

Article

Advancing ADAS Perception: A Sensor-Parameterized Implementation of the GM-PHD Filter

Christian Bader ^{1,2,*}  and Volker Schwieger ¹ 

¹ Institute of Engineering Geodesy, University of Stuttgart, Geschwister-Scholl-Str. 24D, 70174 Stuttgart, Germany; volker.schwieger@iigs.uni-stuttgart.de

² Daimler Truck AG, Fasanenweg 10, 70771 Leinfelden-Echterdingen, Germany

* Correspondence: christian.bader@daimlertruck.com

Abstract: Modern vehicles equipped with Advanced Driver Assistance Systems (ADAS) rely heavily on sensor fusion to achieve a comprehensive understanding of their surrounding environment. Traditionally, the Kalman Filter (KF) has been a popular choice for this purpose, necessitating complex data association and track management to ensure accurate results. To address errors introduced by these processes, the application of the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter is a good choice. This alternative filter implicitly handles the association and appearance/disappearance of tracks. The approach presented here allows for the replacement of KF frameworks in many applications while achieving runtimes below 1 ms on the test system. The key innovations lie in the utilization of sensor-based parameter models to implicitly handle varying Fields of View (FoV) and sensing capabilities. These models represent sensor-specific properties such as detection probability and clutter density across the state space. Additionally, we introduce a method for propagating additional track properties such as classification with the GM-PHD filter, further contributing to its versatility and applicability. The proposed GM-PHD filter approach surpasses a KF approach on the KITTI dataset and another custom dataset. The mean OSPA⁽²⁾ error could be reduced from 1.56 (KF approach) to 1.40 (GM-PHD approach), showcasing its potential in ADAS perception.

Keywords: GM-PHD filter; multi-object tracking; sensor fusion



Citation: Bader, C.; Schwieger, V. Advancing ADAS Perception: A Sensor-Parameterized Implementation of the GM-PHD Filter. *Sensors* **2024**, *24*, 2436. <https://doi.org/10.3390/s24082436>

Academic Editor: Ikhlas Abdel-Qader

Received: 17 February 2024

Revised: 2 April 2024

Accepted: 7 April 2024

Published: 11 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern vehicles are equipped with a variety of Advanced Driver Assistance Systems (ADAS) functions, such as advanced emergency braking assistance, blind spot information system, and adaptive cruise control. These functions actively support the driver and require a reliable environment perception, typically on the object track level. Current-generation vehicles with Level 2 systems fuse and track object detections from multiple sensors with different sensing domains and mounting positions to increase the overall system's Field of View (FoV), performance, and robustness.

This multi-sensor multi-object tracking problem can be solved by a variety of approaches. Approaches that estimate full trajectories by taking all measurements into account at once, such as shadowing filters [1], are able to achieve good tracking results; however, these are typically not capable of processing the data sequentially, which is required by most online applications, where at each time only the measurements of the current time-step are known. Particle filters are one type of online-capable approach that processes measurements recursively. They have particular strengths in representing the uncertainties of nonlinear systems and non-Gaussian noise, but require a large number of particles to do this accurately, especially for high-dimensional systems. Although real-time-capable implementations are possible, as shown by [2] using Rao–Blackwellization, the authors still estimated the computational speed to be slower compared to a Kalman filter by a factor of 100 with 10^3 particles and a state vector of dimension $n_x \approx 5$. Because this

paper is interested in solutions with computational requirements in the range of the Kalman filter and is less focused on highly nonlinear and non-Gaussian applications, the particle filter is not considered a viable choice.

A popular and computationally efficient solution to the multi-sensor multi-object tracking problem is the use of a Kalman Filter (KF), such as the examples in [3–8]. The states of each object are estimated by a separate KF [9], which essentially makes it an extension of a single target filtering approach. Therefore, such approaches require a data association scheme that matches new measurements to the already tracked objects and a track management scheme to handle the appearance and disappearance of tracks. Errors such as incorrect assignment during data association can occur during these two supporting steps. Such a situation can then lead to incorrect estimation of the number of objects and their states.

In contrast, the family of Random Finite Set (RFS) filters estimates both the cardinality (the number of objects) and their states. Therefore, they offer true multi-target filtering without data association along with complex track management. The Probability Hypothesis Density (PHD) filter [10] estimates the probability hypothesis density (or intensity function) of a random set variable. The Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter [11] is an efficient Gaussian implementation of the PHD filter. This paper shows an approach that makes this filter an alternative to KF with a performance advantage that is computationally lightweight and suitable for real-time applications. A key aspect of the presented approach is that it is designed in such a way that a Kalman filter-based approach can be replaced without much effort. Therefore, it is applicable to systems with partially overlapping or non-overlapping sensor FoVs and asynchronous sensor updates, which are common in real-world setups currently covered by KF approaches.

The standard GM-PHD filter assumes the use of a single sensor. Next to the single sensor PHD filter, Mahler introduced the inter-corrector PHD [10] as a multi-sensor solution. This was achieved using multiple single-sensor Bayesian filters [12]. Improvements for multi-sensor applications of the PHD filter followed in [13] along with Gaussian mixture implementations and developments in [14,15]. However, these approaches assume sensor systems with completely overlapping FoVs and synchronous measurements, which is not the case in current-generation Level 2+ systems. In general, many GM-PHD-based fusion approaches use multiple single sensor filters together with a combination rule such as the Generalized Covariance Intersection (GCI) for fusion. These approaches are usually not suitable for partial/non-overlapping sensor FoVs. To overcome this shortage, ref. [16] presented a collaborative GM-PHD filter that fuses the intensity functions of multiple filter sources using Covariance Intersection on nearby Gaussian components only.

In [17], a comparison of two approaches to asynchronous radar sensor fusion with the PHD filter was presented. Sequence process PHD (SP-PHD) fusion fuses measurements of multiple sensors sequentially using a global filter, while fixed nodes PHD fusion uses separate sensor PHD filters that are fused using the GCI. This was followed by [18], where distributed PHD fusion approaches were shown next to the SP-PHD fusion for asynchronous sensor setups. However, these approaches have not been described for partially overlapping FoVs. Overall, there are only a very few publications focusing on real-world automotive applications, where asynchronous sensor setups and partially/non-overlapping sensor FoVs are common.

Most publications have assumed constant values for sensor-specific parameters such as clutter density and detection probability. However, this is not the case for real-world sensors, and can lead to tracking errors in the case of partial/non-overlapping sensor FoVs. This paper shows that the application of sensor-specific models for these parameters can solve multiple problems of GM-PHD filters in real-world applications. In [19], this was addressed by differentiating the detection probability between inside and outside the FoV [16]. In addition [20], introduced occlusion models for those areas where tracks are not detectable. Ref. [21] proposed a detection probability model for sonar applications, taking the object distance into account. This paper proposes a more versatile model for

the detection probability with distance dependency that fits the capabilities of automotive sensors. Compared to previous publications, the clutter density is modeled in a similar way.

Because many systems fuse additional track attributes such as the classification in addition to the track states, this paper presents a new approach to handle the propagation of additional attributes with the GM-PHD filter. In addition, an adaptive birth model and a gating process are proposed to increase computational efficiency.

The presented approach uses an SP-PHD scheme for asynchronous sensor measurements in a way that can replace typical KF implementations. This enables a direct performance comparison of the proposed GM-PHD approach to a KF approach on both the KITTI dataset [22] and a custom dataset with multiple radars and a camera. In the evaluation, the proposed GM-PHD approach substitutes for the KF approach, with as many parameters as possible remaining the same in order to ensure robust comparability. By testing on two real-world datasets and considering multiple sensor combinations, a statement can be made about its suitability for real systems, which has not been possible to date.

2. The GM-PHD Filter

RFS filters estimate a set of multiple-target states; as such, they cover both the state estimation and the estimation of the number of states with their birth and death. Therefore, no explicit data association between individual measurements and tracks is required, as the state set is estimated as a whole.

The PHD filter [10] approximates the exact multi-target filter by only propagating the first-order statistical moment (PHD) of an RFS. The PHD is also called the intensity function. Under the assumption of a Poisson RFS, there is a closed-form analytic solution for the prediction and update step, as both the cardinality distribution and the spatial distribution are defined by the PHD.

The GM-PHD filter uses a Gaussian mixture to represent the PHD $v_{k-1}(\mathbf{x})$ of an object state \mathbf{x} at time step $k-1$, which has the advantage of fast computation. Because the first-order statistical moment of an RFS is estimated, the PHD $v(\mathbf{x}_0)$ can be interpreted as the density of the expected number of targets at \mathbf{x}_0 . The PHD is always represented as a sum of J_{k-1} weighted Gaussians with weight $w_{k-1}^{(i)}$, mean $\mathbf{m}_{k-1}^{(i)}$, and covariance $\mathbf{P}_{k-1}^{(i)}$ [11]:

$$v_{k-1}(\mathbf{x}) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{k-1}^{(i)}, \mathbf{P}_{k-1}^{(i)}). \quad (1)$$

2.1. GM-PHD Prediction

Similar to the Kalman filter, the GM-PHD filter is divided into prediction and update. However, in practical implementations the additional computation steps prune and merge are used to keep the number of Gaussian components at a low number, as shown by [23,24]. The GM-PHD predicted intensity $v_{k|k-1}(\mathbf{x})$ is calculated by combining the prediction $v_{S,k|k-1}(\mathbf{x})$ with the spawn intensity $v_{\beta,k|k-1}(\mathbf{x})$ and birth intensity $\gamma_k(\mathbf{x})$ [11]:

$$v_{k|k-1}(\mathbf{x}) = v_{S,k|k-1}(\mathbf{x}) + v_{\beta,k|k-1}(\mathbf{x}) + \gamma_k. \quad (2)$$

The spawn intensity models the spawning of new targets from existing ones, while the birth intensity models the birth of new targets; both are represented as a Gaussian mixture. The developed model for the birth intensity is shown in Section 3.4.2, while the spawn intensity is not used here. The prediction $v_{S,k|k-1}(\mathbf{x})$ is calculated by

$$v_{S,k|k-1}(\mathbf{x}) = p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{S,k|k-1}^{(j)}, \mathbf{P}_{S,k|k-1}^{(j)}), \quad (3)$$

$$\mathbf{m}_{S,k|k-1}^{(j)} = \mathbf{A}_{k-1} \mathbf{m}_{k-1}^{(j)}, \quad (4)$$

$$\mathbf{P}_{S,k|k-1}^{(j)} = \mathbf{Q}_{k-1} + \mathbf{A}_{k-1} \mathbf{P}_{k-1}^{(j)} \mathbf{A}_{k-1}^T, \quad (5)$$

using the survival probability $p_{S,k}$ [11]. The means and covariances are predicted using the state transition matrix A_{k-1} and process noise covariance matrix Q_{k-1} . Similar to the extended Kalman filter, an extended Kalman PHD filter equation set is described in [11], which allows nonlinear process models.

