Robotica: (2020) volume 38, pp. 761–774. © Cambridge University Press 2019
doi:10.1017/S0263574719001024

Detection and Tracking of Moving Obstacles (DATMO): A Review

Ángel Llamazares†*, Eduardo J. Molinos‡ and Manuel Ocaña†

†*Department of Electronics, University of Alcalá, Alcalá de Henares, Spain*
‡*Institut für Mess - und Regelungstechnik, Karlsruhe Institut fur Technologie, Karlsruhe, Germany*
Email: eduardo.molinos@kit.edu
†*Department of Electronics, University of Alcalá, Alcalá de Henares, Spain*
Email: mocana@depeca.uah.es

(Accepted May 29, 2019. First published online: July 12, 2019)

SUMMARY
Working with mobile robots, prior to execute the local planning stage, they must know the environment where they are moving. For that reason the perception and mapping stages must be performed previously. This paper presents a survey in the state of the art in detection and tracking of moving obstacles (*DATMO*). The aim of what follows is to provide an overview of the most remarkable methods at each field specially in indoor environments where dynamic obstacles can be potentially more dangerous and unpredictable. We are going to show related DATMO methods organized in three approaches: model-free, model-based and grid-based. In addition, a comparison between them and conclusions will be presented.

KEYWORDS: DATMO; Grid-based; Model-free; Model-based.

1. Detection and Tracking of Moving Obstacles

In order to achieve a safe navigation in a partial or completely unknown environment, also including possible dynamic objects, the classical mapping methods are usually not enough. They only take into account the occupancy and not dealing with dynamic obstacles.

For that reason, the robot has to manage a representation of the world that be achievable, that include information to predict the future of the environment, continuously update and with a reliability level. There are different approaches that can deal with dynamic obstacles, called detection and tracking of moving obstacles (*DATMO*) methods:

1. The first kind of methods is based on the assumption acknowledge of all objects around the robot, learning their tracks and velocities on an *off-line* process. These methods are only applicable in structural and constant environments, such as the industrial ones. For that reason, they are out of the aim of this paper.
2. The second kind of approaches try to estimate the position and velocity of the objects in an *on-line* process. Usually in two steps: *data association* phase and a *multi-target tracking* algorithm. These methods do not take into account the unknown space and can suffer problems in a crowded environment. These methods are *Objects-based DATMO* (Section 1.1) and can be divided in *Model-free* and *Model-based* approaches, described in Sections 1.1.1 and 1.1.2, respectively.
3. The third alternative to these classic *object framework* is the methods that maintain a probabilistic knowledge of the environment in a *dynamic occupancy grid* domain, including the actual occupation and the estimation of the cell's velocities. These methods are the *Grid-based DATMO* approaches, described in Section 1.2.

* Corresponding author. E-mail: angel.llamazares@uah.es

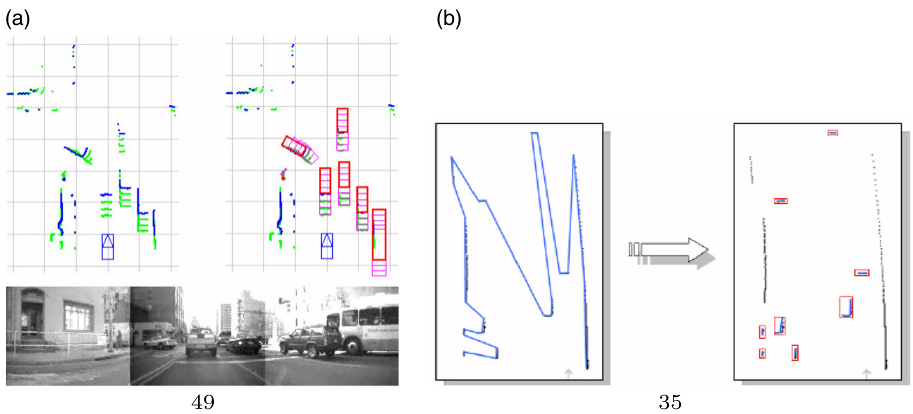

Fig. 1. DATMO examples: LIDAR-based object detection. (a) Ref. [1]; (b) Ref. [2].

1.1. Objects-based DATMO approaches

The *DATMO* problem has been under research in the last decades. *DATMO* is the process of observe the states of objects surrounding in a dynamic environment with the exteroceptive sensors, assuming an accurate pose estimate available, to obtain the position and the trajectories of the dynamic objects. Usually, the objects-based approaches try to track different objects independently, keeping them in a list. This is a *multi-objects* or *multi-target* tracking problem, whose main difficulty is the *data association* problem. This problem relies on the knowledge of:

- Whether new measurement from sensors corresponds to an existing object.
- When an object should be: *created* as new object, *maintained* or *deleted*.

Many works focus on the *multi-objects* tracking problem assuming that the measurements are uniquely from moving objects, although static objects or spurious measurements exist in most of the real applications. The sensors widely used in this area are the range sensors (sonar or *LIDAR*), or cameras, although the *LIDAR* is usually the main sensor due to its high accuracy and resolution to detect objects. Some illustrative examples are shown in Fig. 1.

Previously to the *data association* stage, is necessary segment the data. The simplest way is the *clustering*, based on the range discontinuities of the data. Others approaches trying to search some features in the data, such a *lines* or *"L" shapes*. The clusters have to be assigned to the target, taking into account that multiple clusters can come from the same target.

