Robust multi-target sensing/tracking in the Bayesian Occupancy Filter framework

Kamel Mekhnacha, David Raulo

Robust multi-target sensing/tracking in the Bayesian Occupancy Filter framework

Kamel Mekhnacha* — **David Raulo***

* *Probayes, 345, rue Lavoisier, F-38030 St Ismier Cedex*
{Kamel.Mekhnacha, David.Raulo}@probayes.com

ABSTRACT. *We present the "Bayesian Occupancy Filter" (BOF) and the "Fast Clustering-Tracking" algorithms as a framework for robust sensing and multi-target tracking using multiple sensors. Perceiving of the surrounding physical environment reliably is a major demanding in smart systems requiring a high level of safety such as car driving assistant, autonomous robots, and surveillance. The dynamic environment need to be perceived and modeled according to the sensor measurements which could be noisy. To fit such a requirement, we propose a hierarchical approach in which two filtering layers are used:*

(i) Robust grid-level sensor fusion using the "Bayesian Occupancy Filter" algorithm in order to construct an occupancy/velocity grid representation of the environment.

(ii) Robust object-level tracking using the "Fast Clustering-Tracking".

RÉSUMÉ. *Nous présentons les algorithmes "Bayesian Occupancy Filter" (BOF) et "Fast Clustering-Tracking" comme un cadre pour la perception robuste et le suivi de cibles multiples en utilisant plusieurs capteurs. Percevoir l'environnement physique d'une manière fiable est une exigence clef dans les systèmes intelligents nécessitant un grand degré de sécurité tels que l'assistante à la conduite automobile, la robotique autonome et la surveillance. L'environnement dynamique doit être perçu et modélisé en utilisant les mesures capteurs généralement bruitées. Pour répondre à cette exigence, nous proposons une approche hiérarchique dans laquelle deux niveaux de filtrage sont utilisés:*

(i) Fusion robuste de capteurs en utilisant l'algorithme "Bayesian Occupancy Filter" dans lequel l'environnement dynamique est représenté par une grille d'occupation/vitesse.

(ii) Suivi robuste d'objets en utilisant l'algorithme "Fast Clustering-Tracking".

KEYWORDS: *Bayesian inference, multi-sensor data fusion, multi-target tracking, occupancy grid.*

MOTS-CLÉS : *Inférence Bayésienne, fusion multi-capteurs, suivi de cibles multiples, grille d'occupation.*

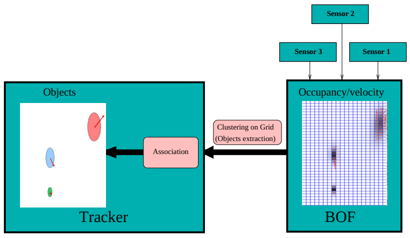

Figure 1. *Robust Sensing/Tracking system architecture.*

1. Introduction

It has been shown that the dynamic environment can be efficiently and robustly represented by the Bayesian Occupancy Filter (BOF) (Tay *et al.*, 2008). In the BOF framework, this environment is decomposed into a grid-based representation (Moravec, 1988) in which both the occupancy and the velocity distributions are estimated for each grid cell. In such a representation, concepts such as objects or tracks do not exist and the estimation is achieved at the cell level. However, the object-level representation is mandatory for applications needing high-level representations of obstacles and their motion.

To achieve this, a natural approach is to perform clustering on the BOF output grid in order to extract objects. We present in this paper a novel clustering-tracking algorithm. Its main ideas are:

1) using the prediction result of the tracking module as a form of feedback to the clustering module in order to avoid searching in the complete grid,

2) using both occupancy and velocity estimates in order to better separate the extracted clusters.

This algorithm reduces drastically the complexity of the data association. Compared with the traditional data association approaches such as the joint probabilistic data association (JPDA) algorithm (Bar-Shalom *et al.*, 1988), our algorithm demands less computational cost, so as to be suitable for environments with large amount of dynamic objects and/or a large amount of sensor observations.

