldots$	NT^m
z^1	$D_{1,1}$	$D_{1,2}$	$\ldots$	$D_{1,n}$	T	∞	$\ldots$	∞
z^2	$D_{2,1}$	$D_{2,2}$	$\ldots$	$D_{2,n}$	∞	T	$\ldots$	∞
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
z^m	$D_{m,1}$	$D_{m,2}$	$\ldots$	$D_{m,n}$	∞	∞	$\ldots$	T

Table 1: General formulation of the assignment matrix.

Using modified Munkres algorithm [51], the assignment problem can be efficiently resolved and the result is used in the following manner:

- Measurements assigned to known targets are processed by the targets corresponding IMM algorithms.
- Measurements assigned to NT are used to initialize new IMMs. Let us note that a measurement assigned to NT is either originated by a real new target or by noise, so the measurement is not immediately confirmed as a new target. The confirmation is done by the means of a score function described in the following paragraph.
- Target which does not receive any measurement is updated by its predicted one (trajectory prediction). Its score will decrease up to the point it will be considered as a disappeared target. This action is ensured by the score function.

In some previous works [30], targets are confirmed or deleted depending on how often they are detected or not.

A more efficient manner to validate and delete targets is to use a score function which represents the quality of the targets tracks.

4.3. Targets appearances and disappearances management

The score function is a sequential probability ratio test. The test was first introduced by Wald in [53] and its use in the targets tracking framework is clearly detailed in [30, pages 327 − 334]. Since it is well known and clearly presented in textbooks, only the outline of the test is given next. They are two hypothesis for each tracked target: a true one, or a false one.

The log-likelihood ratio $L_i(k)$ for each target i at time step k is updated sequentially by:

$$L_i(k) = L_i(k-1) + \Delta L_i(k) \, . \tag{16}$$

Once the ratio is calculated, it is compared to two thresholds T_1 and T_2 which depend on the false target confirmation and true target deletion probabilities. Then either a target is confirmed, deleted, or the test continues until a decision is made.

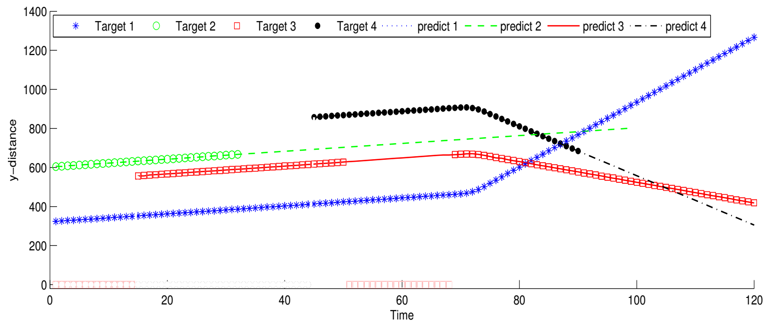

Figure 4: y-direction evolutions over time of 4 targets.

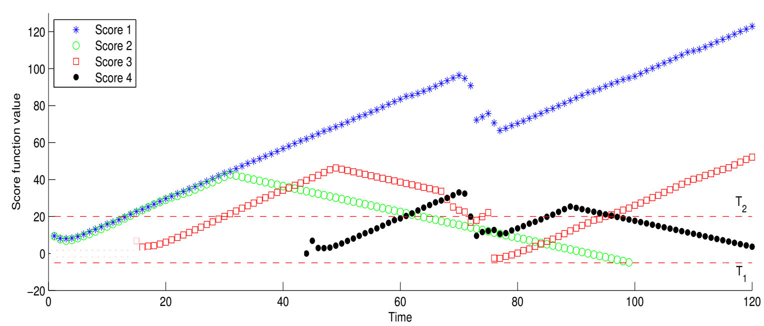

Figure 5: Targets score functions.

Example of multi-target appearances and disappearances management. This simulated scenario is based on the model described in Section 6.1.1 Equation (24). Figures 4 and 5 show respectively the y direction evolutions of four targets and the evolutions of their score functions. Difficulties managed in this simulation are summarized hereafter:

- Targets 3 and 4 appear respectively at time steps 15 and 45. Their respective score functions are initialized at the same time (cf Figure 5).

- Targets 2, 3 and 4 are not detected respectively at time steps 35, 55 and 90. Their score functions adopt a decreasing evolution. It is possible to see that target 2 is deleted at time step 100, when its score function reaches the deletion threshold T_1. In real life the missing detection can be caused among other things by a remoteness targets or a possible occlusion.

- Target 3 reappeared at time step 69 after a missing detection period, its corresponding score function increases again after a decreasing period.

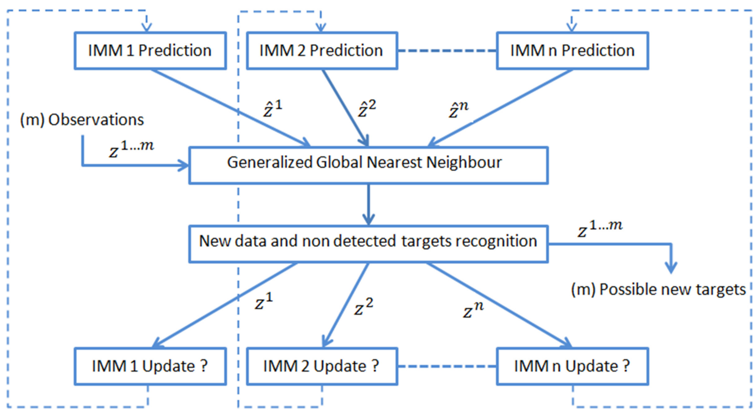

Figure 6: Multi-target tracking algorithm architecture.

Figure 6 gives an overall view of the global architecture for multi-target tracking which is used in this article. In this figure, "*IMM n Update ?*", for example, means that target n is updated only if an observation is associated to it, else it is considered as non-detected.

Next section deals with the last step of the global algorithm which is the classification step.

5. Targets classification

A kinematic data based classification is considered. The problem is firstly studied for a single target problem [6] where the provided solution is based on Bayesian model. Using the credal formalism, the Bayesian solution was then enhanced in [14]. The idea is based on the discretization of the state space in the IMMs (set of linear models), this allows the characterization of the different behaviors of the different targets types. It is considered that the IMM algorithms contain an exhaustive list of all the possible evolutions models of the considered targets. The list of models is given by:

$$M = [m_1, m_2, ..., m_r], \tag{17}$$

where r represents the total number of models.