2.2. GM-PHD Update

After the prediction step, the predicted intensity $v_{k|k-1}(\mathbf{x})$ for time k is represented by the Gaussian mixture

$$v_{k|k-1}(\mathbf{x}) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{k|k-1}^{(i)}, \mathbf{P}_{k|k-1}^{(i)}). \quad (6)$$

The GM-PHD update step then calculates the corrected intensity $v_k(\mathbf{x})$ at time k using the detection probability $p_{D,k}$ and the clutter density $\kappa_k(\mathbf{z})$. The update is calculated by iterating over all measurements \mathbf{z} of the measurement set Z_k [11]:

$$v_k(\mathbf{x}) = (1 - p_{D,k})v_{k|k-1}(\mathbf{x}) + \sum_{\mathbf{z} \in Z_k} v_{D,k}(\mathbf{x}; \mathbf{z}) \quad (7)$$

with

$$v_{D,k}(\mathbf{x}; \mathbf{z}) = \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(\mathbf{z}) \mathcal{N}(\mathbf{x}; \mathbf{m}_{k|k}^{(j)}(\mathbf{z}), \mathbf{P}_{k|k}^{(j)}), \quad (8)$$

$$w_k^{(j)}(\mathbf{z}) = \frac{p_{D,k} w_{k|k-1}^{(j)} q_k^{(j)}(\mathbf{z})}{\kappa_k(\mathbf{z}) + p_{D,k} \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(\mathbf{z})}, \quad (9)$$

$$q_k^{(j)}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{H}_k \mathbf{m}_{k|k-1}^{(j)}, \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1}^{(j)} \mathbf{H}_k^T), \quad (10)$$

$$\mathbf{m}_{k|k}^{(j)}(\mathbf{z}) = \mathbf{m}_{k|k-1}^{(j)} + \mathbf{K}_k^{(j)} (\mathbf{z} - \mathbf{H}_k \mathbf{m}_{k|k-1}^{(j)}), \quad (11)$$

$$\mathbf{P}_{k|k}^{(j)} = [\mathbf{I} - \mathbf{K}_k^{(j)} \mathbf{H}_k] \mathbf{P}_{k|k-1}^{(j)}, \quad (12)$$

$$\mathbf{K}_k^{(j)} = \mathbf{P}_{k|k-1}^{(j)} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1}^{(j)} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \quad (13)$$

Here, $p_{D,k}$ is the detection probability and $\kappa_k(\mathbf{z})$ is the clutter density, while updating of the Gaussian components is performed using the observation matrix \mathbf{H}_k and the observation noise covariance matrix \mathbf{R}_k .

The equations presented here are the basis of the proposed tracking approach. However, various optimizations and adaptations are proposed around them in order to create an implementation that allows a fast and reliable application in real-world scenarios. These can be found in the following sections.

3. Proposed Multi-Sensor Multi-Object Tracking Approach

Multi-sensor multi-object tracking is considered here as estimating the number of all surrounding objects as well as their states over time while assigning track attributes, a confidence rating, and unique IDs to form trajectories. Therefore, the i -th tracked object $\mathbf{T}_k^{(i)}$ of time step k is represented by a tuple $\mathbf{T}_k^{(i)} = (\tau_k^{(i)}, \mathbf{x}_k^{(i)}, \mathbf{c}_k^{(i)}, p_k^{(i)}(\exists))$ including the track's unique ID $\tau_k^{(i)}$, state $\mathbf{x}_k^{(i)}$, classification $\mathbf{c}_k^{(i)}$, and probability of existence $p_k^{(i)}(\exists)$. The classification $\mathbf{c}_k^{(i)}$ is represented as a probability distribution or pseudo-probability distribution over all possible classes, which is necessary for fusion and estimation over time. The set of all n tracked objects is $\mathbf{T}_k = \{\mathbf{T}_k^{(1)}, \mathbf{T}_k^{(2)}, \dots, \mathbf{T}_k^{(n)}\}$.

Multiple independent sensors that may have different FoVs update the tracks with measurements \mathbf{z}_k and classification measurements $\mathbf{c}_{z,k}$. The goal is to estimate the states and

attributes of all existing tracks using the independent measurements provided by different sensors. Smart sensors with integrated object tracking capabilities might be used in real-world applications by treating the tracks of each time step as individual measurements.

3.1. Kalman Filter Approach

A Kalman filter-based approach is implemented here for comparison, as many current systems use this approach. In Kalman filter pipelines such as [3–6,8], a separate filter estimates the states of each tracked object. This is supported by a gating and data association approach that maps the incoming measurements of a sensor to the existing tracks. In this implementation, the chi-square test following ([25], p. 112) for gating and a global nearest neighbor data association scheme [26] based on the Mahalanobis distance is used. The distances between detections and tracks are calculated using the Mahalanobis distance D_M with the predicted measurement $\hat{z}_{k|k-1}$ and the innovation covariance matrix S_k ,

$$D_M(z) = \sqrt{[z_k - \hat{z}_{k|k-1}]^T S_k^{-1} [z_k - \hat{z}_{k|k-1}]} \quad (14)$$

and the association is then solved using the Hungarian algorithm [27]. A track management scheme is responsible for adding new appearing tracks and removing disappearing tracks accordingly. In this work, the Bayesian formulation for existence probability estimation from [28] is used as a basis for the track management. Non-updated tracks are removed if the existence probability is below a certain threshold, while tracks are forwarded if the existence probability is above a minimum threshold.

The state vector of each track contains the two-dimensional position p_x and p_y , speed v_x and v_y , and acceleration a_x and a_y , as well as the Oriented Bounding Box (OBB) orientation ϕ and OBB dimensions, consisting of length l , width w , and height h . The state x_k vector is described as follows:

$$x_k = [p_x \ p_y \ v_x \ v_y \ a_x \ a_y \ l \ w \ h \ \phi]^T. \quad (15)$$

Similarly, the measurement vector z_k consists of the position p_x and p_y as well as the dimensions l , w , and h and orientation ϕ , resulting in $z_k = [p_x \ p_y \ l \ w \ h \ \phi]^T$. The system dynamics are described by a constant acceleration model with transition function f and time difference Δt , as the last time step $k - 1$ is defined by

$$x_{k|k-1} = f(x_{k-1|k-1}, u_{k-1}) = \begin{bmatrix} p_x + \Delta t v_x + \frac{1}{2} \Delta t^2 a_x + \Delta p_{x,u} \\ p_y + \Delta t v_y + \frac{1}{2} \Delta t^2 a_y + \Delta p_{y,u} \\ v_x + \Delta t a_x \\ v_y + \Delta t a_y \\ a_x \\ a_y \\ l \\ w \\ h \\ \phi + \Delta \phi_u \end{bmatrix} \quad (16)$$

with the control vector $u_{k-1} = [v_{ego} \ \omega_{ego}]^T$ including the ego velocity v_{ego} and the ego yaw rate ω_{ego} , which results in

$$\Delta p_{x,u} = -v_{ego} \Delta t + \sin(\omega_{ego} \Delta t) \omega_{ego} \Delta t p_x + \cos(\omega_{ego} \Delta t) \omega_{ego} \Delta t p_y, \quad (17)$$

$$\Delta p_{y,u} = -\cos(\omega_{ego} \Delta t) \omega_{ego} \Delta t p_x + \sin \omega_{ego} \Delta t \omega_{ego} \Delta t p_y, \quad (18)$$

$$\Delta \phi_u = -\omega_{ego} \Delta t. \quad (19)$$

The extended Kalman filter equations are used for the prediction.

3.2. Substituting the KF with the GM-PHD Filter

In order to replace a KF-based Multi-Object Tracking (MOT) framework with a GM-PHD-based one, the interfaces must be similar. The input of a KF framework is a list of detections, each containing a measurement vector z_k and attributes, such as the classification c_z . Figure 1 shows an overview of the Kalman filter framework on the left-hand side and substitution by a GM-PHD framework on the right-hand side. With the KF, the states are estimated separately for each track. This means that the additional attributes, such as the classification, can also be calculated separately for each track. In contrast, the GM-PHD filter estimates the global PHD. The tracks are then extracted from the PHD in a separate step. Section 3.4 shows an approach to propagating additional attributes such as the classification alongside the PHD to provide this additional information for the extracted tracks.