The proposed sensing/tracking framework is based on a hierarchical approach in which two filtering layers are used (Fig. 1). In this architecture, the output grid of the BOF filter (i.e, the probability distributions over the occupancy and over the velocity of the cell) is used as input for the object tracker by extracting object hypothesis from the grid.

The paper is organised as follows. In the next section, the "Bayesian Occupancy Filter" (BOF) algorithm is briefly described. The "Fast Clustering-Tracking" algorithm is presented in section III. Finally, conclusions are drawn in section IV.

2. Bayesian Occupancy Filter (BOF)

The Bayesian Occupancy Filter (BOF) is represented as a two-dimensional grid-based decomposition of the environment. Each cell of the grid contains two probability distributions. The probability distribution over the occupancy of the cell, and the probability distribution over its velocity. Given a set of input sensor readings, the BOF algorithm allows to update the occupancy/velocity estimates of each grid cell.

BOF is a special implementation of the Bayesian filter approach (Jazwinski, 1970) (Thrun *et al.*, 2005). This approach addresses the general problem of recursively estimating the posterior probability distribution $P(X^k \mid Z^k)$ of the state X of a system conditioned on its observation Z. This posterior distribution is obtained in two stages: prediction and estimation. The prediction stage computes an a priori prediction of the target's current state known as the prior distribution. The estimation stage then computes the posterior distribution by using the prediction with the current measurement of the sensor.

In the case of the BOF, using this prediction/estimation scheme allows filtering out false alarms, miss-detections, and localization errors in sensors data readings. Figure 2 shows an example of BOF output using a computer vision car detector. The input of the BOF in this case is, for each time step, a set of bounding boxes corresponding to the detected vehicles (red boxes). The output represents a grid of occupancy probability (blue-to-red mapped color) and mean velocity (red arrows) estimates.

The Bayesian model presented in the following text is a reformulation of the one we presented in (Tay *et al.*, 2008). The aim of this reformulation is to make clearer the strong link between the discretization of the space and the discretization of the velocity, which reduces the number of the used random variables and makes the model easier to explain. The key idea of the model is to represent the 2D space using a regular grid. Given this space discretization and assuming that objects do not overlap, the velocity of a given c cell at a time t is directly linked to the identity of its antecedent cell A_c from which the content of cell c moved between $t-1$ and t. In other words, we can define the velocity of a given cell by providing the index of its antecedent. Therefore, estimating the velocity of a given cell is equivalent to estimating a probability table over all its possible antecedents. Possible antecedents of a cell are defined by providing a neighbourhood from which the cell is reachable in a time step. This model applies also to velocities needing more than one time step for a neighbour cell to reach c. However, for simplicity we will assume only one-step velocities (neighbours reaching c in one time step).

2.1. *The BOF model*

2.1.1. *Variables*

For a given cell having $c \in \mathcal{Y}$ as index in the grid, let:

– $A_c^t \in \mathcal{A}_c \subset \mathcal{Y}$ represents each possible antecedent of cell c over all the cells in the grid domain $\mathcal{Y}$. The set of antecedent cells of cell c is denoted by $\mathcal{A}_c$ and is defined as a neighbourhood of the cell c.

– $A_c^{t-1} \in \mathcal{A}_c \subset \mathcal{Y}$ the same as A_c^t but for the previous time step.

– $O_c^t \in \mathcal{O} \equiv \{0, 1\}$ is a boolean variable representing the state of the cell in terms of occupancy at time t, either $[O_c = 1]$ if occupied, $[O_c = 0]$ if empty. Given the independency hypothesis, the occupancy of each cell at time t is considered apart from the occupancy of its neighbouring cells at time t.

– $Z_i^t \in \mathcal{Z}, 1 \leq i \leq S \in \mathbb{N}$, is a generic notation for measurements yielded by each sensor i, considering a total of S sensors yielding a measurement at the considered time instant.