Based on an *a priori* knowledge, the different models are clustered so that each group of models corresponds to a specific behavior. Finally, knowing the behavior(s) of a given target, its class can be determined.

The set of possible behaviors can be defined by $B = [b_1, b_2, ..., b_{nb}]$, where nb is the number of behaviors. The set of models belonging to behavior b_i is defined by $M_{b_i} \subseteq M$, with $i = 1, ..., nb$. The number of models in M_{b_i} is noted by r_{b_i}.

As an example, in pedestrian recognition, if the set M contains 5 constant velocity models in an increasing order, the behavior of slowly walking mode can gather the 2 first models, which can correspond to the class of old persons for instance.

Bayesian and credal classification methodologies used in this article are now described in the sequel. Note that for simplicity reasons, targets indices have been removed.

5.1. Bayesian classification

5.1.1. Behaviors likelihoods calculation

At each time step k, each IMM provides the *a posteriori* probability μ_j and the likelihood Λ_j concerning the r different models m_j in M, where $j = 1, \ldots, r$. Models likelihoods for each given target evolution are calculated as follows:

$$\Lambda_j = \frac{exp[-d_j^2/2]}{\sqrt{(2\pi)^q |S_j|}}, \tag{18}$$

where d_j^2 is a squared Euclidean distance between the j^{th} model predicted measurement $\hat{z}_j$ and its assigned real measurement, it is calculated as in Equation (15), q is the measurement dimension and S_j is the j^{th} model expected measurement error covariance matrix.

Once the probabilities and likelihoods of the models are obtained, the clustering described above is adopted to determine the behaviors likelihoods l_{b_i}:

$$l_{b_i} = \sum_{j:m_j \in M_{b_i}} \mu'_j \Lambda_j \ , \quad i = 1, ..., nb \ , \tag{19}$$

with:

$$\mu'_j = \frac{\mu_j}{\sum\limits_{l:m_l \in M_{b_i}} \mu_l} \ , \quad l = 1, ..., r_{b_i} \ . \tag{20}$$

Based on the already calculated behaviors likelihoods and an *a priori* probability distribution $P(b_i|Z_{k-1})$, the *a posteriori* probabilities $P(b_i|Z_k)$ of the behaviors can be calculated using Bayes inference rule given by Equation (9).

In order to calculate probabilities on the set $C = \{c_1, c_2, ..., c_{nc}\}$ of nc possible classes, a projection of the behaviors probabilities on the classes space is realized. This operation is referred to as probabilities conditioning.

5.1.2. Probabilities conditioning

The probabilities conditioning is necessary in the case where the behaviors set B and classes set C are not in one-to-one correspondence. As explained in [14], the conditioning step can be performed using the following equation:

$$P(C) = M \times P(B), \tag{21}$$

where $P(C)$ is an nc dimension vector containing the probabilities of C elements, $P(B)$ is an nb dimension vector containing the probabilities of B elements and M is an $nc \times nb$ matrix expressing the behaviors and classes relations (conditional probabilities $P(c_i|b_j)$).

For example, if the slowly walking mode behavior probability is equal to 1, by knowing that a slowly walking person can correspond to an old person or a middle-aged person, the conditional probabilities can be expressed by: $P(old\ person|$slowly walking mode$) = 1/2$ and $P(middle - aged\ person|$slowly walking mode$) = 1/2$. As it can be remarked, the conditioning step depends on the considered application, therefore more details are provided in the proposed simulation examples in Section 6.

5.2. Classification within the TBM

As in the Bayesian classification, the credal classification uses the IMM models probabilities μ_j and likelihoods Λ_j to calculate behaviors plausibilities computed as behaviors likelihoods in the Bayesian case by:

$$Pl(\{b_i\}) = \sum_{j:m_j \in M_{b_i}} \mu'_j \Lambda_j \quad i = 1, ..., nb\ , \tag{22}$$

where likelihoods Λ_j are calculated using Equation (18) and μ'_j are normalized as in Equation (20). Once the plausibility $Pl(\{b_i\})$ of each behavior ($b_i \in B$) is obtained, mass functions on B can be computed using the Generalized Bayesian Theorem [12, 26] with Equations (10) and (11).

The Generalized Bayesian Theorem provides a mass function on the behaviors set B. In order to obtain the targets classes, the behaviors mass function has to be projected on the classes space C.

5.2.1. Belief function conditioning

The conditioning step is crucial in the new classification chain when the relation between behaviors and classes are known. In the credal classification method, it is more precise than in the Bayesian case. It transfers mass functions defined on B to mass functions defined on C using the following equation:

$$m^C = \bar{M} \times m^B, \tag{23}$$

where $\bar{M}$ is a matrix expressing the relations between behaviors and classes, it contains masses $m(A|D)$, with $A \subseteq C$ and $D \subseteq B$. For example, if m^B(slowly walking mode) = 1, the corresponding conditioning for this simple example can be expressed as follows: $m(\{old\ person, middle - aged\ person\}|$slowly walking mode$) = 1$, because, even old and middle-aged persons can walk slowly. This conditioning example illustrates a big difference between belief function and Bayesian models. With belief functions, it is not required to estimate the respective *a priori* frequencies of the two classes *old* and *middle − aged* when walking (1/2 with the Bayesian approach).

5.2.2. Decision making

The resulting cumulated mass function m_k is supposed having all the available information. In order to make a decision about the tracked targets classes, the cumulated mass function is simply transformed into a pignistic probability using Equation (8).

Bayesian and credal classification schemes are respectively illustrated in Figure 7 and in Figure 8.

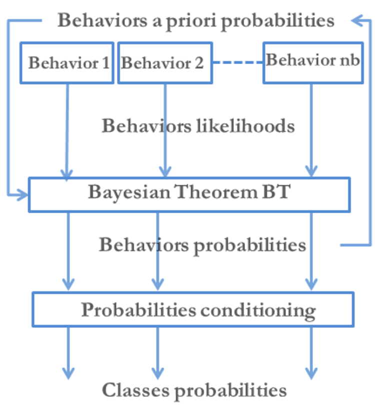

Figure 7: Bayesian classification.