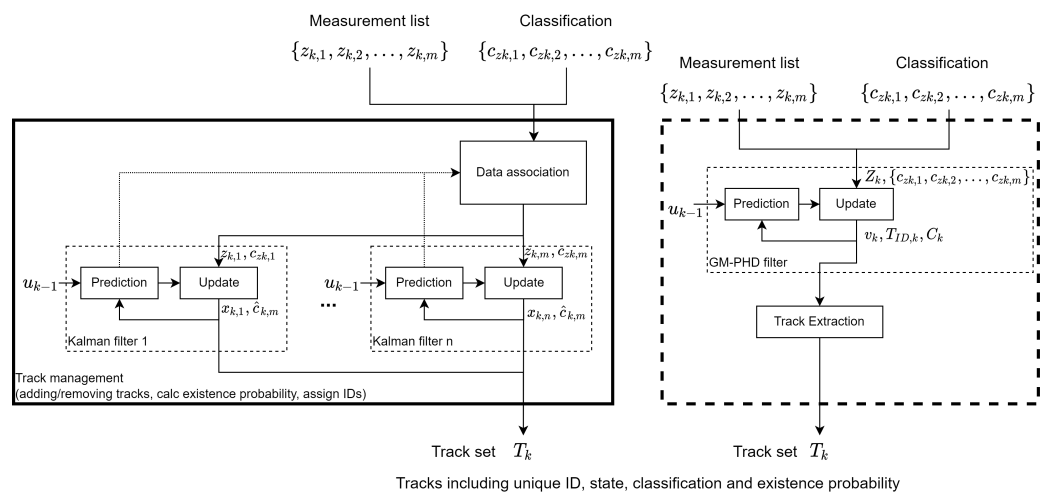


Figure 1. Comparison of KF and GM-PHD framework for automotive MOT application.

3.3. Sensor-Based Parameter Models

3.3.1. Detection Probability

The original implementation of the GM-PHD filter [11] assumes a constant detection probability over the entire state space. However, this assumption is not the case with real sensor systems. For example, if a track with state x_1 is located outside the FoV of the sensor s_1 , as shown in Figure 2, then the detection probability is 0; depending on the sensor technology, the detection probability will also change with environmental influences such as the weather or illumination conditions. For almost all sensors, there is a decrease in capability with increasing distance, which can be modeled mathematically.

The detection probability model is an important part of proper sensor fusion in the proposed framework; it strongly influences the appearance and disappearance, as it highly affects the weight calculation of the GM-PHD. In this framework, if a Gaussian component has a high detection probability and no corresponding measurement exists, then the weight will decrease; in turn, if a component has a very low detection probability and no corresponding measurement exists, then the weight is barely effected. Note that the detection probability model can also be applied to the existence probability estimation used for the KF framework.

We propose a detection probability model based on the FoV and the track's distance. Therefore, a mathematical model $p_{Dm,k}^{(s_p)}(x)$ for the detection probability of track x is defined based on experimental measurements for each sensor s_p . Next to the FoV, which obviously sets limits to the detection probability, the distance is a key parameter. Due to the decreasing resolution at higher distances, the detection probability will decrease with the distance for all types of perception sensors. In addition, certain sensors, such as automotive radars, can have direction-dependent sensitivities and maximum detection ranges, which can result in

more complex detection probability models. To ensure general applicability to different types of sensors, the proposed model is based only on the distance and a radial FoV.

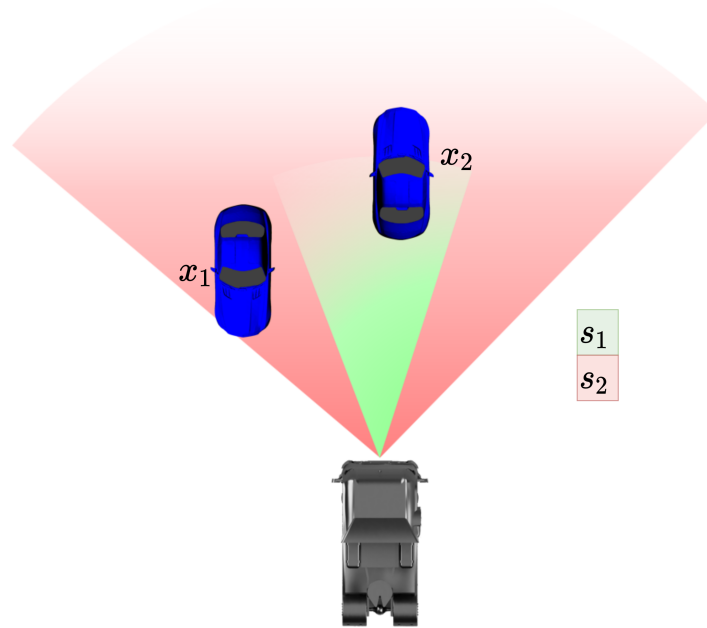


Figure 2. Example of two sensors s_1 and s_2 covering different FoVs and the states x_1 and x_2 of two tracked objects.

Overall, the formula for the detection probability model $p_{Dm,k}^{(s_p)}(\mathbf{x})$ is as follows:

$$\begin{aligned} p_{Dm,k}^{(s_p)}(\mathbf{x}) &= p_{dist}^{(s_p)}(\mathbf{x}) p_{FoV}^{(s_p)}(\mathbf{x}), \\ p_{dist}^{(s_p)}(\mathbf{x}) &= k_2^{(s_p)} d(\mathbf{x})^2 + k_1^{(s_p)} d(\mathbf{x}) + k_0^{(s_p)}, \\ p_{FoV}^{(s_p)}(\mathbf{x}) &= \begin{cases} 0, & \mathbf{x} \text{ out FoV} \\ 1, & \mathbf{x} \text{ in FoV} \end{cases}. \end{aligned} \quad (20)$$

In this description, $p_{dist}^{(s_p)}(\mathbf{x})$ is the second-order polynomial describing the distance dependency, with $d(\mathbf{x}) = \sqrt{p_x^2 + p_y^2}$ being the distance of the track's OBB center.

To verify this model and calculate the parameter values, the detection probabilities were experimentally determined for the “Car” class. Using the Point-RCNN [29] detector for Lidar and Yolo-Mono-3D [30] for monocular camera, the detection probability was calculated using the training data from the KITTI tracking dataset. For each time step, each of the ground truth tracks was compared with the respective measurements. If a measurement with an overlapping OBB was found for a ground truth track, it was counted as a true detection. Next, the percentage of detected tracks within the sensor FoV was calculated depending on the distance in order to obtain the detection probabilities.

As shown on the top of Figure 3, the second-order polynomial used by the model can accurately represent the distance dependent detection probability for both the camera and the Lidar detector. The Bird’s Eye View (BEV) result of this polynomial model in combination with the sensor’s FoV is shown at the bottom of Figure 3 and compared to the measured result.

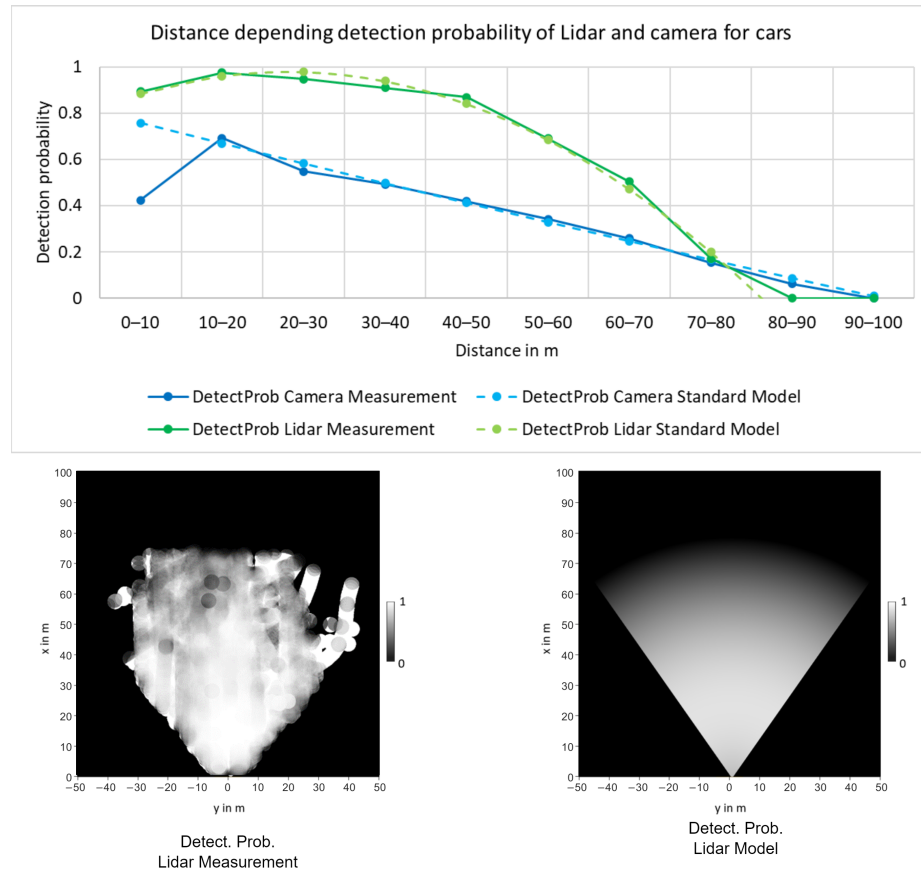


Figure 3. Detection probability distance model for “Car” class. On the top, the detection probability models of the camera and Lidar with respect to the distance; on the bottom, the detection probability measurements and model in BEV.

The GM-PHD implementation from [11] assumes the detection probability to be constant, but shows a closed-form update for a state-dependent detection probability provided as a mixture. However, this form results in numerous Gaussian components, which greatly increases the computational effort, as shown in Section 4.4. To reduce this, the constant assumption is violated and

$$p_{D,k} = p_{Dm,k}^{(s_p)}(\mathbf{m}_{k|k-1}^{(j)}) \quad (21)$$

is used instead. The experiments in Section 4.3 show that the proposed solution still works for real-world data. In [16], state-dependent detection probability was similarly used without a mixture model. However, their emphasis was on FoV and occlusion rather than on presenting a comprehensive model that encompasses distance dependency. Likewise, ref. [19] differentiated the detection probability depending on the object being inside or outside the FoV, and did not consider the distance dependency.

3.3.2. Clutter Density

Similar to the detection probability, the clutter density is described as a mathematical model. In contrast to the detection probability, the GM-PHD standard filter equations do not include the assumption of a static value. Therefore, a mathematical model for the clutter density can be used directly.