The *data association* stage can be computed with simple methods, such as *Nearest Neighbor Rule (NNR)*, assigning each data cluster to the closest target, widely used at high enough update rates to assume the assignment unambiguous. One step forward are some variations of these methods, as *Global Nearest Neighbor (GNN)*, that ensures that each cluster is associated to one target. This method is appropriate in recognition-based approach, where each cluster represents the whole object. In the case that the ambiguity should be high, there are other commonly used methods:

- *Multiple Hypothesis Tracking (MHT)* by ref. [3]: This algorithm is a *hypothesis-oriented MHT* that can handle measurements that came from a varying number of targets, maintaining multiple association hypotheses between clusters of measurements and targets and keeping it between consecutive scans. For that reason the number of possible hypotheses increases each time step due to new possible hypotheses are created from initializing new target from each new measurement in addition to the previously existing hypotheses. On the other hand, exists the *track-oriented MHT* that only kept a selected targets trees, where each tree contains a number of tracks which are not compatible. In order to evaluate the hypotheses formation, including all the data association aspects,[4] proposed the track scored called log-likelihood ratio *LRR*, mathematically equivalent to the probabilistic expression by ref. [3], where each hypothesis obtains a score summing the track existence scores of all targets within it. As the combinatorial increase, being necessary techniques to reduce them, such as tracks scoring techniques[5,6] or N-scan pruning[7] or an algorithm presented by ref. [8] where the N-best hypotheses are determined in a polynomial time.

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

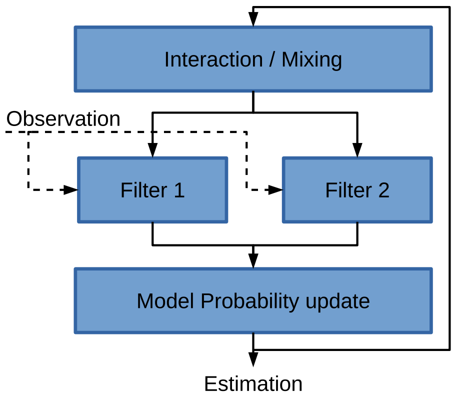

Fig. 2. IMM filter diagram.

- *Joint Probabilistic Data Association Filter (JPDAF)* by ref. [9]: This algorithm only maintains one association hypothesis, similar to *GNN*, avoiding the problem of the increasing combinatorial. The association is not hard as in *GNN*, is a soft assignment based on the probability of each measurement being associated to each target. For target k the measurement update is taken into account all possible assignments the probability of data clusters (Eq. 1). The probability that the measurement m is from target k be denoted by β_{mk}:

$$p(Z_t|S_t^k) = \eta \sum_{m=1}^{M_t} \beta_{mk} p(Z_t^m|S_t^k) \tag{1}$$

where K is the number of targets, M_t the data clusters $(Z_t^1, \ldots, Z_t^{M_t})$ and η is a normalization constant. There are some variants and extensions of *JPDAF*.[10] The variants that are used in *DATMO* approaches rely on parametric belief representations.

The *target tracking* stage in *DATMO* is usually performed with a Kalman filter (KF)[11] approach; this kind of filter is recommended when we are working with Gaussian noise processes of zero mean in linear dynamic system and observation models. Unfortunately, in most applications of interest the system dynamics and observation equations are nonlinear and suitable for the extended Kalman Filter (EKF).[12] In addition, when occlusion occurs in nonlinear motion scenarios, even these methods often fail to moving objects of interest. Considering the ambiguity caused by the occlusion among multiple moving objects, unscented Kalman filtering (UKF)[13] technique can be applied for reliable object detection and tracking. Different from conventional KF, which cannot achieve the optimal estimation in nonlinear tracking scenarios, UKF can be used to track both linear and nonlinear motions due to the unscented transform. Authors initially detect moving targets by means of an efficient block matching technique, providing rough location information for multiple object tracking. Then, more accurate location information is estimated for each moving object by a nonlinear tracking algorithm. These parametric filters are computationally cost-efficient, but it cannot represent complex beliefs when arise due to the data ambiguities. In contrast, there are non-parametric filters, such as particle filter (PF),[14] that represent the belief by a set of particles that can represent an arbitrary belief. Nevertheless, the computational cost is exponential in the dimensionality of the state and usually the base of the exponent is large.

Other approaches use *multiple model-based* moving object tracking. Due to the lack of a priori information, the on-line mode learning is a hard task. For that reason, the motion mode of moving objects usually is approximately composed of several motion models such as constant velocity model, constant acceleration model and turning model. In this way, the mode learning problem is simplified to a multi-model selection problem. Even so, due to the time variant of the model of the moving obstacles, it is necessary a suboptimal approach with merge or reduce the number of the mode history hypotheses to avoid the exponentially increasing of them. *Interactive Multiple Model (IMM)* filter[15] is widely uses in that case. This approach computes the state estimate at time k under each possible current model using a suitable mixing of the previous model-conditioned estimate as the initial condition at each filter, as shown in the *IMM* diagram in Fig. 2.

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

On the other hand, the *DATMO* problem can be addressed as a batch problem over a fixed time window. It could be solved using different approaches based on *Markov Chain Monte Carlo (MCMC)*.[16] These methods start with a desired probability distribution, then produce samples of them by constructing a Markov chain with an equilibrium distribution equal to the desired distribution. The resulting state of the chain can be used as a sample from the desired distribution once the methods walk the chain a enough steps. To construct the Markov chain is usually easy, while obtain the number of steps needed can be difficult.

1.1.1. DATMO model-based approaches. These methods know the class of the objects to be detected a priori. With this information, each object is detected based on a parametric model (of its shape) to track it separately. These methods have the restriction to detect only the specific objects described in the parametric model.

Some of these methods focus on people detection, for example, in work,[17] authors train a boosting classifier to detect legs separately with *KF* and constant velocity models, then, they group the legs of each person to track them with an *MHT*. Other authors[18] only detect the legs of people on the 2D range scanner and, to robust the tracking, apply the *JPDAF*. Topp and Christensen[19] extend the previous work detecting the people whose legs are not directly visible.

Other works, instead of detecting people, focus on detecting and tracking vehicles, such as ref. [1] that represents the vehicles with a box model. The data association and tracking of the dynamic obstacles were implemented with *GNN* and *KF*. Also, the authors present an algorithm that optimizes the box models over the best trajectories of the vehicles in a sliding window of laser scans to solve detection and tracking simultaneously using a *Data-Driven Markov Chain Monte Carlo (DDMCMC)* algorithm. The authors improve the odometry with a fast incremental scan matching. The system was tested with Navlab data set.[20]

Another example of model-based vehicle detection and tracking is the work by Zhao and Thorpe;[21] authors proposed an *IMM* filter with three different motion models to determinate all of possible motions of the vehicles that are tracked. On the other hand, the work by Granström[22] does not focus on people or vehicles; he used a *PHD* filter to detect and track rectangular and elliptical targets. Chavez-Garcia and Aycard[23] classified the objects in four classes of interest: pedestrian, bike, car or truck, using Adaboost by ref. [24] (a boosting-based learning algorithm). In addition, they used an *IMM* with constant velocity, acceleration and turning models to track the objects and a pruned *MHT* to perform the data association.