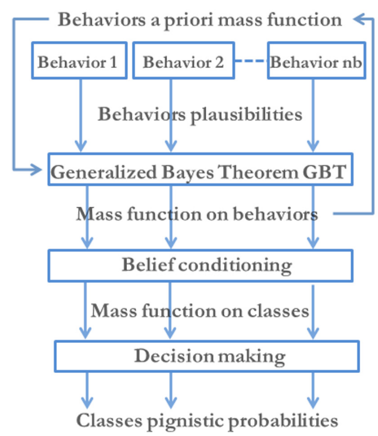

Figure 8: Credal classification.

6. Simulation examples

6.1. Multi-aircraft tracking and classification example

6.1.1. Description

In this simulation, targets dynamic models are chosen in the same manner as those used in the works of Ristic et *al.* [6], Ristic and Smets [14], and Caron et *al.* [39] which treat the case of a single target tracking and classification problem.

The state vector of all the targets is given by: $x = [x \quad \dot{x} \quad y \quad \dot{y}]$. It represents the position and the velocity in (x, y) directions. The state vector of each target evolves according to the following equation:

$$x_k = F x_{k-1} + B u_k + w_k \,, \tag{24}$$

where:

$$F = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} (\Delta T)^2/2 & 0 \\ \Delta T & 0 \\ 0 & (\Delta T)^2/2 \\ 0 & \Delta T \end{bmatrix},$$

with ΔT the sampling time. The state noise covariance matrix is taken equal to $0.005(B \times B')$, where B' is matrix B transpose.

Targets measurements are taken according to Equation (13), with a measurement noise variance equal to 0.2 and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Vector $u = \begin{bmatrix} a_x & a_y \end{bmatrix}^T$ in Equation (24) represents a given acceleration mode. The acceleration limitations for the *a priori* known targets classes are expressed as follows:

$$-L_i \leq \{a_x, a_y\} \leq L_i \ , \tag{25}$$

where $L_i = 0g, 2g$ and $4g$ are respectively the acceleration limits of the classes c_1, c_2 and c_3, with $g = 9.81\ m/s^2$ being the gravitational acceleration.

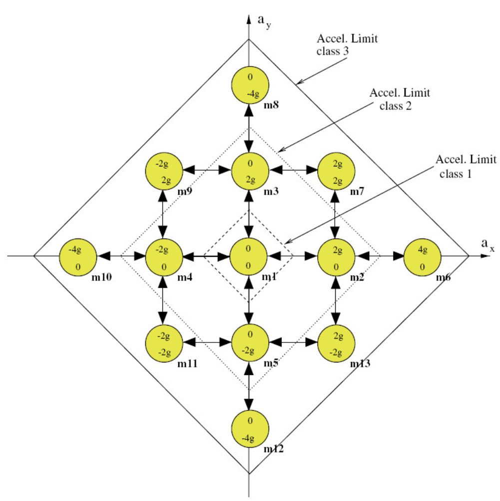

Figure 9: Acceleration modes over targets classes (standard example initially studied in [6] for a single target classification).

The different acceleration modes, in this example, are initially defined in [6]. Their distribution over possible classes are illustrated in Figure 9. Let us denote the set of models in Figure 9 by $M = \{m_1, ..., m_{13}\}$. Based on *a priori* knowledge, the behaviors models sets $M_{b_i} \subseteq M$ can be defined. In this example, three different behaviors are defined:

- Behavior 1 (b_1) corresponds to targets evolving with constant velocity only (e.g. liners). The models set $M_{b_1} = \{m_1\}$ contains the only model corresponding to a zero acceleration ($u = [0 \quad 0]^T$).

- Behavior 2 (b_2) concerns targets evolving with constant velocity and performing medium maneuvers (e.g. bombers). The models set of behavior b_2 is given by: $M_{b_2} = \{m_1, ..., m_5\}$, it contains the models whose acceleration is limited to $2g$.

- Behavior 3 (b_3) is associated to targets evolving with constant velocity and performing both medium and sharp maneuvers (e.g. fighters). Models set of behavior b_3 is given by: $M_{b_3} = \{m_1, ..., m_{13}\}$, it contains all the possible evolution models.

6.1.2. Classes conditioning

Tracked targets are classified according to the set of classes $C = \{c_1, c_2, c_3\}$ with:

- Class 1 (c_1): liners class.

- Class 2 (c_2): bombers class.

- Class 3 (c_3): fighters class.

Beliefs/probabilities on the set of classes C can be obtained by converting the behaviors beliefs/probabilities using the following relations:

- relation 1: a target with behavior 1 can correspond to a liner, a bomber or a fighter. All of them can evolve with a constant velocity. This relation can be written as: $b_1 = \{c_1, c_2, c_3\}$.

- Relation 2: a target with behavior 2 performs a medium maneuver, may correspond to a bomber or a fighter. Liners are supposed unable to perform any maneuver. This relation can be written as $b_2 = \{c_2, c_3\}$.

- Relation 3: a target in behavior 3 performs a sharp maneuver, it can only be a fighter, because liners and bombers can not perform sharp maneuvers. This relation can be written as $b_3 = \{c_3\}$.

For the Bayesian classification task, the conditioning is performed using the Equation (21), according to the relations above. The corresponding conditioning matrix M is given by:

$$M = \begin{bmatrix} 1/3 & 0 & 0 \\ 1/3 & 1/2 & 0 \\ 1/3 & 1/2 & 1 \end{bmatrix}.$$

which corresponds to the following conditions:

- if $P(b_1) = 1 \Longrightarrow P(c_1|b_1) = \frac{1}{3},\ P(c_2|b_1) = \frac{1}{3},\ P(c_3|b_1) = \frac{1}{3}$.

- If $P(b_2) = 1 \Longrightarrow P(c_1|b_2) = 0,\ P(c_2|b_2) = \frac{1}{2},\ P(c_3|b_2) = \frac{1}{2}$.

- If $P(b_3) = 1 \Longrightarrow P(c_1|b_3) = 0,\ P(c_2|b_3) = 0,\ P(c_3|b_3) = 1$.