2.1.2. *Joint distribution factors*

The following expression gives the decomposition of the joint distribution of the relevant variables according to Bayes' rule and dependency assumptions:

$$P(A_c^{t-1}\, A_c^t\, O_c^t\, Z_1^t \cdots Z_S^t) = \\ P(A_c^{t-1}) P(A_c^t \mid A_c^{t-1}) P(O_c^t \mid A_c^{t-1}) \prod_{i=1}^{S} P(Z_i^t \mid A_c^t\, O_c^t). \quad [1]$$

The parametric form and semantics of each component of the joint decomposition are as follows:

– $P(A_c^{t-1})$ is the probability for a given neighbouring cell A_c to be the antecedent of c at time $t-1$. In order to represent the fact that cell c is *a priori* equally reachable from all possible antecedent cells in the considered neighbourhood, this probability table is initialised as uniform and is updated in each time step.

– $P(A_c^t \mid A_c^{t-1})$ is the distribution over antecedents at time t given the antecedent of cell c at $t-1$. It represents the prediction (dynamic) model over velocity. If we assume a perfect *constant velocity hypothesis* between the two time frames $t-1$ and t, this distribution is simply:

$$P(A_c^t \mid A_c^{t-1}) = P(A_{A_c^{t-1}}^{t-1}).$$

In other words, the predicted probability is simply the probability at the preceding time instant for the antecedent at $t-1$.

Considering imperfect *constant velocity hypothesis* may be done by introducing:

- $E \in \{0, 1\} \equiv$ "There was a prediction error",

- $P(E) = \epsilon$ the probability of violating the *constant velocity hypothesis* (a parameter of the model).

If we define:

- $P(A_c^t \mid A_c^{t-1} \neg E) = P(A^{t-1}_{A_c^{t-1}})$,

- $P(A_c^t \mid A_c^{t-1} E) = \mathcal{U}(A_c^t)$ (Uniform predicted antecedent (velocity) when *constant velocity hypothesis* is violated),

then, $P(A_c^t \mid A_c^{t-1})$ may be written as a mixture:

$$\begin{aligned} P(A_c^t \mid A_c^{t-1}) &= P(\neg E)P(A_c^t \mid A_c^{t-1} \neg E) + P(E)P(A_c^t \mid A_c^{t-1} E) \\ &= (1-\epsilon)P(A^{t-1}_{A_c^{t-1}}) + \epsilon\, \mathcal{U}(A_c^t) \\ &= (1-\epsilon)P(A^{t-1}_{A_c^{t-1}}) + \epsilon/\|\mathcal{A}_c\|. \end{aligned}$$

– $P(O_c^t \mid A_c^{t-1})$ is the distribution over occupancy given the antecedent of cell c at $t-1$. It represents the prediction (dynamic) model over occupancy. If we assume a perfect *constant velocity hypothesis* between the two time frames $t-1$ and t, this distribution is simply:

$$P(O_c^t \mid A_c^{t-1}) = P(O^{t-1}_{A_c^{t-1}}).$$

In other words, the predicted probability is simply the probability at the preceding time instant for the antecedent at $t-1$. When considering imperfect *constant velocity hypothesis*, $P(O_c^t \mid A_c^{t-1})$ may be written as a mixture:

$$\begin{aligned} P(O_c^t \mid A_c^{t-1}) &= P(\neg E)P(O_c^t \mid A_c^{t-1} \neg E) + P(E)P(O_c^t \mid A_c^{t-1} E) \\ &= (1-\epsilon)P(O^{t-1}_{A_c^{t-1}}) + \epsilon\, \mathcal{U}(O_c^t) \\ &= (1-\epsilon)P(O^{t-1}_{A_c^{t-1}}) + \epsilon/2. \end{aligned}$$

– $P(Z_i^t \mid A_c^t\, O_c^t)$ is the *direct model* for sensor i. It yields the probability of a measurement given the occupancy O_c^t and the antecedent (velocity) A_c^t of cell c. Measurements for all sensors are assumed to have been taken *independently from each other*.

For sensors providing measurements depending exclusively of occupancy, this distribution can be written as $P(Z_i^t \mid O_c^t)$. In the same manner, for sensors providing measurements depending exclusively of the velocity, this distribution can be written as $P(Z_i^t \mid A_c^t)$.