The clutter density can be modeled similar to the detection probability. Therefore, the mathematical model for $\kappa_k^{(s_p)}(\mathbf{z})$ of sensor s_p is experimentally defined based on the distance.

The distance dependency was experimentally determined using the Point-RCNN [29] detector for Lidar and Yolo-Mono-3D [30] for monocular camera on the training data of the KITTI tracking dataset. Figure 4 shows the measured clutter density of the “Car” class for both the Lidar and the camera. As shown, the clutter density has a peak value for medium distances and appears to be lower for both near and far distances. This behavior can be modeled using a sinusoidal function, which leads to the approximations shown in Figure 4.

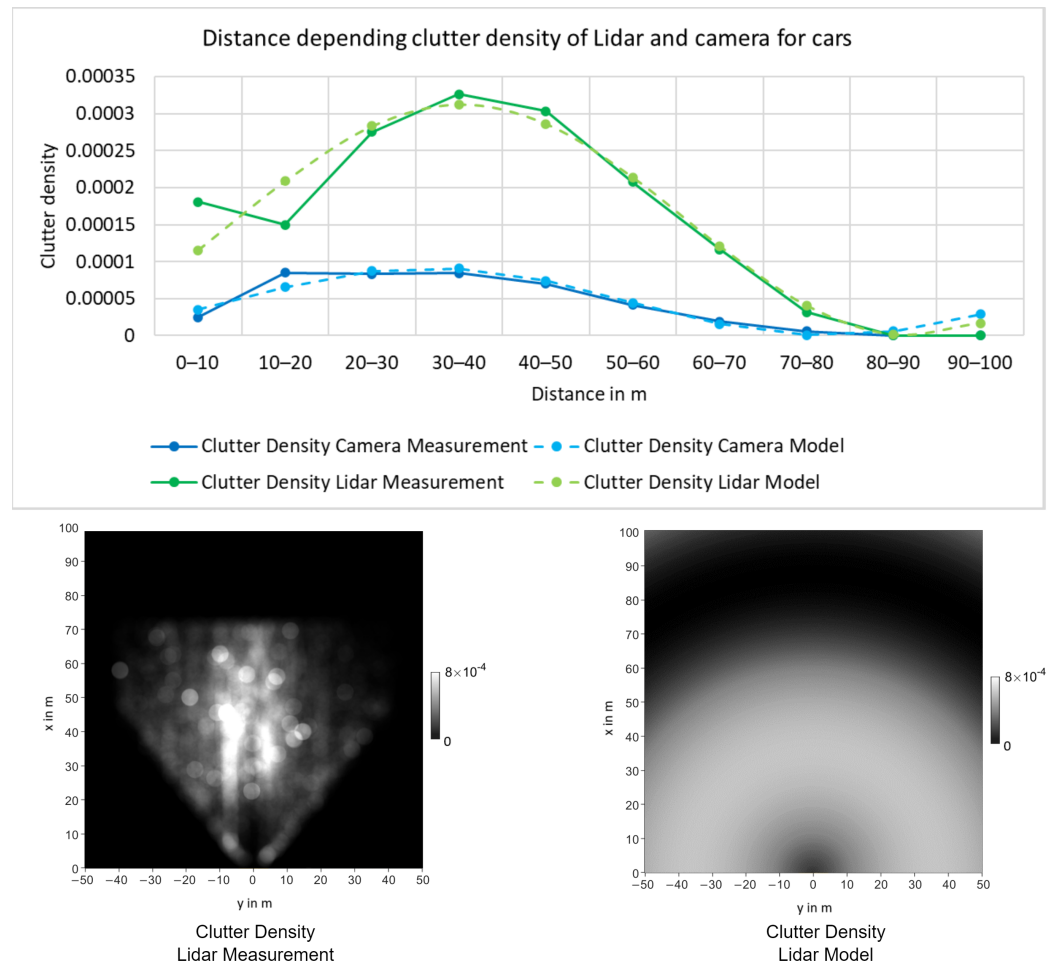


Figure 4. Clutter density distance model for "Car" class. On the top, clutter density models of the camera and Lidar with respect to the distance; on the bottom, the Lidar clutter density measurement and model in BEV.

The overall mathematical description for the clutter density $\kappa_k^{(sp)}(z)$ of measurement z is described by

$$\kappa_k^{(sp)}(z) = k_0 \sin(k_1 d(z) + k_2) + k_0. \quad (22)$$

3.4. Computational Steps for the GM-PHD Approach

This section describes the processing steps of the approach using the GM-PHD filter. These steps use the GM-PHD filter equations, and additionally propagate the tags for the unique ID and the classification over time. As described by [23], a unique identifier, or tag, $\tau_{k|k}$, is assigned to each Gaussian component at time step k , which together form a set

$$\tau_{k|k} = \left\{ \tau_{k|k}^{(1)}, \dots, \tau_{k|k}^{(J_{k|k})} \right\}, \quad (23)$$

where $J_{k|k}$ is the number of Gaussian components at time step k and τ_{max} is the currently highest unique ID. To additionally track the classification of each Gaussian component, the respective classification set $C_{k|k}$ with

$$C_{k|k} = \left\{ \mathbf{c}_{k|k}^{(1)}, \dots, \mathbf{c}_{k|k}^{(J_{k|k})} \right\} \quad (24)$$

is introduced.

3.4.1. Step 0: Initialization

For initialization, the number of Gaussian components is set to $J_0 = 0$ and the tags and classification sets are empty, with $\tau_0 = \emptyset$ and $C_0 = \emptyset$. After the one-time initialization, steps 1–5 are repeated.

3.4.2. Step 1: Prediction

The prediction step predicts the PHD at the current time step using the constant acceleration model from Section 3.1 and the extended Kalman PHD equations from [11]. In addition, the survival probability $p_{S,k}$ is calculated for each time step using a base survival probability $p_{S,base}$ and the time difference from the last time step Δt , as follows:

$$p_{S,k} = p_{S,base}^{\Delta t} \quad (25)$$

The proposed implementation does not model the spawning process of Equation (2), which ignores certain scenarios for simplicity and sets $v_{\beta,k|k-1}(\mathbf{x}) = 0$. New objects can appear anywhere in the monitored space; however, as the computational effort should be kept low, we want to avoid creating a high number of random Gaussian component. Thus, adaptive birth models depending on the measurements are considered here, and a new adaptive approach is presented. In [31,32], the authors proposed adaptive birth processes which create new Gaussian components based on all measurements. However, this inevitably creates an overlap between birth components and already-tracked components in many areas. These overlaps do not truly represent the birth of new targets. Instead, Gaussian components should only be added in areas where no components are currently present. In this way, the influence of the newly added components on the update of the existing components is reduced, and both the total number of Gaussian components and the computational effort can be reduced while ensuring that the birth of new targets remains well-modeled.

In the presented approach, a birth measurement set $Z_{\gamma,k-1}$ is generated in the previous update step $k-1$ that includes all measurements with a sum of generated weights $w_{sum,k-1}^{(j)}$ below a threshold t_{w_γ} :

$$Z_{\gamma,k-1} = \left\{ \mathbf{z}_{\gamma,k-1}^{(j)} \in Z_{k-1} \forall w_{sum,k-1}^{(j)} < t_{w_\gamma} \right\}. \quad (26)$$

The threshold was found experimentally and set to $t_{w_\gamma} = 0.01$. For the sum of weights, the measurement weights $q_{k-1}^{(l)}(\mathbf{z})$ from Equation (10) are used combined with the predicted weights $w_{k-1|k-2}$ from the previous time step:

$$w_{sum,k-1}^{(j)} = \sum_{l=1}^{J_{k-1|k-2}} w_{k-1|k-2}^{(l)} q_{k-1}^{(l)}(\mathbf{z}). \quad (27)$$

The sum of weights $w_{sum,k-1}^{(j)}$ indicates the influence of the j -th measurement on the PHD update. If the influence is very low, it is assumed that no Gaussian component of the intensity function is in close range to the measurement. Therefore, $Z_{\gamma,k-1}$ is a subset of the measurement set Z_{k-1} that only contains measurements in areas where no Gaussian components are present in the intensity function. Next, a Gaussian mixture $\hat{\gamma}_{k-1}$ is generated based

on these measurements with $\mathbf{m}_{\gamma,k-1}^{(j)} = [p_x^{(j)} \ p_y^{(j)} \ 0 \ 0 \ 0 \ 0 \ l^{(j)} \ w^{(j)} \ h^{(j)} \ \phi^{(j)}]^T$, $\mathbf{P}_{\gamma,k-1}^{(j)} = \mathbf{P}_0$ with initial covariance \mathbf{P}_0 and $w_{\gamma,k-1}^{(j)} = w_0$ and with initial weight w_0 . Then, $\hat{\gamma}_{k-1}$ is predicted to time step k using the GM-PHD prediction to form γ_k .

For each of the birth Gaussian components, a new birth tag is added along with the set of birth tags at the current time step, as proposed by [23]. Each tag is initialized with a new unique ID based on τ_{max} , which is increased by J_{γ_k} afterwards:

$$\tau_{k|k-1} = \tau_{k-1} \cup \left\{ \tau_{\gamma_k}^{(1)}, \dots, \tau_{\gamma_k}^{(J_{\gamma_k})} \right\}, \quad (28)$$

$$\tau_{\gamma_k}^{(j)} = \tau_{max} + j. \quad (29)$$

Similarly, a classification probability vector is added for each Gaussian component. The initial classification is calculated by updating a uniform classification distribution \mathbf{c}_U with the measurement classification probability vector $\mathbf{c}_{z_{\gamma,k-1}}^{(j)}$. In principle, this can be done with any classification update function $f_c(\mathbf{c}, \mathbf{c}_z)$ capable of updating the classification in a meaningful way; however, this is not further described here. The classification set is accordingly extended with

$$\mathbf{C}_{k|k-1} = \mathbf{C}_{k-1} \cup \left\{ \mathbf{c}_{k|k-1}^{(1)}, \dots, \mathbf{c}_{k|k-1}^{(J_{\gamma_k})} \right\}, \quad (30)$$

$$\mathbf{c}_{k|k-1}^{(j)} = f_c(\mathbf{c}_U, \mathbf{c}_{z_{\gamma,k-1}}^{(j)}). \quad (31)$$

3.4.3. Step 2: Update

The update step follows the overall procedure from [23], with adaptations for improved tracking results and computation speed in ADAS applications.