For the credal classification, a different conditioning is made following Equation (23). It transfers beliefs expressed on the set of behaviors B to the set of classes C such that:

- $m_k(\{c_1, c_2, c_3\}|b_1) = 1$ (cf Relation 1)
- $m_k(\{c_2, c_3\}|b_2) = 1$ (cf Relation 2)
- $m_k(\{c_1, c_2, c_3\}|\{b_1, b_2\}) = 1$ (cf Relations 1 and 2)
- $m_k(c_3|b_3) = 1$ (cf Relation 3)
- $m_k(\{c_1, c_2, c_3\}|\{b_1, b_3\}) = 1$ (cf Relations 1 and 3)
- $m_k(\{c_2, c_3\}|\{b_2, b_3\}) = 1$ (cf Relations 2 and 3)
- $m_k(\{c_1, c_2, c_3\}|\{b_1, b_2, b_3\}) = 1$ (cf Relations 1, 2 and 3)

The corresponding complete conditioning matrix $\bar{M}$ has a size $(2^3 = 8) \times (2^3 = 8)$, and is given by:

$$\bar{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Belief conditioning, in this example is similar to those performed in [14] for single target classification. This matrix enables to compute the mass functions on the classes space C in order to perform the credal classifications.

6.1.3. Simulation results

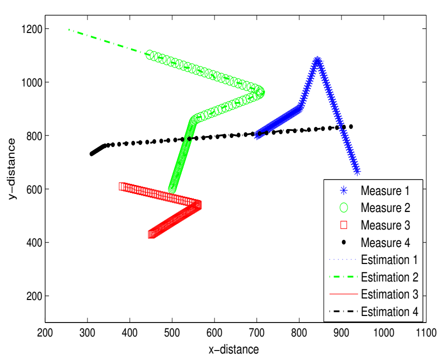

Figure 10: Targets measurements and estimations on x and y positions.

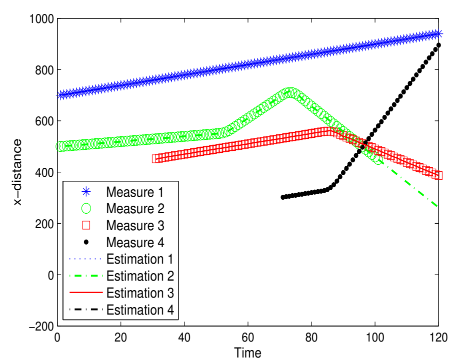

Figure 11: Targets measurements and estimations on x positions over time.

In this scenario, 4 targets are involved and perform some maneuvers in (x, y) space. Measurements and estimations are presented in Figures 10 and 11.

For example, it can be noticed that targets 3 and 4 respectively appear at time steps 30 and 70.

Target 2 evolves firstly with a constant velocity in order to minimize fuel use and then performs two maneuvers: a first medium acceleration in x direction during time period $50-54$, and a sharp deceleration during time period $70-75$. Finally it disappears at time step 100. The algorithm continues to predict its trajectory until the end of the simulation at time step 120 because its score function has not yet reached the deletion threshold.

According to this scenario, a full ignorance is expected on the true class of target 2 before its first medium maneuver (as it has only flown with a constant velocity), then a complete doubt between the bomber and fighter classes is expected after the first medium maneuver and before the sharp maneuver. Finally after this second maneuver, only the fighter class remains possible.

The evolution of target 2 behaviors likelihoods and probabilities are, respectively, presented in Figures 12 and 13. These results correspond to averages of 20 Monte Carlo simulations.

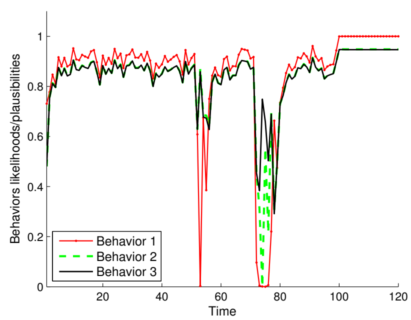

Figure 12: Behaviors likelihoods/plausibilities of target 2.

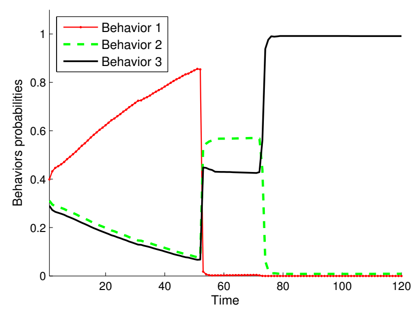

Figure 13: Behaviors probabilities of target 2.

Behaviors likelihoods/plausibilities are calculated using Equation (19) (or (22)). It can be remarked that the likelihood/plausibility of behavior b_1 drops to zero during the two performed maneuvers and the likelihood/plausibility of behavior b_2 drops to zero during the sharp maneuver.

More significantly, it can be observed that during the period of time preceding the first maneuver the likelihood/plausibility of behavior b_1 is slightly more important than the likelihoods/plausibilities of behaviors b_2 and b_3. As a consequence, it can be seen in Figure 13 that during this same period of time before the first maneuver, the probability corresponding to behavior b_1 tends to grow while all behaviors are supposed to be equally probable.

This deviation is due to the fact that as the behaviors are nested ($b_1 \subset b_2 \subset b_3$) and behavior b_1 is composed of the least number of models (only m_1), its likelihood during the period of constant velocity is less influenced by non-concerned models unlike behaviors b_2 and b_3. Formally it can be observed from Equation (19). The same explanation can be adopted after the second maneuver where b_2 is also unjustifiably advantaged over b_3 because the number of models in behavior b_2 is less than the number of models in behavior b_3.

Figure 13 is based on Bayes rule and does not concern the credal classification. The conditioning operations described above are used to correct such an issue.

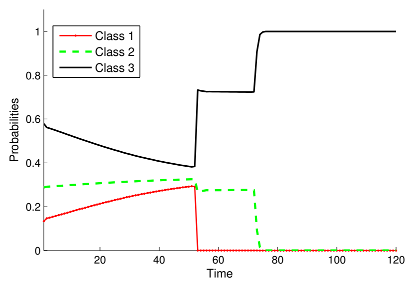

Figure 14: Target 2 Bayesian classification (after conditioning).

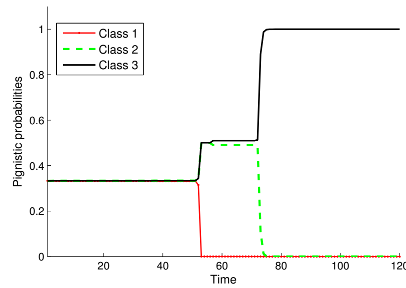

Figure 15: Target 2 credal classification.

Using Bayesian conditioning on the behaviors probabilities in Figure 13, classes probabilities are obtained as shown in Figure 14. It can be seen that, even if the target is finally classified as a fighter (which is the truth), the Bayesian classifier fails in doubt situations. On the other hand, the pignistic probabilities derived from the conditioned classes mass function, given in Figure 15, shows that the credal classifier succeeds to manage the imprecisions on behaviors and classification results are as expected.