2.2. *Occupancy and velocity estimation using the BOF model*

At each time step, the estimation of the occupancy and velocity of a cell is answered through Bayesian inference on the model given in Equation [1]. This inference leads to a Bayesian filtering process. In this context, the prediction step propagates cell occupancy and antecedent (velocity) distributions of each cell in the grid to get the prediction $P(O_c^t\ A_c^t)$. In the estimation step, $P(O_c^t\ A_c^t)$ is updated by taking into account the observations yielded by the sensors $\prod_{i=1}^{S} P(Z_i^t \mid A_c^t\, O_c^t)$ to obtain the a posteriori state estimate $P(O_c^t\ A_c^t \mid [Z_1^t \cdots Z_S^t])$. This allows by marginalization to compute $P(O_c^t \mid [Z_1^t \cdots Z_S^t])$ and $P(A_c^t \mid [Z_1^t \cdots Z_S^t])$ that will be used for prediction in the next iteration.

Figure 2. *Example of BOF output using a computer vision car detector as input (red boxes). The images are provided by a camera mounted on the moving ego-vehicle. The BOF output is projected back on the image. It represents a grid of occupancy probability (blue-to-red mapped color) and the mean velocity (red arrows) estimates.*

It's important to notice that the distribution $P(A_c^t)$ over velocity is updated even when no velocity sensors are available. Indeed, suppose we have only one occupancy sensor described by the model $P(Z_{\text{OCC}}^t \mid O_c^t)$. The a posteriori distribution $P(A_c^t \mid [Z_{\text{OCC}}^t])$ leads to the formula:

$$P(A_c^t \mid [Z_{\text{OCC}}^t]) \propto \sum_{A_c^{t-1} \in \mathcal{A}_c} P(A_c^{t-1}) P(A_c^t \mid A_c^{t-1}) \sum_{O_c^t \in \{0,1\}} P(O_c^t \mid A_c^{t-1}) P([Z_{\text{OCC}}^t] \mid O_c^t).$$

This allows to update the velocity distribution even when no velocity sensors are available. In this case, the update is based exclusively on the occupancy observations.

When an additional velocity sensor $P(Z_{\text{VEL}}^t \mid A_c^t)$ is available, it should be used to update the estimate [2] as follows:

$$P(A_c^t \mid [Z_{\text{OCC}}^t \; Z_{\text{VEL}}^t]) \propto P(A_c^t \mid [Z_{\text{OCC}}^t]) P([Z_{\text{VEL}}^t] \mid A_c^t).$$

3. The "Fast Clustering-Tracking" algorithm

In many applications, the object level representation is demanded. We propose to use a layered architecture as shown in Fig. 1 to obtain this representation. In our former work (Tay *et al.*, 2008), the data association is achieved using a classical JPDA algorithm. However, in a cluttered environment with a large number of moving objects, the JPDA (Bar-Shalom *et al.*, 1988) suffers from the combinational explosion of hypothesises. To overcome this problem, we propose a novel object detecting and tracking algorithm. This algorithm could be roughly divided into a clustering module, an ambiguous association handling module and a tracking and track management module.

3.1. *Clustering*

The clustering module takes the occupancy/velocity grid of the BOF as the input and extracts object level reports from it. A natural algorithm to achieve this is to connect the eight-neighbor cells according to an occupancy threshold $occ\_threshold$. In addition to the occupancy values, a threshold of the Mahalanobis distance between the velocity distributions $vel\_threshold$ is also employed to distinguish the objects that are close to each other but with different moving velocities.

In order to avoid searching for clusters in the whole grid, we use the predicted targets' states as a form of feedback. For a given target with ID id, the predicted state is used to define a region of interest (ROI) in which the clustering process starts. After a starting point with an occupancy probability value greater than the $occ\_threshold$ is found in the ROI, the id is propagated in the ID grid using the connectivity criterion among the non-associated cells (cells with $ID = 0$).