A gating step is introduced to improve the computation speed. The GM-PHD update step adds many Gaussian components due to the update of each prior Gaussian component with each measurement component. Many of these components have very low weights and would be pruned in the next step anyway. To reduce this number of low-weight components in advance, a gating process is introduced. Because gating in ADAS applications can be interpreted spatially, the gating is based solely on the position in terms of the X and Y coordinates. The minimum of the squared Mahalanobis distance $D_{M,pos}$ and the Euclidean distance $D_{Euclid,pos}$ defining the gate are calculated using the position difference $\mathbf{z}_{p,zm}^{(i)}$ and the position covariance $\mathbf{P}_p^{(i)}$, which only include the elements related to the X–Y position:

$$\mathbf{z}_{p,zm}^{(i)} = \left[(z_1 - m_{k|k-1,1}^{(i)}) \quad (z_2 - m_{k|k-1,2}^{(i)}) \right]^T, \quad (32)$$

$$\mathbf{P}_p^{(i)} = \begin{bmatrix} P_{k|k-1,11}^{(i)} & P_{k|k-1,12}^{(i)} \\ P_{k|k-1,21}^{(i)} & P_{k|k-1,22}^{(i)} \end{bmatrix}. \quad (33)$$

The distances are then calculated with

$$D_{M,pos}(\mathbf{z}_{p,zm}^{(i)}, \mathbf{P}_p^{(i)}) = \left[\mathbf{z}_{p,zm}^{(i)} \right]^T \left[\mathbf{P}_p^{(i)} \right]^{-1} \left[\mathbf{z}_{p,zm}^{(i)} \right], \quad (34)$$

$$D_{Euclid,pos}(\mathbf{z}_{p,zm}^{(i)}) = \sqrt{\left[\mathbf{z}_{p,zm}^{(i)} \right] \left[\mathbf{z}_{p,zm}^{(i)} \right]^T}. \quad (35)$$

Applying the gating based on these distances, Equation (8) changes to

$$v_{D,k}(\mathbf{x}; \mathbf{z}) = \sum_{j \in V_k(\mathbf{z}, \gamma_{DA})} w_k^{(j)}(\mathbf{z}) \mathcal{N}(\mathbf{x}; \mathbf{m}_{k|k}^{(j)}(\mathbf{z}), \mathbf{P}_{k|k}^{(j)}), \quad (36)$$

where only components of the gating indices $V_k(\mathbf{z}, \gamma_{DA})$ are taken into account. For these gating indices, either the squared Mahalanobis distance or the Euclidean distance is below the gating threshold γ_{DA} :

$$V_k(\mathbf{z}, \gamma_{DA}) = \left\{ i = 1, \dots, J_k \mid \min(D_{M,p}(\mathbf{z}_{p,zm}^{(i)}, \mathbf{P}_p^{(i)})^2, D_{Euclid,p}(\mathbf{z}_{p,zm}^{(i)})) \leq \gamma_{DA} \right\}. \quad (37)$$

The update step uses the sensor-based parameter models proposed in Section 3.3 for the detection probability and clutter density. Therefore, the following applies and is used in the update equations:

$$p_{D,k} = p_{Dm,k}^{(sp)}(\mathbf{m}_{k|k-1}^{(j)}), \quad (38)$$

$$\kappa_k(\mathbf{z}) = \kappa_k^{(sp)}(\mathbf{z}). \quad (39)$$

Because new Gaussian components are generated during the update step, new tags and classifications are added to the tag set and classification set with m measurements, as follows:

$$\tau_{k,u} = \left\{ \tau_{k|k-1} \cup \tau_{k|k-1}^{z_1} \cup \dots \cup \tau_{k|k-1}^{z_m} \right\}, \quad (40)$$

$$\tau_{k|k-1}^z = \left\{ \tau_j \in \tau_{k|k-1} \mid j \in V_k(\mathbf{z}, \gamma_{DA}) \right\}, \quad (41)$$

$$C_{k,u} = \left\{ C_{k|k-1} \cup C_{k,u}^{z_1} \cup \dots \cup C_{k,u}^{z_m} \right\}, \quad (42)$$

$$C_{k,u}^z = \left\{ f_c(\mathbf{c}_k^{(j)}, \mathbf{c}_{z_{\gamma,k-1}}^{(j)}) \mid j \in V_k(\mathbf{z}, \gamma_{DA}) \right\}. \quad (43)$$

For each Gaussian component created by a measurement, a tag is added that has the same value as the tag of the updated component. Similarly, for each Gaussian component created by a measurement, the classification probability vector is updated by the measurements' classification probability vector $\mathbf{c}_{z_{\gamma,k-1}}^{(j)}$ using the function $f_c(\mathbf{c}, \mathbf{c}_z)$.

3.4.4. Step 3: Pruning

The pruning step from [23] is implemented without adaptations, which means that all components with low weights below a truncation threshold τ are eliminated. The classification set $C_{k,u}$ is pruned in a similar way to the tag set.

3.4.5. Step 4: Merging

The merging procedure in general follows the one proposed in [23], and has the goal of merging Gaussian components that have a low distance based on a distance criterion. In contrast to [11,23], and similar to [33], this approach uses the Kullback-Leiber Divergence (KLD) D_{KL} as the distance criterion, which is defined as follows:

$$D_{KL}(\mathcal{N}_i \parallel \mathcal{N}_j) = \frac{1}{2} \left(\text{tr}(\mathbf{P}_j^{-1} \mathbf{P}_i) - k + (\mathbf{m}_j - \mathbf{m}_i)^T \mathbf{P}_j^{-1} (\mathbf{m}_j - \mathbf{m}_i) + \ln \left(\frac{\det \mathbf{P}_j}{\det \mathbf{P}_i} \right) \right). \quad (44)$$

Here, k is the dimension of the covariance matrices \mathbf{P}_i and \mathbf{P}_j . This distance is well-suited to being used as a criterion for whether two Gaussian components should be merged, as it provides an indication of the possible information loss.

When merging multiple components, the tag $\tau_k^{(i)}$ of the component with the highest weight $w_k^{(i)}$ is kept for the merged component. If components with the same tag still exist after the merging procedure, the component with the highest weight keeps the tag and all others are assigned a new one, similar to the proposal from [23].

The classification is merged similarly to the mean. Therefore, the merged classification distribution vector \tilde{c}_k is calculated using the weighted average of all merged components L :

$$\tilde{c}_k = \frac{1}{\tilde{w}_k} \sum_{i \in L} w_k^{(i)} c_k^{(i)}. \quad (45)$$

3.4.6. Step 5: Track Extraction

In the last step, the tracks are extracted from the PHD estimation. To be interpreted as a track, the weight of a Gaussian component has to exceed a weight threshold w_{\min} creating the track set T_k :

$$T_k = \left\{ (\tau_k^{(i)}, \mathbf{m}_k^{(i)}, \mathbf{c}_k^{(i)}, \tilde{p}_k^{(i)}(\exists)) \mid w_k^{(i)} > w_{\min} \right\}. \quad (46)$$

To provide an existence probability for further processing steps, the weight of the Gaussian component is interpreted as a pseudo-existence probability with $\tilde{p}_k^{(i)}(\exists) = \min(w_k^{(i)}, 1)$.

3.5. Track Confirmation Strategy

After the track extractions, the relevant information for ADAS functions, such as the unique IDs, states, classifications, and pseudo-existence probabilities, are available; however, the tracks may still be unstable. The existence and stability of the tracks is determined by the calculated weights of the Gaussian components and the propagated ID tags in the GM-PHD filter. However, there are situations in which these methods reach their limits:

- Short-term occlusion of objects, where the track may disappear and a new one with a new ID is subsequently created.
- Clutter tracks created due to random false detections.
- If one of several sensors systematically fails to detect a target, the weight of the track may be low even though the object is tracked over a long period.

A track confirmation strategy is applied to overcome these limitations and pass only those tracks confirmed based on several criteria to the ADAS function. This consists of a confirmation list that is updated after each update cycle based on the track set T_k . Each element in the confirmation list consists of a confirmation track $\hat{T}_k^{(i)}$, an ID alias, the unobserved time, the time of first appearance t_{fa} , and a confirmation flag, as shown in the example in Figure 5. The unobserved time is the time since the ID of the track was last updated. Figure 5 shows an example scenario with two cars around the ego vehicle. The arrows show the trajectories of the tracks provided by MOT. One car is directly in front of the ego vehicle, but the provided existence probability is low. The other car is overtaking and occluded for a short duration, during which the tracks are lost. In addition, there is a false positive track in the right top corner. This scenario represents several different errors that can possibly occur for a MOT system.

In each update cycle of the confirmation strategy, the tracks in the confirmation list should be updated. Here, the ID alias of the stored confirmation track is compared with the ID of the tracks in the track set. If no match is found, then the unobserved time is increased and the state is predicted using the motion model. In Figure 5, this is the case during occlusion of the overtaking car with $\hat{\tau}^{(2)} = 2$. Otherwise, the unobserved time is set to 0.

Next, ID switches are searched for all tracks in the confirmation list that have not yet received an assignment from the update track set. If a track is found in the update track set for a non-updated track in the confirmation list for which the Euclidean distance of the location is below the threshold, then an ID change is assumed and the ID alias is set to the new ID, which is indicated in Figure 5 when the occluded car is tracked again. In this way, objects that have been lost for a short time, e.g., due to occlusion, can be found again while preserving their original unique ID. For all tracks in the update track set that

have not yet been assigned to a track in the confirmation list, a new element is created in the confirmation list.