It can also be mentioned that after time step 100, when target 2 is not detected, the behaviors likelihoods are still constant because the target is updated with its predicted measurement. In the non-detection time, the classifiers (Bayesian and credal) still believe that target 2 is a fighter.

Results presented in this section shows therefore that the credal model improves the Bayesian classification results obtained in [6]. They also illustrate the strength of the credal classification in a multi-target context by assuming that each target is at each time updated with its corresponding measurement.

Influences on the classification of false assignments are presented in Section 6.2.

6.2. *Tracking and classification of nearby targets*

The problem of crossing and nearby targets is addressed in this section which aims at studying the robustness of the tracking algorithm, especially the assignment step ensured here by the generalized GNN algorithm.

6.2.1. *Case of two targets evolving very closely*

The extreme situation of two targets evolving very closely is considered. The goal is to measure the rate of false assignments by varying sensor noise variance. In this simulation, sensor noise varies from 0% to 200% of the distance between the two targets. For each value of noise variance, an average value, of the false assignments rate on 100 Monte Carlo simulations, is calculated. Results are presented in Figure 16.

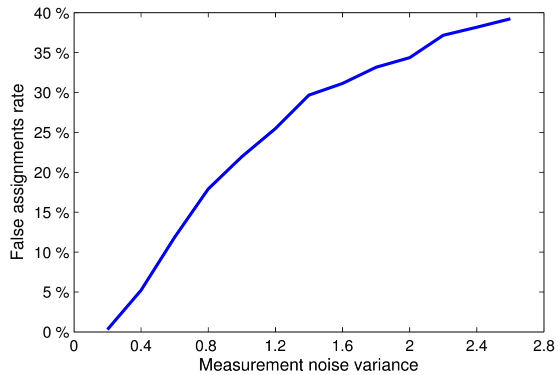

Figure 16: Rate of false assignments of GNN algorithm for two closely moving targets, with different sensor noise values.

It can be observed that the rate of false assignments is relatively low even for an extreme scenario of targets evolving closely and sensor noise exceeding the distance between the targets.

The false assignment rate would be more important if the measurements were assigned to the targets estimates of time step $k-1$. The capacity of the IMMs to predict the expected measurements of the targets at time step k allows the reduction of the false assignment rate.

6.2.2. Case of a fighter crossing an unknown target and evolving closely

In this section, a simple scenario is considered to illustrate what can be the impact of false assignments on the proposed multi-target classification. A fighter and an unknown aircraft evolve in (x, y) space according to the previously described models. Figure 17 illustrates the time evolution of the targets according to the y direction (the scenario is the same in x direction). Targets pass each other and then evolve closely. The same figure shows the false assignment happened at time step 38. The influence of this false assignment on the two targets classifications is presented in Figure 18.

Note that in this example, during all the surveillance period, target 2 evolves only with a constant velocity while target 1 starts its movement with a constant velocity and makes two strong maneuvers, namely, acceleration and deceleration around time steps 23 and 35. Given this information, target 1 is expected to be classified as a fighter (class c_3) after its strong maneuvers, and a perfect doubt concerning the classification of target 2 is expected.

Figure 18 gives the credal classification results of targets 1 and 2 respectively. As expected, it can be seen that target 1 is correctly classified as a fighter after its first strong maneuver and its classification is not influenced by the false assignment occurred at time step 38 (a target classified in the third class can not be brought into doubt). On the other hand, it can be seen that the false assignment at time step 38

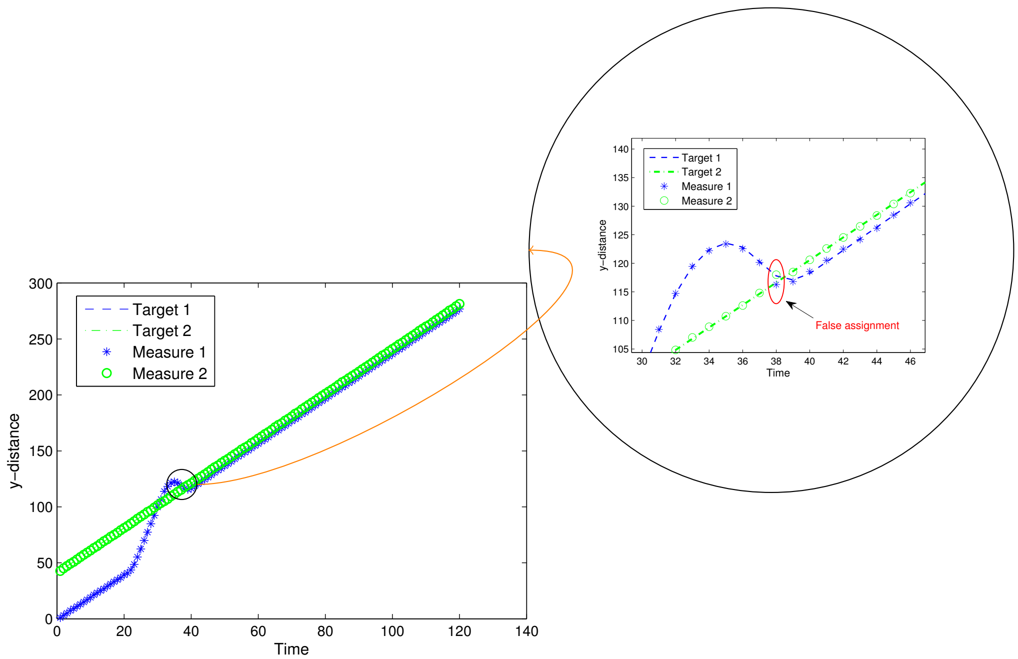

Figure 17: False assignment of the measurements of a fighter and an unknown aircraft.

has deteriorated the classification of the second target, each class having to be equally probable. The false assignment misleads the classifier to a wrong classification by advantaging the third and second classes over the first one.