1.1.2. DATMO model-free approaches. These methods do not impose restrictions in obstacle's type or shape and the semantic information about the object is not needed. On the other hand, these methods only detect the current moving obstacles, not the potentially dynamic object that will move in the future.

Some of these methods, included in the systems that participate in the *DARPA Urban Challenge*,[25,26] work as follows: first, they segment different measurements from several lasers with a basic region-growing algorithm in segments to extract geometric features of the objects (corners and line ends of the *L shapes*); second, with these features, they compose a list of possible objects and, finally, they extract the objects that have a significant speed, tracking them with a *KF* with constants turn rate and acceleration and a fixed process noise. The output velocity has a delay that varies between 1.1 and 3.5 s.

Others methods are based on estimating a static map of the environment at the same time that detect the moving obstacles using the knowledge of occupancy probabilities to compute the likely moving objects. In ref. [27], the authors represent data in grid maps computing an *Iterative Closest Point (ICP)* algorithm, then an *MHT* algorithm is used for data association and an *IMM* algorithm for tracking. The approach was tested in crowded urban environments at high speeds. Though, the method proposed by the authors need to include a prior digital map, not available everywhere. This work is extended by ref. [28], estimating the robot pose at the same time that difference the dynamics and statics objects with an extended *Iterative Dual Correspondence (IDC)* algorithm and integrating the identification of dynamics objects within the estimation process. Also, an *NNR* is used for data association. Other similar work is the Toyota's tracking system[29] that combines *Simultaneous Localization And Mapping (SLAM)* with the tracking of moving obstacles.

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

In ref. [30], the authors improve the odometry with a fast incremental scan matching and the occupancy map that is based on the grid framework by Elfes.[31] The dynamic obstacles are detected taking into account their inconsistencies with the grid map. Finally, an *MHT* with an adaptative *IMM* are used for the data association and tracking stages.

Using map differences between the occupancy map of the static obstacles, they can detect moving obstacles and then identify the objects with an *Expectation-Maximization (EM)* algorithm.[32]

In the work by Yang and Wang[33] authors used a variant of *Random Sample Consensus RANSAC* to estimate the robot pose and the moving obstacles and a decision tree based on spatiotemporal consistency tests to manage the track merging and splits. Hahnel et al.[34] proposed an *EM* algorithm to determine whether a laser point is static or dynamic solving a set of hidden indicator variables of each laser point. The same authors[35] presented a probabilistic technique to create maps that contain people. To track people, the algorithm implemented is an *Sample-based Joint Probabilistic Data Association Filters (SJPDAF)* using the laser measurements. To identify the moving people and the static obstacles, the authors used an occupancy probability maps.

Other approach is to focus on the detection part, solving the data association and moving object detection problems using a joint *Conditional Random Field (CRF)* framework, as in the works of refs. [36] and [37]. In ref. [38], the authors dealt with the data association in two levels: a coarse level predicting the boundary points of the objects computing an *Euclidean Minimum Spanning Tree (EMST)-Efficient Graph-Based Image Segmentation (EGBIS)* clustering algorithm, then computed an *ICP*, and presented a variant of *JCBB (Joint Compatibility Branch and Bound)* in the fine level.

1.2. Grid-based DATMO

These methods represent the environment as an occupancy grid, where each cell is tracked at a sub-object level, instead of segment the environment into objects to track them. These approaches avoid the *object concept*, also the problem of *multi-object* detection and tracking, sometimes very difficult to solve. Other advantage of these methods is that the sensor's data fusion from different sensors can be computed at the raw data level using occupancy grid maps, where the data association is not needed, unlike in the previous *DATMO* approaches.

One of the most popular approaches at the base of several works is the *Bayesian Occupancy Filter (BOF)* by ref. [39]. The *BOF* evaluates the environment occupancy regardless the kind of the object. The occupancy of the cells is computed as an probabilistic formulation initially proposed by Elfes,[31] dealing with uncertainty due to the noise or the inaccurate sensors through the *Bayes theorem*.

On the other hand, there are others approaches that are based on the *Evidence Theory*, also called *Dempster–Shafer theory*[40] taking into account the inconsistencies between consecutive grids as evidences of conflict. This method has the advantage of modelling a cell in three states: *free*, *occupied* or *not observed yet*. Although, a few works[41,42] are based on these methods, due to tuning the parameters needed are usually a challenging task. For that reason this paper is focused on the methods based on *BOF*.

The *BOF* is described as a representation of the space in a cell's grid. Each cell describes a portion of the environment and includes different kinds of information about this part of the space, such as occupancy, velocities, riskiness, so on. This representation has several advantages:

- To be a generalist representation allows to fuse the information from different sensors and using in indoors and outdoors.
- The use of *prediction* and *estimation* stages taking into account the previous occupancy history helps to overcome some occlusions.
- The probabilistic framework based on *Bayes theorem* handles the uncertainty and the noise in the sensor's measures, increasing the robustness.
- As the cells are independent, this allows to parallelize processes, increasing the performance.

The work by Saval-Calvo et al.[43] defined the five layers of the *BOF* corresponding to the five main parts, from the bottom layer closer to the raw data sensor to the top one, where the high-level algorithms are implemented, shown in Table I.

The definition of *BOF* following the description in ref. [44] is as follow: the 2D Euclidean space is divided into a finite number of cells, each represents a position in the plane. The state of the system $O(t)$ at time t is the list of the states of all the cells of the grid: *Occ*, when the cell is *occupied* or *Emp* if

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

Table I. Levels of BOF.