A report for the tracker is a 4-dimentional observation corresponding to the position and the velocity of an extracted cluster. The 2D position component of this vector is computed as the mass center of the region corresponding to the cluster pixels (cells) set. We also compute the corresponding covariance matrix representing the uncertainty of the observed position. The 2D velocity component is simply the weighted mean of the estimated velocities of all cells of the cluster. It comes also with a covariance matrix representing the uncertainty of the observation velocity.

3.2. *Re-clustering and tracks merging*

During the clustering process, three possible situations need to be considered.

– **Case 1:** no cell with $P(occ) \geq occ\_threshold$ is found. The target has not been observed and no association is needed.

– **Case 2:** a cluster C of non-associated cells having $\forall c(i,j) \in C, P(occ(i,j)) \geq occ\_threshold$ is extracted. These cells are then associated to the target id: $\forall c(i,j) \in C,\ ID(c_i, c_j) = id$. This situation occurs when there is no ambiguity in the association. This is an advantageous situation allowing a fast clustering-association procedure. Fortunately, this case is the most frequent one when applying the algorithm to the real data.

– **Case 3:** cells having $P(occ(i, j)) \geq occ\_threshold$ exist. However, they have already been assigned to other IDs. In this conflicted case, an observation (cluster) could be possibly generated by two (or more) different targets.

The first two cases are normal cases, however, the third case is refereed to as an ambiguous association case which need to be dealt with in a special manner. The ambiguous association could occur in the following two situations:

1) Different targets are being too close to each other and the observed cluster is in fact the union of more than one observations generated by different targets.

2) The different tracked targets are corresponding to a single object and should be merged into one.

We take a re-clustering strategy to deal with the first situation and a cluster merging strategy to deal with the second one.

Suppose when an ambiguous association occurs, a set of tracks $T_1, T_2, \cdots, T_m$ are the potential candidates to be associated to the observed cluster. We have to cut up the extracted cluster and generate a sub-cluster (possibly empty) for each candidate. This re-clustering is achieved by a k-means (Bishop, 2006) algorithm using a simple Cartesian distance. The considered distance is taken between the center of the sub-cluster and a given cell. In this way, the first cause of the ambiguous association is handled.

To deal with the second cause of the ambiguous association, we introduce a concept of "alias" which is in the form of a two-tuples to represent the duplicated tracks. When an ambiguous association between two tracks T_i and T_j is detected, an alias $ALIAS(T_i, T_j)$ is initialized and added to a potential alias list.

At each frame, the tracker updates this list by confirming or disproving the existence of each alias hypothesis $ALIAS(T_i, T_j)$ according to the observation of the ambiguous association. At a given time step t, if the ambiguous association occurs between T_i and T_j, and the alias $ALIAS(T_i, T_j)$ is found in the potential alias list, the probability $P^t\left(S(T_i, T_j)\right)$ is increased by a confirming step using a Bayesian filtering approach as follows:

$$P^t(S \mid F) = \frac{P^{t-1}(S) \times P(F \mid S)}{P^{t-1}(S) \times P(F \mid S) + [1 - P^{t-1}(S)] \times P(F \mid \neg S)},$$

where:

– $S \equiv$ "the T_i and T_j tracks are alias for the same object".

– $F \equiv$ "an ambiguous association between the tracks T_i and T_j is observed".

The probability values $P(F \mid S)$ and $P(F \mid \neg S)$ are constant parameters of the tracker. The former denotes the probability of observing an ambiguous association when the two concerned tracks are alias of the same object and is set to a constant value 0.8. The second denotes the probability of falsely observing an ambiguous association and is set to 0.1.

When $ALIAS(T_i, T_j)$ is found in the potential alias list but is not observed as an ambiguous association, its probability is decreased in a similar manner:

$$P^t(S \mid \neg F) = \frac{P^{t-1}(S) \times P(\neg F \mid S)}{P^{t-1}(S) \times P(\neg F \mid S) + [1 - P^{t-1}(S)] \times P(\neg F \mid \neg S)}.$$

According to the probability $P^t\,(S(T_i, T_j))$, the decision of merging of tracks T_i and T_j could be made.