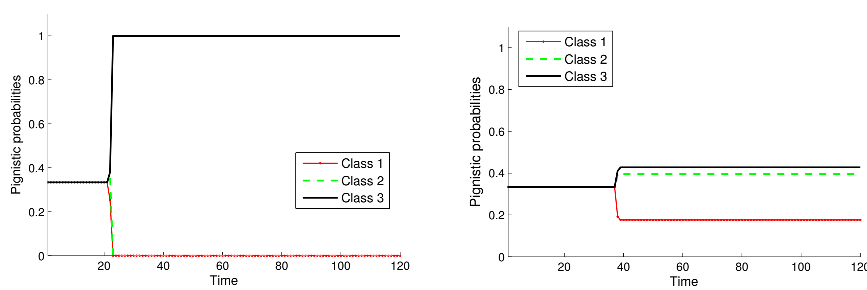

Figure 18: Credal classification results of two crossing targets: target 1 (on the left) and target 2 (on the right).

This article is focused on the classification stage, so only a simple assignment algorithm is used, i.e. the GNN algorithm. It could be replaced by any advanced assignment algorithm like JPDA and MHT to overcome such problem of conflicting situations. Some credal solutions [15, 17, 41, 54, 55, 56] can also be integrated to overcome the problem tackled in this section. This would enhance the results of the proposed classification strategy.

6.3. *Pedestrians tracking and classification example*

6.3.1. *Description*

This section proposes a classification problem concerning pedestrian activity recognition. The structure of the dynamical models is close to the one used in the previous aircraft tracking example. Indeed, this example is designed to be interesting, not for the tracking issue, but for the classification stage.

However, compared to the previous examples based on aircraft targets, two important differences are introduced. First, two classes of pedestrians are considered. They contain a common pedestrian behavior and a distinct one (there is no relation of strict inclusion as in the aircraft example). Secondly, pedestrians classes are not constant over time which conducts to a different classification problem.

With slight adaptations, the credal classification algorithm is shown to provide correct results, demonstrating that the methodology presented in this article can be applied to a wide variety of applications.

As in the former example, the state vector consists in the position and velocity on x and y directions. The dynamical model is given by:

$$x_k = A(s_x, s_y)x_{k-1} + w_k \ , \tag{26}$$

with:

$$A(s_x, s_y) = \begin{bmatrix} 1 & s_x\Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_y\Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where s_x and s_y represent the targets speeds (*meter*/*second*) according to x and y directions, respectively. According to different values of s_x and s_y, a set $M = \{m_1, m_2, ..., m_7\}$ of 7 different models are designed and used by each target IMM to track its movement. Pedestrians behaviors are defined according to the set of models M as follows:

- behavior 1 (b_1) corresponds to the static mode. Behavior 1 models set is given by: $M_{b_1} = \{m_1\}$, where m_1 corresponds to the matrix $A(s_x, s_y)$ with, $(s_x, s_y) = (0, 0)$.
- Behavior 2 (b_2) corresponds to a walking mode. Behavior 2 models set is given by: $M_{b_2} = \{m_2, m_3, m_4\}$, it corresponds to the speed models $(s_x, s_y) = \{(2, 0), (0, 2), (2, 2)\}$.
- Behavior 3 (b_3) corresponds to a running mode. Behavior 3 models set is given by: $M_{b_3} = \{m_5, m_6, m_7\}$, it corresponds to the speed models $(s_x, s_y) = \{(6, 2), (2, 6), (6, 6)\}$.

6.3.2. *Classes conditioning*

According to the behaviors described above, the problem consists in distinguishing among the pedestrians, the ramblers and the sportsmen. This means to decide on the set $C = \{c_1, c_2\}$ which corresponds to C = {ramblers, sportsmen}. The conditioning step in this example is done according to the following relations:

- Relation 1: ramblers with static and walking behaviors ($c_1 = \{b_1, b_2\}$).
- Relation 2: sportsmen with walking and running behaviors ($c_2 = \{b_2, b_3\}$).

These relations show that the two classes are overlapping. Both ramblers and sportsmen can adopt a walking mode. It is supposed that sportsmen do not stay static and ramblers do not run.

In order to obtain a mass function on the classes space, a mass function on behaviors space B is conditioned according to Equation (21), with a matrix $\bar{M}$ expressing the following conditions from Relations 1 and 2:

- $m_k(c_1|b_1) = 1$
- $m_k(\{c_1, c_2\}|b_2) = 1$
- $m_k(\{c_1, c_2\}|\{b_1, b_2\}) = 1$
- $m_k(c_2|b_3) = 1$
- $m_k(\{c_1, c_2\}|\{b_1, b_3\}) = 1$
- $m_k(\{c_1, c_2\}|\{b_2, b_3\}) = 1$
- $m_k(\{c_1, c_2\}|\{b_1, b_2, b_3\}) = 1$

The complete conditioning matrix $\bar{M}$ transforms the belief mass on power set 2^3 of three behaviors on a power set 2^2 of two classes, it is given as follows:

$$\bar{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

This conditioning provides a mass function on classes set C which is used to classify pedestrians as explained in credal classification section.

6.3.3. Simulation results

An example including two pedestrians is provided. Pedestrians evolve in (x, y) space. Pedestrians time evolutions according to y direction are exposed in Figure 19. Evolutions in x direction are identical. Pedestrian 1 first walks, then runs and then stops. It is then expected that after a doubt, Pedestrian 1 should be classified as a sportsman and after that as a rambler.

As this is shown in Figure 20, the standard classification algorithm described in Section 5.2 fails. In fact, it cannot even compute the pignistic probabilities after time step 120. This is due to the fact that the class of pedestrian 1 has converged to a sportsman, and suddenly at time step 120 the likelihoods are associated to another class, and there is an absolute conflict between the two belief masses in Equation (11). The credal classification algorithm exposed in Section 5.2 is not adapted for time varying classes.

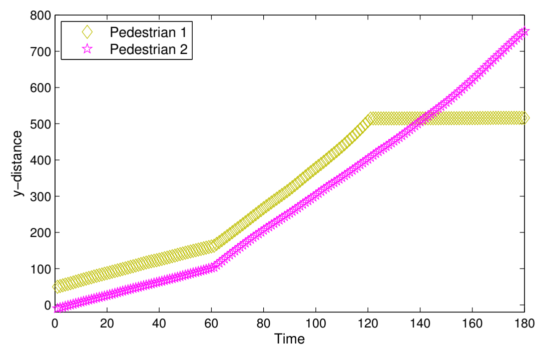

Figure 19: Pedestrians trajectories.