Level	Task
5th	High-level applications
4th	Clustering
3rd	BOF core
2nd	Grid estimation
1st	Pre-process

the correspondent space is *free*. Given a probabilistic sensor model $P(z(t)|o(t))$ where $z(t)$ is the current observation, the grid is updated following the Bayes rule. Under the hypothesis that each cell of the grid is independent from its neighbour cell, each cell state estimation is updated independently.[31] To handle dynamic obstacles, each cell of the *BOF* maintains not only an estimation of its occupation probability, but also a discretized representation of the *Probabilistic Distribution Function (PDF)* over velocities. A minimum and maximum velocity value is considered for eventual objects in the space, and the *PDF* is approximated by a finite histogram over regularly distributed velocity values v_n with $n \in 1...N$. The discretization step is chosen according to spatial and time discretization: given q the size of a cell and τ the time step, only integer velocities in terms of $\frac{q}{\tau}$ are taken into consideration. This choice is necessary in order to perform fast and rigorous prediction and updating steps.

The variables used to formalize the *BOF* probability estimation are as follows:

- C is an index that identifies each 2D cell of the grid.
- A is an index that identifies each possible antecedent of the cell c over all the cells in the 2D grid.
- $Z_t \in Z$ where Z_t is the random variable of the sensor measurement relative to the cell c.
- $v \in V = v_1, \dots, v_n$ where v is the random variable of the velocities for the cell c and its possible values are discretized into n cases.
- $O, O^{-1} \in O \equiv occ, emp$ where O represents the random variable of the state of c being either *occupied* or *empty*. O^{-1} represents the random variable of the state of an antecedent cell of c through the possible motion through c. For a given velocity $v_k = (v_x, v_y)$ and a given time step δt, it is possible to define an antecedent for $c = (x, y)$ as $c^{-k} = (x - v_x\delta t, \quad y - v_y\delta t)$.

The decomposition of the joint distribution of the relevant variables according to Bayes' rule and dependency assumptions as state[45] is given by Eq. (2):

$$\begin{aligned} &P(C, A, Z, O, O^{-1}, V) = \\ &\quad P(A)P(V|A)P(C|V, A)P(O^{-1}|A)P(O|O^{1})P(Z|O, V, C) \end{aligned} \tag{2}$$

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

- $P(A)$ is the distribution over all the possible antecedents of the cell c. It is chosen to be uniform because the cell is considered reachable from all the antecedents with equal probability. Consequently, given k antecedents, each one has a probability $P(A) = \frac{1}{k}$.
- $P(V|A)$ is the distribution over all the $\{v_1, \dots, v_n\}$ possible velocities of a certain antecedent of the cell c; its parametric form is a histogram.
- $P(C|V, A)$ is a distribution that explains whether c is reachable from $[A = a]$ with the velocity $[V = v \in \{v_1, \dots, v_n\}]$. In discrete spaces, this distribution is considered a Dirac with value equal to 1 if and only if $c_x = a_x + v_x\delta t$ and $c_y = a_y + v_y\delta t$, which follows a dynamic model of constant velocity. This Dirac distribution is used in the *BOF* by ref. [39], nevertheless, other general distribution approaches could be used.
- $P(O^{-1}|A)$ is the distribution over the occupancy of the antecedents. It gives the probability of the possible previous step of the current cell. Given the antecedents, this probability explains that the previous occupancy is reliable with the current antecedents.
- $P(O|O^{-1})$ is the conditional distribution over the occupancy of the current cell, which depends on the occupancy state of the previous cell. It is defined as a transition matrix: $T = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix}$

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

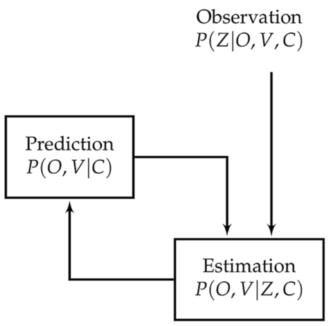

Fig. 3. BOF stages: prediction and estimation.[46]

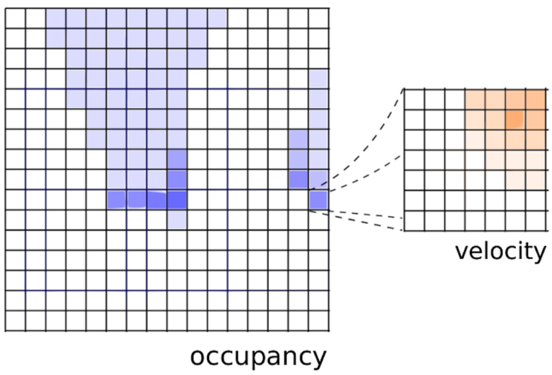

Fig. 4. BOF grid representation.[47]

which allows the system to use the null acceleration hypothesis as an approximation; in this matrix, ε is a parameter representing the probability that the object in c does not follow the null acceleration model.

- $P(Z|O, V, C)$ is the conditional distribution over the sensor measurement values. It depends on the state of the cell, the velocity of the cell and obviously the position of the cell.

The Joint probability $P(O, V|Z, C)$ can be decomposed into observation and prediction probabilities, estimating in this way the occupancy and velocity probabilities of each cell, following the *prediction/estimation* diagram shown in Fig. 3.

The authors discretized the possible velocities. In order to implement the probability distribution, it is possible to do it in form of histograms, as shown in Fig. 4.

1.2.1. Variants and improvements of the BOF approach. Taking into account the *BOF* layers described by ref. [43], several authors had proposed improvements or variants in the different layers. The **first layer** is referred to the *processing the data* from the sensors before computing the grid. Usually, the sensors used in occupancy grids are range (*LIDAR* or sonars) sensors, sometimes also cameras. As these sensors do not provide the measures in a grid space, it is necessary to process these data.

Linear Opinion Pool (LOP) techniques are commonly used to treat the data. In this way, it is possible to avoid that the non-reliable data affect the global result, having a sensor model and a weighting term. The authors in ref. [48] used *LOP* in multi-sensor fusion, such as *LIDAR* and stereo cameras. Other authors,[49] fused multiple layers from laser scanner using *LOP*.