For all elements in the list, the next step checks whether they can be marked as confirmed. To be confirmed, one of the following criteria must be met:

- $p(\exists) > p_{\exists, \min} \wedge t_k - t_{fa} > t_{\min}$
- $t_k - t_{fa} > t_{\text{conf}}$

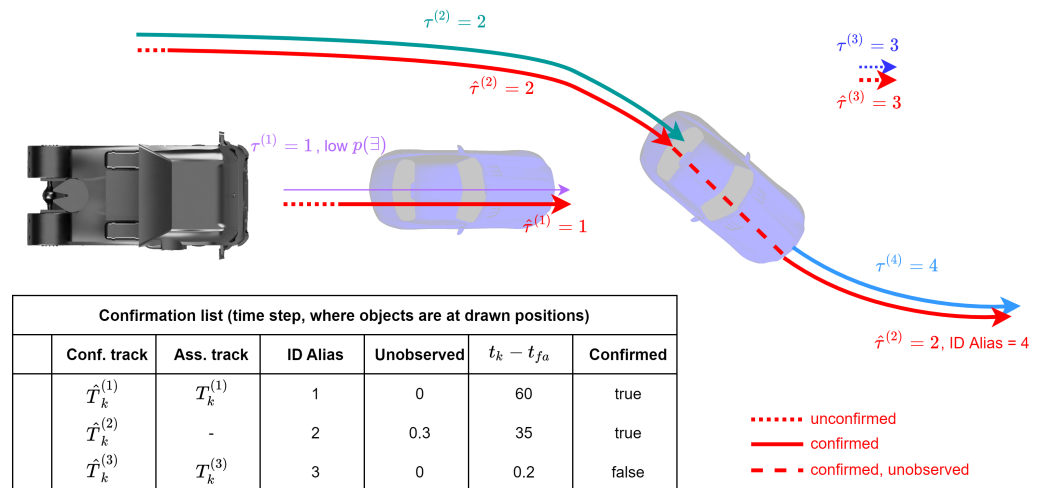


Figure 5. Example situation with track confirmation strategy. The trajectories of the confirmation list are drawn in red, while the trajectories of the track set have different colors. The confirmation list table shows example values at the time where the objects are at the cars' positions.

In order to be confirmed, a track must either have an existence probability $p(\exists)$ greater than a threshold $p_{\exists, \min}$ and at the same time be tracked for a minimum duration t_{\min} , or be tracked for a duration greater than t_{conf} regardless of the existence probability. In Figure 5, the car with low existence probability is confirmed, as it has been tracked for a long duration, while the false positive track is not confirmed, and consequently is not forwarded to the ADAS functions. All tracks in the confirmation list are then checked for deletion using the unobserved time. If the unobserved time is above the threshold, they are removed from the list. The threshold for already-confirmed tracks is higher than the threshold for unconfirmed tracks.

This confirmation strategy can filter out some of the errors shown in Figure 5. In particular, situations where ID changes occur or tracks are lost for a short period of time can be resolved. In addition, it can reduce the amount of false positive tracks provided to the ADAS application. Therefore, the confirmation strategy can help to improve overall system performance. In our experiments, the confirmation strategy was applied to both the GM-PHD approach and the Kalman approach.

4. Results

The proposed approach was evaluated on both the KITTI dataset [22] and a custom dataset recorded using a truck with multiple sensors that had partially overlapping sensor FoVs. The experiments show a comparison between the proposed approach and the KF reference approach on both datasets. In addition, the influence of the parameter models is shown and the runtimes are analyzed.

4.1. Datasets

The sensor setup and sensor FoVs of both datasets are visualized in Figure 6. As shown in the figure, both datasets have a forward-facing camera, while the KITTI dataset uses a larger FoV. The custom setup relies more on radars, with two corner Short Range Radar (SRR) and a central Long Range Radar (LRR). The KITTI dataset is well suited for the

comparisons in this paper, as it is well studied, widely used, and contains ground truth data. For the experiments, only objects of the “Car” type were examined, as they occur most frequently in the dataset and as such are the most meaningful.

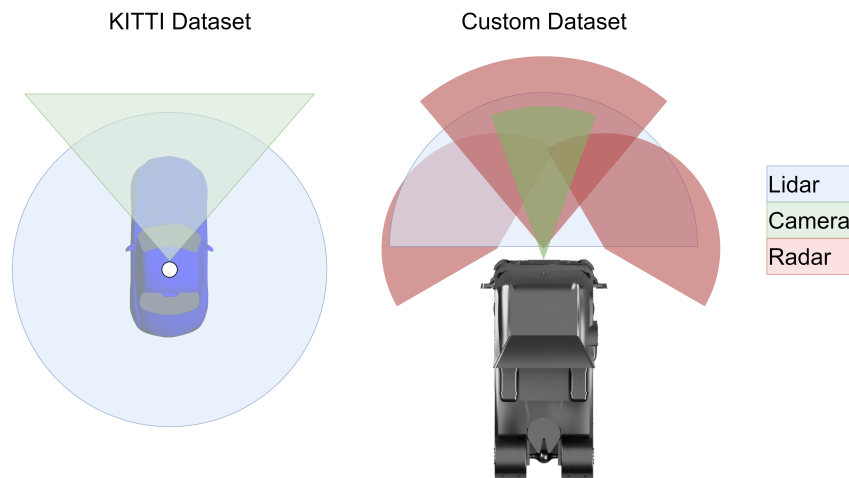


Figure 6. Comparison of dataset FoVs.

A new custom dataset was created to evaluate the tracking and fusion algorithms on the basis of current generation sensors and validate the results from the KITTI dataset. This dataset included sensors comparable to truck series equipment with integrated object detection in order to prove the validity and performance of the presented approaches for use in current-generation vehicles. The Lidar sensor was only used to generate ground truth OBB data in these experiments, with the other sensors used for tracking. The objects provided by the camera and radar sensors are geometrically represented either as a point, as a point with estimated width, or as an L-shape, depending on the situation and distance. In order to create a common representation, the detections of all sensors were converted to point objects without spatial expansion; thus, the length, width, height, and orientation are ignored for the custom dataset.

The custom dataset is small compared to KITTI, and contains three scenes from different operational design domains (highway, rural, and urban). Due to its smaller size, the custom dataset is less meaningful; however, it can still serve for confirmation or questioning of the results on the KITTI dataset, and provides the opportunity to study the fusion of more than two sensors with different and partially non-overlapping FoVs.

4.2. Comparison of Kalman and GM-PHD Filters on Real-World Data

First, a comparison of the GM-PHD approach and the Kalman approach is carried out to show the general potential of the presented approach. Only the FoV-dependency of the detection probability is used, with the parameters otherwise modeled as constant, as the Kalman approach does not take such dependencies into account either. The parameters are optimized for the fused setup for both datasets, and are kept the same for single-sensor tracking. Manual optimization of the parameters was carried out with the goal of achieving optimal results for both GM-PHD and KF. In order to provide a direct comparison of the filter outputs and prevent the comparison from being primarily dependent on the final optimization, the parameters were kept the same for both. Slight improvements were still possible by fine-tuning the parameters depending on the filter; thus, the results for separately tuned parameters are provided for the custom dataset.

The comparison was first conducted using the KITTI dataset. The training dataset of the MOT benchmark was used to show multiple comparisons. The Higher Order Tracking Accuracy (HOTA) metric [34] was used as a performance measure. As the KITTI dataset only contains raw data, the Point-RCNN [29] detection approach was used to provide detections based on Lidar and Yolo-Mono-3D [30] was used to provide camera-based

detections. Therefore, detections in the form of oriented bounding boxes suitable for processing are available. First, the performance of the Kalman and the GM-PHD approach is compared directly.

Figure 7 shows the HOTA results of both approaches on the KITTI dataset. The comparison is shown when using only Lidar, only camera, and fused data. The GM-PHD approach clearly outperforms the Kalman approach for Lidar only, and even more so for fused data. When comparing the sub-metrics on the fused data in Figure 8, the differences are particularly evident for “DetA” and “AssA” [34], which are measures of detection accuracy and association accuracy. This indicates that the GM-PHD approach has a particular advantage when creating/deleting tracks and ensuring correct ID assignment.

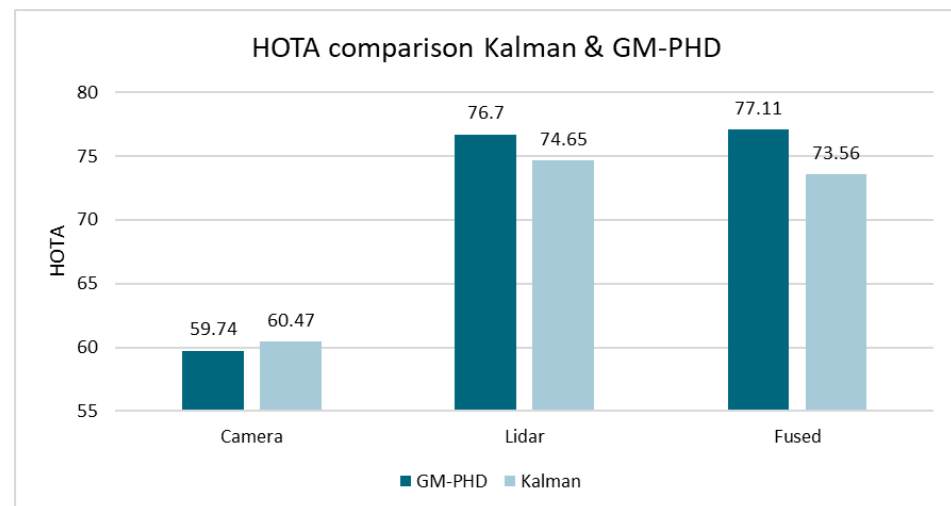


Figure 7. Comparison of HOTA results for Kalman and GM-PHD filters with different tracking setups on the KITTI dataset.