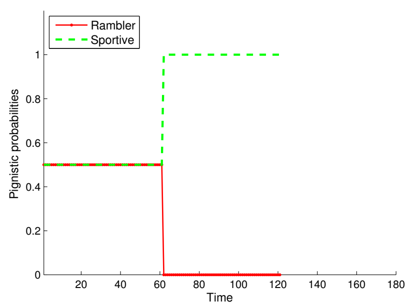

Figure 20: Activity of pedestrian 1 with the standard algorithm (without discounting).

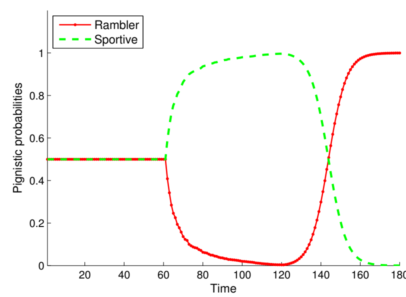

Figure 21: Activity of pedestrian 1 with the twisted algorithm and discounting parameter $\lambda = 0.3$.

A simple solution to enable classification of such targets consists in adding a discounting operation of the instantaneous belief $m_k(.|z)$ in Equation (11), which is obtained from the likelihoods and the Generalized Bayesian Theorem, before applying the conjunctive fusion rule. This way the conflict is lowered between old and new beliefs, and there is an adaptation of the inferred classes. This discounting phase, performed as described in Equation (7) with a value of $\lambda = 0.3$ (chosen small enough to provide smooth and slow transitions) allows the classifier to follow the pedestrian class even if it changes in time, as shown in Figure 21.

Let us also note that discounting the *a priori* belief m_{k-1}, in Equation (11), instead of $m_k(.|z)$ deduced from the GBT also enables to obtain a correct result, but only for a given time. Indeed, the likelihoods are favored in that scheme. They are associated to the doubt if the sportsman starts to walk again, depending on λ. Thus, sooner or later, the classification of pedestrian 1 converges to a doubt again. In Figures 22 and 23, the previous simulation is continued so that a walking mode appears near time step 240. These figures show the differences in the classifications outputs based on the two distinct discountings.

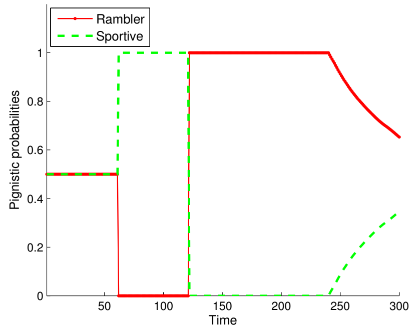

Figure 22: Cumulative belief (m_{k-1}) discounting, with $\lambda = 0.9$.

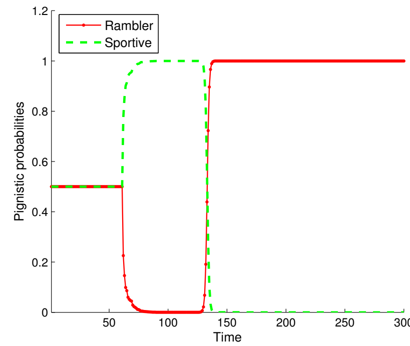

Figure 23: Instantaneous belief (m_k) discounting, with $\lambda = 0.9$.

More generally, a discounting of both belief functions could be introduced simultaneously, with optimization of the discounting factors λ_i, according to the dynamics of the targets classes and other factors. The fusion rule could also be changed. But these parameters would be very application dependent, and they cannot be studied further in this article.

To conclude this section, the pedestrian example shows that the standard algorithm proposed here can serve as a basis for more complex problems, since it can be adapted easily to handle various situations while providing satisfactory results.

7. Conclusion

Multi-target classification is a fundamental problem, when it comes to classify multiple targets simultaneously. It is much more complex than the single target classification problem, and even more when targets randomly appear and disappear for various reasons.

This article presents a solution for multi-target classification with belief functions exploiting the results provided by Interacting Multiple Model (IMM) algorithms in charge of the tracking of the targets. Including the generalized Global Nearest Neighbor (GNN) algorithm to solve assignment problems and a score function to handle targets appearances and disappearances, this full complete scheme for multi-target tracking and classification has been tested on two different kind of examples: one with constant class targets and one with time varying class targets. It has been shown that the full algorithm using belief functions outperforms standard Bayesian classifiers in both situations.

In future work, tests of the credal classification may be undertaken by replacing any elementary component of the decision chain by a more advanced one. For example, the tracking task can be handled by PHD or particular filters and in the same way the assignment task can be ensured by other solutions like JPDA, MHT and so on.

More deep investigations on time varying classes may be realized (for example, the automatic computation of the discounting rate). Likewise, the assignment problem may be developed with belief functions.

Acknowledgement

The authors are very grateful to the anonymous reviewers for their valuable comments, which have helped them to improve the clarity and the quality of this article.

References

[1] Y. Bar-Shalom, K-C. Chang, and H.A.P. Blom. Automatic track formation in clutter with a recursive algorithm. In *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 1402–1408, 1989.

[2] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, and J. Wiley. *Estimation with applications to tracking and navigation*. Wiley Library, 2001.

[3] S.S. Blackman. *Multiple-target tracking with radar applications*. Artech House Radar Library, 1986.

[4] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, San Diego, USA., 1987.

[5] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[6] B. Ristic, N. Gordon, and A. Bessell. On target classification using kinematic data. *Information Fusion*, 5(1):15–21, 2004.

[7] W. Mei, G. Shan, and Y. Wang. A second-order uncertainty model for target classification using kinematic data. *Information Fusion*, 12(2):105–110, 2011.

[8] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial intelligence*, 66(2):191–234, 1994.

[9] A.P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society. Series B*, pages 205–247, 1968.

[10] G. Shafer. *A mathematical theory of evidence*. Princeton university press, Princeton, NJ, USA, 1976.

[11] K. Sentz and S. Ferson. Combination of evidence in Dempster-Shafer theory. Technical Report SAND 2002-0835, Sandia National Laboratories, Albuquerque, New Mexico, 2002.

[12] F. Delmotte and P. Smets. Target identification based on the Transferable Belief Model interpretation of Dempster-Shafer model. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(4):457–471, 2004.

[13] B. Ristic and P Smets. The TBM global distance measure for the association of uncertain combat ID declarations. *Information Fusion*, 7(1):276–284, 2006.

[14] P. Smets and B. Ristic. Kalman filter and joint tracking and classification based on belief functions in the TBM framework. *Information Fusion*, 8(1):16–27, 2007.