In the **second layer** (*Grid-estimation*): The authors of ref. [50] used the difference in the occupancy between time steps to compute the velocity, keeping the computation of the object detection and tracking in the occupancy grid. In ref. [46], the authors introduced velocity information reducing noise in estimations, improving the predictability and reducing the computational cost. While ref. [39] represents the velocity in two dimensions (V_x, V_y), corresponding to the two possible directions, increasing the dimensions of BOF to *4D BOF*, Chen et al.[50] introduce a histogram of velocities in the

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

cell (*2D Histogram BOF*). The *4D BOF* has the advantage that represent objects that overlap others and the *2D Histogram BOF* have a less computational cost.

Other variant introduced by Yguel et al.[51] is taking into account a non-constant velocity. It is especially important at urban or indoor scenarios where the objects do not have constant velocities. To afford this, the cells are updated at different frequencies, based on their velocities. Additionally, the authors handled the aliasing problem; depending on the objects orientation, the number of cells occupied by them is different.

The authors in ref. [49] added a motion detection stage to recognize the moving obstacles, based on keeping update a free and occupied count arrays, that means the number of times that each cell has been free or occupied. With that counts and an heuristic, the system is able to identify the moving part in the environment.

Variants on the core of the *BOF* method (**third layer**) have been introduced by several authors, reducing the computational cost using a prior knowledge of the environment or representing the dynamism of the cells using PFs. The authors in ref. [52] proposed a *Bayesian Occupancy grid Filter for dynamic environments Using prior Map knowledge (BOFUM)* that predicts the more precise cells movements taking into account a prior knowledge about the cell environment. The object motion is usually dependent on its localization, restricting his movements based on behaviour patterns. For example, in urban scenarios, it is more probable that the cars follow the lanes than crossing them perpendicularly, neither they drive on the sidewalk. On the other hand, the movements of pedestrians are not so restricted. The authors applied a reachability matrix that contains the changing probability of the cells based on the area. They defined three possible areas (lane, sidewalk and unknown) with a cost function to compute the probability of one cell being the antecedent of another based on three assumptions:

- The area changing is unlikely.
- If an object is moving off the lane and the antecedent cell is in other areas, it is highly probable to be a pedestrian.
- The vehicles usually moving on the lane and the moving out the lane or lane change probabilities are low.

Other authors used *BOFUM*; in ref. [53], the authors accelerated the convergence of the *BOF* using a prior map knowledge. Also, in this work including uncertainty in the motion model and a importance sampling (*IS*) recursively to approximate *BOFUM* calculations, it is similar to the *PF* in a discrete cell framework. With this approach, the authors improved the time performance of the system at least 40 times. Other improvement added is the property of the cells that can move in object groups, defined as *Bayesian Occupancy Filter Using Groups (BOFUG),* allows to infer object classes only from the occupancy measurement. On the other hand, the disadvantages of the systems based on *BOFUM* are that the required maps are not always available and the aliasing problem is not resolved.

Another variant is to use *PF* to obtain the occupancy and velocity of the cells. In ref. [54], the authors defined an algorithm to compute the environment dynamics with a variable number of particles per cell depending on its occupancy. The sensors used are stereo vision and the approach is based on three stages: first, the prediction, reallocating the particles taking into account the speed and ego-motion of the vehicle; second, processing the measurement incorporating the measured data to weight the particles and re-sample them; and finally, estimation of the occupancy and velocity. The moving particles represent the environment without discretization in its positions, correcting in this way some aliasing problems. They compared the results with the *Lucas–Kanade* Optical Flow. This method has been later named as *Sequential Monte Carlo-Bayesian Occupancy Filter (SMC-BOF)*. Based on this approach, authors in ref. [55] proposed a data fusion between laser and radar.

Based on the same idea of using *PF* and to increase the performance, the authors in ref. [47] introduced the *Hybrid Sampling Bayesian Occupancy Filter (HSBOF)* to avoid the high number of particles unnecessarily used in the case of static cells. The authors did not use particles in the static cells neither in the free areas, mixing static occupancy based on the original *BOF* and dynamic occupancy modelled with a moving particles with the same *PF* principle. The particles in the dynamics cells are composed of a velocity vector and an exact position, to avoid aliasing. Also, each particle has a weight associated to represent the distribution of the speed in the cell.

As it is shown in Fig. 5, each cell can be decomposed into three sections: *static occupied*, *dynamic* and *free*.

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

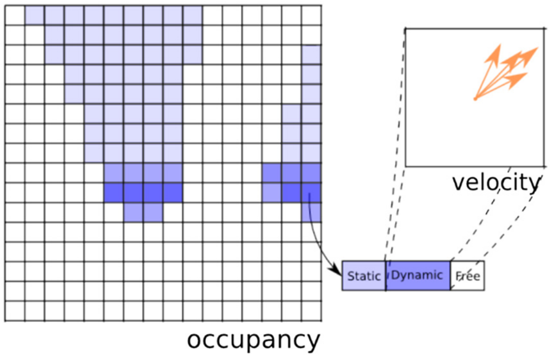

Fig. 5. HSBOF grid representation.[47]

This approach has velocity detection problems in the case of linear geometry objects, such as the highway median strip, that are detected as dynamic.

Although in the *BOF* framework the *object concept* does not exist, in some applications the representation of object or track is necessary. In this case, some authors added a level of *clustering* above the *BOF* in order to extract the objects. These variants are located in the **fourth layer** (*Clustering*) of Table I.

The work by Mekhnacha et al.[56] proposed clustering the cells of the *BOF* in objects. The input of the tracker is the occupancy and velocity grids of the *BOF*. The first approach that they test is a layer with a *Joint Probabilistic Data Association Filters (JPDAF)* above the *BOF* layer, but the hypothesis number that the *JPDAF* generates increases rapidly. In order to decrease the computational cost, they proposed the *Fast Clustering-Tracking Algorithm (FCTA)*. The tracker creates a grid with the same size of the *BOF* grid that contains the corresponding identifier between each cell and the cluster to which it belongs. The algorithm is divided into three modules; *clustering*, an eight-neighbour approach with an occupancy and velocity thresholds to classify the cells in groups; *re-clustering and merging* that handles the ambiguity in association and a *track module* based on a *KF*. They test the system with two pedestrian and a car using a SICK *LIDAR* and can manage the occlusions between the pedestrians.