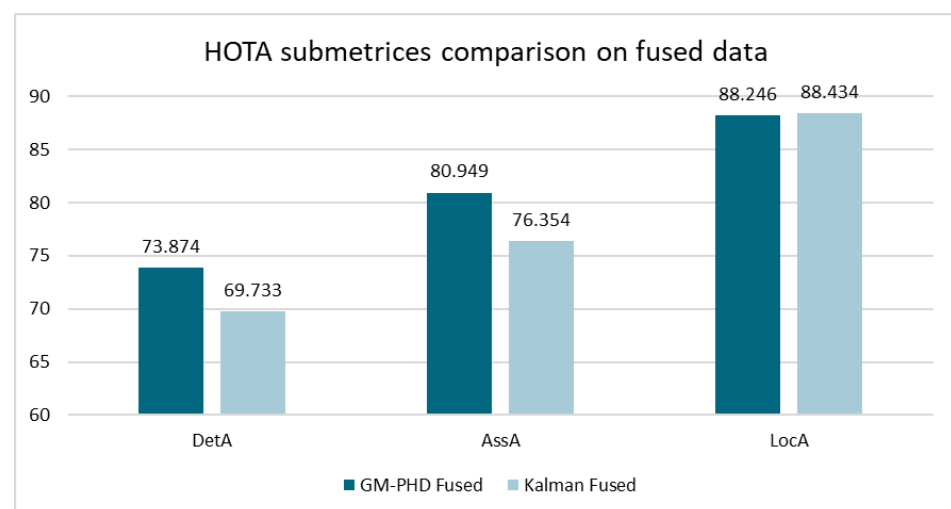


Figure 8. HOTA submetrics for fused tracking setup on the KITTI dataset.

Both filters were then compared using the custom dataset. Here, the Second Order Optimal Sub-Pattern Assignment (OSPA⁽²⁾) metric [35] with cut-off $c = 2.5$ and order $p = 1$ was used, while the Euclidean distance was used as the base distance. Note that lower values of this metric represent a lower error. For a simplified comparison, the mean OSPA⁽²⁾ distance over all time steps of all three scenes was used. Figure 9 shows the results for the single-sensor tracking and fused setups. As shown, the GM-PHD filter achieves lower mean distances for all tracking setups except camera-only. In addition to the results from Figure 9, where the parameters were the same for the GM-PHD filter and the KF, the results

for separately fine-tuned covariances are presented: the GM-PHD filter achieves an OSPA⁽²⁾ distance of 1.478, while the KF achieves 1.522 for the fused setup. While this performance difference is slightly decreased compared to the shared parameters, the GM-PHD approach still clearly outperforms the KF approach in this comparison.

Overall, the performance of the GM-PHD approach outperforms the Kalman approach in most tracking setups, and in particular for the fused data. The main advantages lie in its stable handling of appearing/disappearing tracks. For both datasets, the tracking performance in the camera-only setup is worse with the GM-PHD approach compared to the Kalman approach. This is probably due to the fact that the camera detections produce few clutter objects, but the position of the existing detections has a higher variance, which has a negative effect on the calculation of weights by the GM-PHD filter. In addition, the parameters are optimized for the fused setup and not fine-tuned for the single sensor tracking, which might have led to suboptimal results.

As shown by Figure 9, sensor-based parameter models can further reduce the OSPA⁽²⁾ distance, which is shown in detail in the following section.

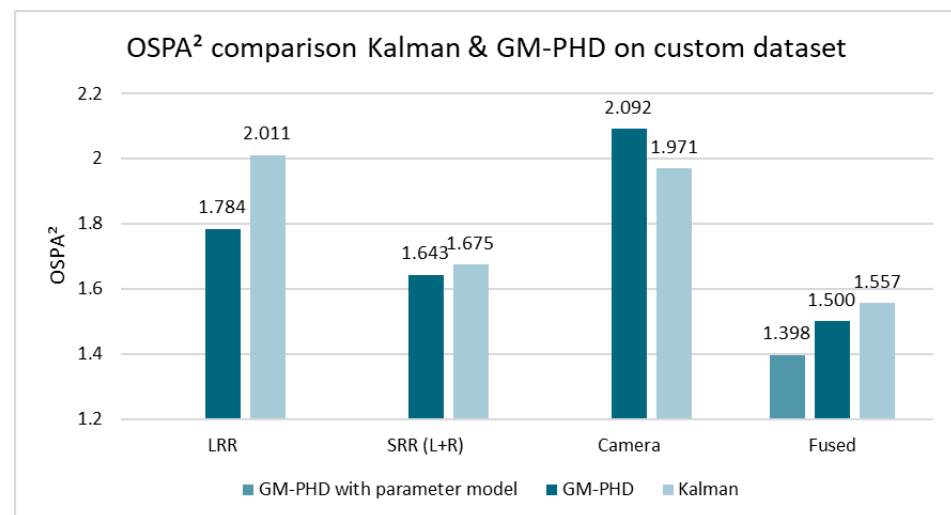


Figure 9. Comparison of OSPA⁽²⁾ error results for the Kalman and GM-PHD filters for different tracking setups on the custom dataset (smaller is better).

4.3. Influence of Sensor-Based Parameter Models

The sensor-based parameter models for detection probability and clutter density were evaluated with the GM-PHD approach. By applying the FoV to the detection probability, objects can be tracked through the FoVs of several sensors without applying additional rules. Figure 10a shows this on the custom dataset with an overtaking vehicle that is initially only in the FoV of the left SRR, then moves into the FoVs of all combined sensors. Without FoV modeling, stable tracking only occurs in the area of the combined view of all sensors (marked with an orange circle).

Because the detection capabilities of all sensors are quite different with respect to the distance, a similar effect occurs at higher distances, where some sensors are no longer capable of detecting an object while others remain able to detect it. Figure 10b shows a track from the custom dataset with a trajectory that is moving away from the ego vehicle. At a certain point, it is outside the capabilities of all sensors except the LRR; thanks to the distance-dependent model proposed here, it can still be tracked, while the track is lost when applying constant parameters (marked with an orange circle).

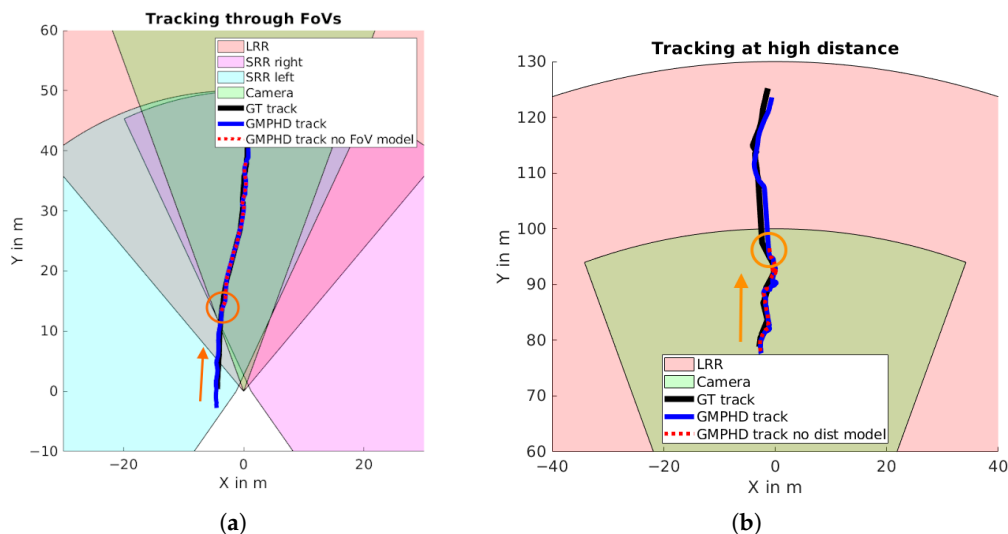


Figure 10. Example situations showing benefits of sensor-based parameter models. (a) An overtaking car moving in the direction of the orange arrow crosses multiple sensor FoVs. Stable tracking without the FoV model is reached at the overlap of all sensor FoVs. (b) Without the distance model for detection probability, the track of a car moving away in the direction of the orange arrow is lost at higher distances (the orange circle).

In addition to these detailed examples, a comparison of the performance with and without the sensor-based models was carried out for the tracking setups with sensor fusion. Because a comparison is not possible without modeling the FoV for the detection probability, this was applied to all scenarios, while the distance dependency was activated/deactivated depending on the test. Table 1 shows the results on the KITTI dataset and on the custom dataset. For both datasets, improvements can be achieved using the detection probability and the clutter density parameter model. However, the improvements on the KITTI dataset are rather small, while the improvements on the custom dataset show more potential. Here, the differences in the detection capabilities between the sensors is high in certain areas of the observed space and therefore needs to be modeled accordingly, which explains the higher improvements. In addition, the clutter density model shows less improvement for both datasets compared to the detection probability model. However, this is no limitation in the usage of the models, since improvements can be achieved in all situations, although they are small for some scenarios.

Table 1. Results of sensor-based parameter models using fused tracking setup on both datasets.

	KITTI Dataset (HOTA)	Custom Dataset (OSPA ⁽²⁾)
Constant model	77.11	1.50
p_d with dist model	77.26	1.42
κ with dist model	77.15	1.48
p_d and κ with dist model	77.34	1.40

4.4. Runtime Evaluation

The runtime of the presented GM-PHD approach was compared with the Kalman approach on the KITTI dataset and on the custom dataset with the Lidar-only setup. In addition, a variant without the gating in the update step was compared, as well as a variant where the detection probability was calculated using a mixture with seven components, as described in [11]. Table 2 shows the HOTA results and the average runtime for one cycle. All approaches were implemented in C++ and tested on an Intel Core I7-9700K CPU.

The runtime can be greatly reduced without significantly affecting the HOTA results by the introduction of gating. The original variant with mixture implementation for the detection probability is considerably slower. Thanks to the presented optimizations, the runtime is only 2.5 times that of the Kalman filter, which is not possible with other GM-PHD approaches. This means that the approach presented here is expected to work in embedded devices.