3.3. *New tracks creation*

For new targets creation, we introduce a concept “cluster seed” to define a cell in the BOF grid where we will try to find, for each step, a new (non-associated) cluster. Indeed, the searching for potential new targets is after all the existing tracks are processed. Thus, only non-associated cells will be processed to extract clusters as the observations for the potential new targets. The “cluster seed” concept is general and can be implemented via various strategies. The simplest strategy is to insert a possible seed in each cell of the grid. However, more sophisticated strategies could be more efficient. For example, cluster seeds could be inserted only in entrance regions of the monitored area.

3.4. *Tracks updating and deleting*

The prediction and estimation of the targets are accomplished by attaching a Kalman filter (Kalman, 1960) with each track. Once associated to a given track, a report (Gaussian distributions for both position and velocity) corresponding to an extracted cluster is used as an observation to re-estimate the position and velocity of the track in a prediction-update step. For non-observed tracks, only a prediction step is taken by applying the dynamic model to the estimation result of the precedent time step.

The deleting of tracks is also achieved in a Bayesian manner. If an existing track T is associated with a given report (cluster), its existence probability is increased using the following formula:

$$P^t(E \mid O) = \frac{P^{t-1}(E) \times P(O \mid E)}{P^{t-1}(E) \times P(O \mid E) + [1 - P^{t-1}(E)] \times P(O \mid \neg E)},$$

where:

– $E \equiv$ “the target T exists”.

– $O \equiv$ “the target T has been observed (associated)”.

The parameters $P(\neg O \mid E)$ and $P(O \mid \neg E)$ are the tracker miss-detections and false alarms probabilities respectively.

If an existing target is not associated with any report (cluster), its existence probability is decreased in the similar way:

$$P^t(E \mid \neg O) = \frac{P^{t-1}(E) \times P(\neg O \mid E)}{P^{t-1}(E) \times P(\neg O \mid E) + [1 - P^{t-1}(E)] \times P(\neg O \mid \neg E)}.$$

According to the existence probability, the track deleting operation is achieved by applying a deleting threshold on it.

4. Conclusion

We presented a novel sensing/tracking algorithm for the BOF framework. This algorithm takes the occupancy/velocity grid of the BOF as input and extracts the objects from the grid with a clustering module which takes the prediction of the tracking module as a feedback to reduce the computational cost. A re-clustering and merging module is proposed to deal with the ambiguous data associations. The extracted objects are then tracked and managed in a probabilistic way. The experiment results show that the presented algorithm is robust as well as computationally efficient so as to be suitable for cluttered environment.

Our approach has been applied on real data and achieved satisfied results in several conditions. The proposed algorithms have been used in several driving assistance projects in both highway and cluttered urban environments. The used sensor modalities include: (i) multi-layer lidars, (ii) computer vision detection algorithms (Fig. 2), and (iii) stereovision-based 3D sensors.

According to confidentiality agreements of the on-going projects, we could not provide these works in the publications. However, we are now preparing new experiments using our own data sets in order to be able to present quantitative experimental results in future publications.

5. References

Bar-Shalom Y., Fortmann T. E., *Tracking and Data Association*, Academic Press, 1988.

Bishop C., *Pattern recognition and machine learning*, Springer, 2006.

Jazwinski A. H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.

Kalman R. E., “ A New Approach to Linear Filtering and Prediction Problems”, *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, n˚ Series D, p. 35-45, 1960.

Moravec H. P., “ Sensor fusion in certainty grids for mobile robots”, *AI Magazine*, vol. 9, n˚ 2, p. 61-74, 1988.

Tay C., Mekhnacha K., Chen C., Yguel M., Laugier C., “ An Efficient Formulation of the Bayesian Occupation Filter for Target Tracking in Dynamic Environments”, *International Journal Of Autonomous Vehicles*, vol. 6, n˚ 1/2, p. 155-171, 2008.

Thrun S., Burgard W., Fox D., *Probabilistic Robotics*, The MIT Press, 2005.