Finally, in the **fifth layer** (*High-level applications*) of Table I are located the works that use the data output of the *BOF* as input of their system. In this way, the authors take advantages of the robustness on the occupancy computation and the capacity of handling the data fusion from different sensors. Some of these works are ref. [30] explained in Section 1.1.2 and ref. [1] explained in Section 1.1.1. Based on *HSBOF*, the work by Yuan et al.[57] combines the information of the *LIDAR* used for occupancy state updating only and the RADAR measurements that are affordable for likelihood evaluation. This approach leads more accurate dynamic estimates and therefore guides the particles flow in a correct and accurate manner. The authors detected the dynamic obstacles in a driving scenario and their movement, but they did not show the numerical velocity of them. The pedestrians are not well detected due to the move–stop–move behaviour.

2. Discussion

In the previous sections, a survey of the *DATMO* approaches has been introduced. This section presents a comparison and a discussion of the most important works on this area.

DATMO is a rapidly developing field, especially in mobile robots and autonomous vehicles. It is still an open problem, due to a *DATMO* fully autonomous system that can rival the human capabilities in all the cases is not yet available. However, it is not easy to compare the methods in a quantitative way due to a lack of numeric results or difficulty of a direct comparison using a common data set. For that reason and in order to show a general view of the performance of the different methods, the comparison has been divided into two tables:

- Table II shows a comparison of some of the *Objects-Based* algorithms referred in Section 1.1. In this case, the comparison is based on some parameters, such as the *sensors* used, the *data association* and *tracking* methods, the *object model* used and the *different dynamic objects* that can be handled.

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

Table II. Object-based DATMO methods comparison

Object-based DATMO methods								
	DATMO method		**Dynamic objects**					
Author	**Data association**	**Tracking**	**Model**	**Kind**	**Sensors**	**Environment**	**2D/3D**	**DATMO contribution**
[27]	MHT	IMM	Free	People, cars, bikes, buses	Laser, odometry	Outdoor	2D	Pioneer of *DATMO*
[35]	SJPDAF		Free	People	2D SICK Laser Tilting 3D Laser, 2D Laser	Indoor Outdoor	2D 3D	Reduce spurious objects with a more robust algorithm
[28]	NNR	EKF	Free	People, doors	SICK 2D Laser, odometry	Indoor	2D	Differentiated static and dynamic objects in estimation problem
[1]	DDMCMC		Based	People, cars, bikes, buses	2D Laser	Outdoor	2D	DDMCMC
	GNN	KF	Based	Objects that can cause collisions in traffic	2D laser, odometry, 2 short range radars			–
[30]	MHT	IMM	Free	People, cars	2D laser, odometry, 2 short-range radars	Outdoor	2D	–
[38]	EMST-EGBIS, ICP + Variant of JCBB	EKF	Free	People, bikes, cars, buses, trucks	SICK LDMRS, odometry	Outdoor	2D	Variant of JCBB and EMST-EGBIS clustering algorithm
[23]	pruned MHT	IMM	Based	People, bikes, cars, trucks	LIDAR, Sonar, cameras	Outdoor	2D	Enhanced representation (kinetic+appearance) at the detection level

https://doi.org/10.1017/S0263574719001024 Published online by Cambridge University Press

Table III. Grid-based DATMO methods comparison

Grid-based DATMO methods									
Method - Author	**Computational Cost**	**Parallelization Possibility**	**Velocity estimation**	**Aliasing robustness**	**Representation free space**	**Accuracy/noise ratio**	**Prior knowledge needed**	**Handling dynamic obstacles**	**DATMO contribution**
4D BOF[39]	Very high	High	Medium	Low	Simple	Low	No	Medium	Pioneer of Grid Based *DATMO*
BOFUM[52]	High	Medium	High	Medium	Simple	Medium	Yes	High	Introduced map restrictions to reduce computational cost
BOFUG[53]	Medium	Medium	High	Medium	Simple	Medium	Yes	High	Added the cells property that can move in object groups
SMC-BOF[54]	Medium	Low	High	High	Complex	High	No	High	Introduced the *PF* to the cells
HSBOF[47]	Low	Medium	High	High	Simple	High	No	High	Mixed the static part of *BOF* with the dynamic part based on *PF*

- Table III shows a comparison of some of the *Grid-Based* methods referred in Section 1.2. It is not easy to compare the different variants and improvements based on *BOF* approach due to their different focuses and the lack of similar experiment and numeric results. For that reason, the comparison is based on performance parameters, such as *computational cost* (memory usage and speed), *possibility of parallelization* in dedicated hardware, *estimation of the velocity* of the cell, the *aliasing robustness* due to the grid discretization, the simplicity of the *free space representation*, the ratio between the *occupancy probability accuracy* and the unexpected noise that appears in the free cells, the *prior knowledge needed* of the map and the capability to manage dynamic obstacles.

3. Conclusions

Several conclusions can be drawn from the review of DATMO methods and the analysis carried out in previous sections and the information shown in Tables II and III.

Objects-Based Methods do not take into account the unknown space. Also they can suffer problems in a crowded environment, due to some of the data association methods, such as *MHT* and *JPDAF,* suffer a rapidly combinatorial increase (*combinatorial explosion of hypothesis*), no works have been found using Probabilistic Multi-Hypothesis Tracking (*PMHT*) that can be useful in crowded environments. Other disadvantage is that these methods do not take into account the participants of unusual appearance, such as unusual vehicles (construction equipment or cars with caravans, etc.), people wearing fully loose clothes or costumes, people pushing objects (baby carriage, shopping cart, etc.) or small participants (kids, pets, etc.). One advantage of these approaches is that they usually provide more accurate velocity estimation of the objects (previous known objects or trained models).

Since *DATMO* systems are used for obstacle avoidance or as *ADAS*, both of them require a very quick response. Taking into account the high dimension of the data provided from the sensors (*LIDAR*, stereo cameras, etc.), it is necessary to reduce this dimension while system performance is not endangered. In this way, the *Grid-Based Methods* address this issue.