Table 2. Runtime comparison of multiple test scenarios.

	HOTA	Runtime in ms
Kalman approach	74.65	0.285
GM-PHD as proposed	76.86	0.715
GM-PHD without gating	76.84	1.629
GM-PHD without gating and with mixture p_d	75.81	15.015

In general, fast reactions are as important as accuracy for ADAS functions. If an emergency function needs to pursue the absolute minimum reaction times, KF might be the preferable choice. In real systems, it is necessary to calculate the updates for all sensors with cycle times of 10–20 Hz. If only limited computing power is available and these cycles cannot be achieved with the GM-PHD filter, then KF needs to be used. However, as KF has been used in sensor fusion systems for years and the available computing power continues to increase, experiments have shown that an update cycle of the GM-PHD filter can be calculated within the range of a millisecond; it can be assumed that this constraint can also be met by the GM-PHD filter in most systems nowadays. In these cases, the improved accuracy of the GM-PHD will benefit the system.

5. Conclusions

This paper shows a GM-PHD MOT approach that uses optimizations to enable easy application in real-time automotive systems. It supports multiple sensors with different FoVs and sensing capabilities by embracing sensor-based parameter models. Furthermore, the proposed approach introduces the possibility of propagating additional track properties, such as the classification, with the GM-PHD filter through time, thereby enhancing the filter's versatility. The proposed approach is able to achieve low runtimes while exceeding a Kalman-based MOT in terms of tracking performance, and can handle the appearance and disappearance of objects as well as the data association problem implicitly due to its RFS properties. The proposed GM-PHD approach achieved an OSPA⁽²⁾ error of 1.40 on the custom dataset, compared to 1.56 for the KF approach; therefore, this paper shows that the proposed GM-PHD filter approach is very well suited as a substitute for KF in automotive MOT and fusion systems. This allows existing systems to be improved in the short term.

Author Contributions: Conceptualization, C.B. and V.S.; methodology, C.B. and V.S.; software, C.B.; investigation, C.B.; resources, C.B.; writing—original draft preparation, C.B.; writing—review and editing, C.B. and V.S.; supervision, V.S.; project administration, V.S. All authors have read and agreed to the published version of the manuscript.

Funding: The publication of this article was funded by the Open Access fund of Universität Stuttgart.

Data Availability Statement: Some or all data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: Author 1 was employed by the company Daimler Truck AG. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADAS	Advanced Driver Assistance Systems
BEV	Bird's-Eye View
FoV	Field of View
GM-PHD	Gaussian Mixture Probability Hypothesis Density
HOTA	Higher-Order Tracking Accuracy
KF	Kalman Filter
KLD	Kullback–Leibler Divergence
LRR	Long-Range Radar
MOT	Multi-Object Tracking
OBB	Oriented Bounding Box
OSPA ⁽²⁾	Second-Order Optimal Sub-Pattern Assignment
PHD	Probability Hypothesis Density
RFS	Random Finite Set
SRR	Short-Range Radar

References

- Zaitouny, A.; Stemler, T.; Algar, S.D. Optimal Shadowing Filter for a Positioning and Tracking Methodology with Limited Information. *Sensors* **2019**, *19*, 931. [[CrossRef](#)] [[PubMed](#)]
- Gustafsson, F.; Gunnarsson, F.; Bergman, N.; Forssell, U.; Jansson, J.; Karlsson, R.; Nordlund, P.J. Particle filters for positioning, navigation, and tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 425–437. [[CrossRef](#)]
- Kampker, A.; Sefati, M.; Rachman, A.S.A.; Kreisköther, K.D.; Campoy, P. Towards Multi-Object Detection and Tracking in Urban Scenario under Uncertainties. In Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018), Funchal, Madeira, Portugal, 16–18 March 2018.
- Wu, H.; Han, W.; Wen, C.; Li, X.; Wang, C. 3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5668–5677. [[CrossRef](#)]
- Mobus, R.; Kolbe, U. Multi-target multi-object tracking, sensor fusion of radar and infrared. In Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy, 14–17 June 2004; pp. 732–737. [[CrossRef](#)]
- Chiu, H.K.; Prioletti, A.; Li, J.; Bohg, J. Probabilistic 3D Multi-Object Tracking for Autonomous Driving. *arXiv* **2020**, arXiv:2001.05673. [[CrossRef](#)]
- Himmelsbach, M.; Luettel, T.; Wuensche, H.J. Real-time object classification in 3D point clouds using point feature histograms. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 994–1000. [[CrossRef](#)]
- Schueler, K.; Weiherer, T.; Bouzouraa, E.; Hofmann, U. 360 Degree multi sensor fusion for static and dynamic obstacles. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 692–697. [[CrossRef](#)]
- Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
- Mahler, R.P.S. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1152–1178. [[CrossRef](#)]
- Vo, B.N.; Ma, W.K. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Trans. Signal Process.* **2006**, *54*, 4091–4104. [[CrossRef](#)]
- Liu, L.; Ji, H.; Zhang, W.; Liao, G. Multi-Sensor Multi-Target Tracking Using Probability Hypothesis Density Filter. *IEEE Access* **2019**, *7*, 67745–67760. [[CrossRef](#)]
- Mahler, R.P.S. Approximate multisensor CPHD and PHD filters. In Proceedings of the 2010 13th International Conference on Information Fusion, Edinburgh, UK, 26–29 July 2010; pp. 1–8. [[CrossRef](#)]
- Liu, L.; Ji, H.; Fan, Z. A cardinality modified product multi-sensor PHD. *Inf. Fusion* **2016**, *31*, 87–99. [[CrossRef](#)]
- Li, B.; Ouyang, W.; Sheng, L.; Zeng, X.; Wang, X. GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1019–1028. [[CrossRef](#)]
- Vasic, M.; Martinoli, A. A Collaborative Sensor Fusion Algorithm for Multi-object Tracking Using a Gaussian Mixture Probability Hypothesis Density Filter. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 491–498. [[CrossRef](#)]
- Li, G.; Yi, W.; Jiang, M.; Kong, L. Distributed fusion with PHD filter for multi-target tracking in asynchronous radar system. In Proceedings of the 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, 8–12 May 2017; pp. 1434–1439. [[CrossRef](#)]
- Li, G.; Yi, W.; Li, S.; Wang, B.; Kong, L. Asynchronous multi-rate multi-sensor fusion based on random finite set. *Signal Process.* **2019**, *160*, 113–126. [[CrossRef](#)]
- Lindenmaier, L.; Aradi, S.; Becsi, T.; Toro, O.; Gaspar, P. GM-PHD Filter Based Sensor Data Fusion for Automotive Frontal Perception System. *IEEE Trans. Veh. Technol.* **2022**, *71*, 7215–7229. [[CrossRef](#)]

20. Törő, O.; Bécsi, T.; Gáspár, P. PHD Filter for Object Tracking in Road Traffic Applications Considering Varying Detectability. *Sensors* **2021**, *21*, 472. [[CrossRef](#)] [[PubMed](#)]
21. Chen, X.; Li, Y.; Li, Y.; Yu, J. PHD and CPHD Algorithms Based on a Novel Detection Probability Applied in an Active Sonar Tracking System. *Appl. Sci.* **2018**, *8*, 36. [[CrossRef](#)]
22. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [[CrossRef](#)]
23. Clark, D.; Panta, K.; Vo, B.N. The GM-PHD Filter Multiple Target Tracker. In Proceedings of the 2006 9th International Conference on Information Fusion, Florence, Italy, 10–13 July 2006; pp. 1–8. [[CrossRef](#)]
24. Panta, K.; Vo, B.N.; Clark, D.E. An Efficient Track Management Scheme for the Gaussian-Mixture Probability Hypothesis Density Tracker. In Proceedings of the 2006 Fourth International Conference on Intelligent Sensing and Information Processing, Bangalore, India, 15 October–18 December 2006; pp. 230–235. [[CrossRef](#)]
25. Challa, S.; Morelande, M.R.; Musicki, D.; Evans, R.J. *Fundamentals of Object Tracking*, 1st ed.; Cambridge University Press: Cambridge, UK, 2011.
26. Konstantinova, P.; Udvarov, A.; Semerdjiev, T. A study of a target tracking algorithm using global nearest neighbor approach. In Proceedings of the 4th International Conference Conference on Computer Systems and Technologies e-Learning–CompSysTech’03, Rousse, Bulgaria, 19–20 June 2003; Rachev, B., Smrikarov, A., Eds.; Association for Computing Machinery: New York, NY, USA, 2003; pp. 290–295. [[CrossRef](#)]
27. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [[CrossRef](#)]
28. Aeberhard, M. Object-Level Fusion for Surround Environment Perception in Automated Driving Applications. Ph.D. Dissertation, Technical University of Dortmund, Dortmund, Germany, 2017.
29. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 770–779. [[CrossRef](#)]
30. Liu, Y.; Wang, L.; Liu, M. YOLOStereo3D: A Step Back to 2D for Efficient Stereo 3D Detection. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, 30 May–5 June 2021; pp. 13018–13024. [[CrossRef](#)]
31. Houssineau, J.; Laneuville, D. PHD filter with diffuse spatial prior on the birth process with applications to GM-PHD filter. In Proceedings of the 2010 13th International Conference on Information Fusion, Edinburgh, UK, 26–29 July 2010; pp. 1–8. [[CrossRef](#)]
32. Ristic, B. *Particle Filters for Random Set Models*; Springer: New York, NY, USA, 2013.
33. Granström, K.; Orguner, U. On the reduction of Gaussian inverse Wishart mixtures. In Proceedings of the 2012 15th International Conference on Information Fusion, Singapore, 9–12 July 2012; pp. 2162–2169.
34. Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; Leibe, B. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 548–578. [[CrossRef](#)]
35. Beard, M.; Vo, B.T.; Vo, B.N. A Solution for Large-Scale Multi-Object Tracking. *IEEE Trans. Signal Process.* **2020**, *68*, 2754–2769. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.