[15] D. Mercier, É. Lefèvre, and D. Jolly. Object association with belief functions, an application with vehicles. *Information Sciences*, 181(24):5485–5500, 2011.

[16] N. Wartelle, F. Delmotte, and D. Gaquer. Classification de scènes à nombre d'objets inconnu. In *MAnifestation des Jeunes Chercheurs STIC*, pages 22–24, 2006.

[17] N. El Zoghby, V. Cherfaoui, and T. Denoeux. Optimal object association from pairwise evidential mass functions. In *Proceedings of the 16th International Conference on Information Fusion*, 2013.

[18] S. Hachour, F. Delmotte, E. Lefèvre, and D. Mercier. Tracking and identification of multiple targets. In *Proceedings of the 7th Workshop Interdisciplinaire sur la Sécurité Globale*, Troyes, France, 2013, available online at: http://www.agence-nationale-recherche.fr/Colloques/WISG2013/articles/Article_Delmotte.pdf.

[19] P. Smets. Analyzing the combination of conflicting belief functions. *Information Fusion*, 8(4):387–412, 2007.

[20] D. Mercier, É. Lefèvre, and F. Delmotte. Belief functions contextual discounting and canonical decompositions. *International Journal of Approximate Reasoning*, 53(2):146–158, 2012.

[21] F. Delmotte. Detection of defective sources in the setting of possibility theory. *Fuzzy Sets and Systems*, 158(5):555–571, 2007.

[22] P. Smets. What is Dempster-Shafer's model. *Advances in the Dempster-Shafer theory of evidence*, pages 5–34, 1994.

[23] A. Benavoli and B. Ristic. Classification with imprecise likelihoods: A comparison of TBM, random set and imprecise probability approach. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, 2011.

[24] P. Smets. Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38(2):133–147, 2005.

[25] B.R. Cobb and P.P. Prakash. On the plausibility transformation method for translating belief function models to probability models. *International Journal of Approximate Reasoning*, 41(3):314–330, 2006.

[26] P. Smets. Belief functions: the disjunctive rule of combination and the Generalized Bayesian Theorem. *International Journal of Approximate Reasoning*, 9(1):1–35, 1993.

[27] T-D. Le Duy, D. Vasseur, M. Couplet, L. Dieulle, and C. Bérenguer. A study on updating belief functions for parameter uncertainty representation in Nuclear Probabilistic Risk Assessment. In *7th International Symposium on Imprecise Probability: Theories and Applications, Innsbruck*, 2011.

[28] A. Aregui and T. Denœux. Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. *International Journal of Approximate Reasoning*, 49(3):575–594, 2008.

[29] A-S. Capelle, O. Colot, and C. Fernandez-Maloigne. Evidential segmentation scheme of multi-echo MR images for the detection of brain tumors using neighborhood information. *Information Fusion*, 5(3):203–216, 2004.

[30] S.S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech House, Norwood, MA, USA, 1999.

[31] G.A Watson and W.D Blair. IMM algorithm for tracking targets that maneuver through coordinated turns. In *Proceedings of Signal and Data Processing of Small Targets*, volume 1698, pages 236–247, 1992.

[32] R.A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems.*, (4):473–483, 1970.

[33] Y. Bar-Shalom and K. Birmiwal. Variable dimension filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, (5):621–629, 1982.

[34] W. Li, Y. Jia, J. Du, and J. Zhang. PHD filter for multi-target tracking with glint noise. *Signal Processing*, 2013, accepted for publication.

[35] Y. Li, H. Xiao, Z. Song, R. Hu, and H. Fan. A new multiple extended target tracking algorithm using PHD filter. *Signal Processing*, 2013, accepted for publication.

[36] B-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. In *Proceeding of the 6th International Conference on Information Fusion*, pages 792–799, 2003.

[37] M-C. Ho, C-C. Chiang, and Y-Y. Su. Object tracking by exploiting adaptive region-wise linear subspace representations and adaptive templates in an iterative particle filter. *Pattern Recognition Letters*, 33(5):500–512, 2012.

[38] A. Mukherjee and A. Sengupta. Likelihood function modeling of particle filter in presence of non-stationary non-Gaussian measurement noise. *Signal Processing*, 90(6):1873–1885, 2010.

[39] F. Caron, B. Ristic, E. Duflos, and P. Vanheeghe. Least committed basic belief density induced by a multivariate Gaussian: Formulation with applications. *International Journal of Approximate Reasoning*, 48(2):419–436, 2008.

[40] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2034–2041, 2012.

[41] N. Megherbi, S. Ambellouis, O. Colot, and F. Cabestaing. Multimodal data association based on the use of belief functions for multiple target tracking. In *Proceedings of the 8th International Conference on Information Fusion*, volume 2, pages 7–14, 2005.

[42] C. Royère, D. Gruyer, and V. Cherfaoui. Data association with belief theory. In *Proceedings of the 3rd International Conference on Information Fusion*, volume 1, pages TUD2–3, 2000.

[43] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *Proceedings of the 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes.*, volume 19, pages 807–812, 1980.

[44] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, 1994.

[45] H.A.P. Blom and E.A. Bloem. Combining IMM and JPDA for tracking multiple maneuvering targets in clutter. In *Proceedings of the 5th International Conference on Information Fusion.*, volume 1, pages 705–712, 2002.

[46] P. Konstantinova, A. Udvarev, and T. Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech-03)*, 2003.

[47] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.

[48] F. Septier, S-K. Pang, A. Carmi, and S. Godsill. On MCMC-based particle methods for Bayesian filtering: Application to multitarget tracking. In *Proceedings of the 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 360–363, 2009.

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

[50] M. Baum and U-D. Hanebeck. Association-free tracking of two closely spaced targets. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 62–67, 2010.

[51] F. Bourgeois and J-C. Lassalle. An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971.

[52] GJ. McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

[53] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[54] A. Ayoun and P. Smets. Data association in multi-target detection using the Transferable Belief Model. *International Journal of Intelligent Systems*, 16(10):1167–1182, 2001.

[55] Y. Lemeret, É. Lefèvre, and D. Jolly. Improvement of an association algorithm for obstacle tracking. *Information Fusion*, 9(2):234–245, 2008.

[56] J. Daniel and J-P. Lauffenburger. Multi-object association decision algorithms with belief functions. In *Proceedings of the 15th International Conference on Information Fusion*, pages 669–676, 2012.