Grid-Based Methods allow to estimate each velocity cell, instead of having to model the dynamic environment. However, the cell size has to be small enough to be able to detect slow velocities, because the velocities are computed when an object changes from one cell to neighbour one, increasing the computational cost. Some approaches, such as, *BOFUM* and *BOFUG,* accelerate the convergence of the *BOF*, but the required maps are not always available. Taking into account the sensor observation history, the system is more robust in changing environments. Then, temporary objects, occlusions and detection problems can be handled. The case that represents the environment of a moving subject (*UGV*, car, etc.) has to deal with the relative velocity between the objects and the subject. This implies to maintain a distribution in the velocities at least twice wider than the maximal motion speed. Thus, the size of the data structure can be huge, limiting the applications of these approaches. These methods are especially useful to fuse measurements from different sensors. Also they are useful for identifying wide variety of objects with different shapes or appearance without previous knowledge of them.

Some approaches of both cases (*Objects and Grid base*d) only take into account the edge of the monitored area to create the new objects to track. This is dangerous in indoor environments, because the moving objects can be appeared suddenly behind doors or other static objects that are not in the edges of the local map.

Acknowledgements

This work has been funded by the Ministerio de Economía y Competitividad under TIN2014-56633-C3-3-R (ABS4SOW project) and by the Cátedra de Ingeniería Avanzada Escribano of the UAH under Catedra 2017-005.

References

1. T.-D. Vu and O. Aycard. “Laser-based Detection and Tracking Moving Objects Using Data-driven Markov Chain Monte Carlo,” **In:** *ICRA* (IEEE, 2009) pp. 3800–3806.
2. F. Nashashibi and A. Bargeton. “Laser-based Vehicles Tracking and Classification Using Occlusion Reasoning and Confidence Estimation,” **In:** *2008 IEEE Intelligent Vehicles Symposium* (June 2008) pp. 847–852.

3. D. Reid. "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control* **24**(6), 843–854 (1979).
4. R. W. Sittler. "An optimal data association problem in surveillance theory," *IEEE Transactions on Military Electronics* **8(2)**, 125–139 (1964).
5. B.-N. Vo and W.-K. Ma. "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing* **54**(11), 4091–4104 (2006).
6. B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne III, R. Mahler and B.-T. Vo, *Multitarget Tracking* (American Cancer Society, 2015) pp. 1–15.
7. S. S. Blackman, R. J. Dempster, M. T. Busch and R. F. Popoli, "Imm/mht solution to radar benchmark tracking problem," *IEEE Transactions on Aerospace and Electronic Systems* **35**(2), 730–738 (1999).
8. I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(2), 138–150 (1996).
9. T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering* **8**(3), 173–184 (1983).
10. I. J. Cox, "A review of statistical data association techniques for motion correspondence," *International Journal of Computer Vision* **10**(1), 53–66 (1993).
11. R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering (ASME)* **82**(1), 35–45 (1960).
12. M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt and S. Thrun, *Junior: The Stanford Entry in the Urban Challenge* (Springer, Berlin, Heidelberg, 2009) pp. 91–123.
13. X. Chen, X. Wang and J. Xuan, Tracking multiple moving objects using unscented Kalman filtering techniques. *CoRR*, abs/1802.01235 (2018).
14. A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing* **10**(3), 197–208 (2000).
15. H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control* **33**(8), 780–783 (1988).
16. C. Andrieu, N. de Freitas, A. Doucet and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning* **50**(1), 5–43 (2003).
17. K. O. Arras, S. Grzonka, M. Luber and W. Burgard, "Efficient People Tracking in Laser Range Data Using a Multi-hypothesis Leg-tracker with Adaptive Occlusion Probabilities," **In:** *ICRA* (IEEE, 2008) pp. 1710–1715.
18. D. Schulz, W. Burgard, D. Fox and A. B. Cremers, "Tracking Multiple Moving Targets with a Mobile Robot Using Particle Filters and Statistical Data Association," **In:** *Proceedings of the 2001 IEEE International Conference on Robotics and Automation, ICRA 2001,* Seoul, Korea (2001) pp. 1665–1670.
19. E. A. Topp and H. I. Christensen, "Tracking for Following and Passing Persons," **In:** *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)* , Edmonton, Alberta, Canada (2005) pp. 70–76.
20. C.-C. Wang, D. Duggins, J. Gowdy, J. Kozar, R. MacLachlan, C. Mertz, A. Suppe and C. Thorpe, *Navlab Slammot Datasets* (Carnegie Mellon University, 2004).
21. L. Zhao and C. Thorpe, "Qualitative and Quantitative Car Tracking from a Range Image Sequence," **In:** *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 1998) pp. 496–501.
22. K. Granström, *Extended target tracking using PHD filters*, PhD thesis, Linköping University (2012).
23. R. O. Chavez-Garcia and O. Aycard, "Multiple sensor fusion and classification for moving object detection and tracking," *IEEE Transactions on Intelligent Transportation Systems* **17**(2), 252–534 (2015).
24. J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.* **28**(2), 337–407 (2000).
25. J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy and J. Williams, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics* **25**(10), 727–774 (2008).
26. C. Mertz, L. E. Navarro-Serment, D. Duggins, J. Gowdy, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppe, C. Urmson, N. Vandapel, M. Hebert and C. Thorpe, "Moving object detection with laser scanners," *Journal of Field Robotics* **30**(1), 17–43 (2013).
27. C.-C. Wang, C. Thorpe and S. Thrun, "Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas," **In:** *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),* Taipei, Taiwan (2003).
28. L. Montesano, J. Minguez and L. Montano, "Modeling the Static and the Dynamic Parts of the Environment to Improve Sensor-based Navigation," **In:** *IEEE International Conference on Robotics and Automation (ICRA),* Barcelona, Spain (2005).
29. T. Miyasaka, Y. Ohama and Y. Ninomiya, "Ego-motion Estimation and Moving Object Tracking using Multi-layer LIDAR," **In:** *Proceedings of the IEEE Intelligent Vehicles Symposium* (2009) pp. 151–156.

30. T.-D. Vu, J. Burlet and O. Aycard, "Grid-based localization and local mapping with moving object detection and tracking," *Inf. Fusion* **12**(1), 58–69 (2011).
31. A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer* **22**(6), 46–57 (1989).
32. R. Biswas, B. Limketkai, S. Sanner and S. Thrun, "Towards Object Mapping in Non-stationary Environments with Mobile Robots," **In:** *2002 IEEE/RSJ International Conference* (2002) pp. 1014–1019.
33. S.-W. Yang and C.-C. Wang, "Simultaneous egomotion estimation, segmentation, and moving object detection," *J. Field Robotics* **28**(4), 565–588 (2011).
34. D. Hähnel, R. Triebel, W. Burgard and S. Thrun, "Map Building with Mobile Robots in Dynamic Environments," **In:** *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE Computer Society Press, 2003).
35. D. Hähnel, D. Schulz and W. Burgard, "Mobile robot mapping in populated environments," *Advanced Robotics* **17**(7), 579–597 (2003).
36. G. D. Tipaldi and F. Ramos, "Motion Clustering and Estimation with Conditional Random Fields," **In:** *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (October 2009) pp. 872–877.
37. J. van de Ven, F. Ramos and G. D. Tipaldi, "An Integrated Probabilistic Model for Scan-matching, Moving Object Detection and Motion Estimation," **In:** *ICRA*(IEEE, 2010) pp. 887–894.
38. D. Z. Wang, I. Posner and P. Newman, "Model-free detection and tracking of dynamic objects with 2D LIDAR," *Int. J. Rob. Res.* **34**(7), 1039–1063 (2015).
39. C. Coué, C. Pradalier, C. Laugier, T. Fraichard and P. Bessiere, "Bayesian occupancy filtering for multi-target tracking: an automotive Application," *Int. Journal of Robotics Research* **25**(1), 19–30 (2006). voir basilic: http://emotion.inrialpes.fr/bibemotion/2006/CPLFB06/.
40. A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Ann. Math. Statist.* **38**(2), 325–339 (1967).
41. M. Kurdej, J. Moras, V. Cherfaoui and P. Bonnifait, *Map-Aided Fusion Using Evidential Grids for Mobile Perception in Urban Environment* (Springer, Berlin, Heidelberg, 2012) pp. 343–350.
42. J. Moras, V. Cherfaoui and P. Bonnifait, "Evidential Grids Information Management in Dynamic Environments," **In:** *17th International Conference on Information Fusion (FUSION)* (July 2014) pp. 1–7.
43. M. Saval-Calvo, L. Medina-Valds, J. M. Castillo-Secilla, S. Cuenca-Asensi, A. Martnez and J. Villagr, "A review of the Bayesian occupancy filter," *Sensors* **17**(2) (2017).
44. C. Fulgenzi, *Autonomous navigation in dynamic uncertain environment using probabilistic models of perception and collision risk prediction.* Thesis, Institut National Polytechnique de Grenoble - INPG (June 2009).
45. M. K. Tay, K. Mekhnacha, M. Yguel, C. Coué, C. Pradalier, C. Laugier and T. Fraichard, *The Bayesian Occupation Filter* (Springer, Berlin, Heidelberg, 2008) pp. 77–98.
46. M. K. Tay, K. Mekhnacha, C. Chen, M. Yguel and C. Laugier, "An efficient formulation of the Bayesian occupation filter for target tracking in dynamic environments," *International Journal of Vehicle Autonomous Systems* **6**(1-2), 155–171 (2008).
47. A. Nègre, L. Rummelhard and C. Laugier, "Hybrid Sampling Bayesian Occupancy Filter," **In:** *2014 IEEE Intelligent Vehicles Symposium Proceedings* (2014) pp. 1307–1312.
48. J. D. Adarve, M. Perrollaz, A. Makris and C. Laugier, "Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools," *IEEE International Conference on Robotics and Automation*, St Paul, Minnesota, USA (2012).
49. Q. Baig, M. Perrollaz and C. Laugier, "A robust motion detection technique for dynamic environment monitoring: A framework for grid-based monitoring of the dynamic environment," *IEEE Robotics Automation Magazine* **21**(1), 40–48 (2014).
50. C. Chen, C. Tay, C. Laugier and K. Mekhnacha, "Dynamic Environment Modeling with Gridmap: A Multiple-object Tracking Application," **In:** *2006 9th International Conference on Control, Automation, Robotics and Vision* (2006) pp. 1–6.
51. M. Yguel, C. Tay, K. Mekhnacha and C. Laugier, *Velocity Estimation on the Bayesian Occupancy Filter for Multi-Target Tracking*, Research Report RR-5836, INRIA (2006).
52. T. Gindele, S. Brechtel, J. Schroder and R. Dillmann, "Bayesian Occupancy Grid Filter for Dynamic Environments Using Prior Map Knowledge," **In:** *2009 IEEE Intelligent Vehicles Symposium* (2009) pp. 669–676.
53. S. Brechtel, T. Gindele and R. Dillmann, "Recursive Importance Sampling for Efficient Grid-based Occupancy Filtering in Dynamic Environments," **In:** *2010 IEEE International Conference on Robotics and Automation* (2010) pp. 3932–3938.
54. R. Danescu, F. Oniga and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Transactions on Intelligent Transportation Systems* **12**(4), 1331–1342 (2011).
55. D. Nuss, B. Wilking, J. Wiest, H. Deusch, S. Reuter and K. Dietmayer, "Decision-free True Positive Estimation with Grid Maps for Multi-object Tracking," **In:** *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)* (2013) pp. 28–34.
56. K. Mekhnacha, Y. Mao, D. Raulo and C. Laugier, "Bayesian Occupancy Filter Based 'Fast Clustering-Tracking' Algorithm," **In:** *IROS 2008,* Nice, France (2008).
57. T. Yuan, D. S. Nuss, G. Krehl, M. Maile and A. Gern, "Fundamental Properties of Dynamic Occupancy Grid Systems for Vehicle Environment Perception," **In:** *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (2015) pp. 153–156.