

A Collaborative Sensor Fusion Algorithm for Multi-Object Tracking Using a Gaussian Mixture Probability Hypothesis Density Filter

Milos Vasic and Alcherio Martinoli

Abstract—This paper presents a method for collaborative tracking of multiple vehicles that extends a Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter with a collaborative fusion algorithm. Measurements are preprocessed in a detect-before-track fashion, and cars are tracked using a rectangular shape model. The proposed method successfully mitigates clutter and occlusion problems. In order to extend the field of view of individual vehicles and increase the estimation confidence in the areas where a target is observable by multiple vehicles, PHD intensities are exchanged between vehicles and fused in the Collaborative GM-PHD filter using a novel algorithm based on the Generalized Covariance Intersection. The method is extensively evaluated using a calibrated, high-fidelity simulator in scenarios where vehicles exhibit both straight and curved motion at different speeds.

I. INTRODUCTION

One of the key components of an intelligent vehicle is its perception system. By having a precise representation of the environment, the intelligent vehicle can identify hazards, assess the level of danger, give warnings and recommendations to a driver or perform evasive actions if the driver is unable to do so in a timely fashion.

There often exist situations in which a vehicle does not have a complete view of the environment surrounding it. For instance, it can be due to occlusions, limited field of view (FOV), or sensor range limitations. In these situations, information coming from other vehicles or the road infrastructure substantially improves the decision making process of the intelligent vehicle. Information sharing is usually achieved using wireless communication links. Communication can be considered as a type of virtual sensor, such as in [1], [2]. For example, it can enable vehicles to see behind corners, as the wireless communication does not have the same line-of-sight constraints as most conventional sensors have. Collaborative fusion considers sharing of tracks or perception data (raw or processed) among two or more vehicles. If a collaborative vehicle uses simpler sensors (e.g., a speed sensor whose output is available on the vehicle's CAN bus), it can transmit sensory raw data. However, in case of richer sensors (e.g., cameras or lidars), due to data volume and communication channel constraints, sharing only processed data (e.g., objects) and filtered data (i.e., tracks) may be the only feasible option.

M. Vasic and A. Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. milos.vasic@epfl.ch, alcherio.martinoli@epfl.ch

This work is financially supported by PSA Peugeot Citroën and has benefited of the technical and administrative support of the EPFL's Transportation Center.

Multiple-object tracking problems are concerned with multiple targets and multiple measurements, where the association of measurements with appropriate targets is not known. Filters based on the Global Nearest Neighbor (GNN) algorithm [3] and Multiple Hypothesis Tracking (MHT) filters [4] involve explicit association between measurements and targets. Issues with such filters are related to having multiple observations of a single object, clutter and occluded objects. Joint Probabilistic Data Association (JPDA) filters [4] weight observations using their association probabilities, but require the number of objects to be known a priori. In our work the number of objects is unknown and varies with time.

Another solution for multi-object tracking is based on Random Finite Set (RFS) models, in which a set of objects of variable cardinality is modeled as a random finite set. Filters based on this theory, such as Probability Hypothesis Density (PHD) filters and Cardinalized PHD (CPHD) filters, deal with the measurement-to-track association implicitly [5]. They can provide higher robustness and accuracy in scenarios where the number of targets is variable and/or not known in advance. Many different implementations of the PHD filter have been developed and used for multi-object tracking; the most common ones are the Gaussian Mixture PHD (GM-PHD) [6] and the Sequential Monte Carlo PHD (SMC-PHD) filter [7]. Typical implementations assume that, if a target is detected, it generates only one measurement. Moreover, in [6] and [7], the shape of the target is not taken into account.

Lidar sensors typically produce multiple measurements per single object, depending on its distance and angle. Targets that are assumed to generate multiple measurements are called extended targets. Granstrom et al. have extended a GM-PHD framework to handle extended targets [8], [9]. Estimation of the target shape was also included in the GM-PHD framework. For instance, a thin stick model was used for bicycle tracking in [10] and tracking of rectangular and elliptical extended targets was addressed in [11]. The drawbacks of these two approaches are the unconventional framework where the measurement model depends on the current set of measurements, as well as a high degree of non-linearity which makes the state update using an Extended or Unscented Kalman Filter difficult. In [12] Swain and Clark tracked the kinematic state of elliptical targets along with their shape parameters by considering hierarchical point process representations of multiple extended targets. The particle representation was used for the shape, while the Gaussian mixture formulation was used as a representation of a kinematic state per particle. Given that in collaborative fusion, information needs to be exchanged between vehicles, and that the GM implementation requires less communication

bandwidth than its SMC counterpart, we find approaches based solely on GM better suited for the problem at hand.

In our approach, we use a detection algorithm prior to tracking to extract features from the observed sensor data. These features are then used as measurements during the GM-PHD update. Our detection algorithm, explained in Section III-B, is similar to the approach presented in [3]. However, to the best of our knowledge, this approach has not been used in the RFS framework so far.

Two major problems related to collaborative fusion are temporal and spatial alignment [13]. Temporal alignment deals with variable delays of wireless communication links. Authors of [14] report that the typical delay of 802.11p wireless communication protocol is in the order of 100-150 ms. Consequently, variables received through communication may not represent faithfully the state of objects at the time they were received. Ideally, vehicle receiving observations should help to predict the evolution of the objects' state between the original measurement time and that of reception. This is obviously more difficult if raw measurements are communicated instead of tracks. Moreover, spatial alignment considers differences in coordinate systems of the sending and receiving vehicles. The difficulty here lies in the uncertainty of sending and receiving vehicle locations.

When using a PHD filter for tracking, communicating tracks is reduced to communicating the PHD intensities. A method for fusing PHD intensity functions is given in [15]. A distributed fusion of SMC-CPHD filters via exponential mixture densities (EMD) has been presented by Uney et al. in [16]. Battistelli et al. used EMD for distributed fusion of GM-CPHD densities [17]. In all three works, it has been assumed that all PHD filters work in the same domain, i.e., that all agents share a common FOV in which targets are sensed. This is a very limiting assumption for the application of aforementioned methods in the field of (moving) vehicles.

In this paper we address the problem of collaborative object tracking. Specifically, we focus on tracking of cars (and not vehicles in general, such as trucks or bicycles). Multiple cars are equipped with sensors and track other cars that are located in their respective FOVs. Cars have the ability to cooperate, i.e., share and fuse object estimates among each other. The contributions of our paper are two-fold. Firstly, we propose a complete method for vehicle tracking based on the GM-PHD filter and a car detection model that takes the car shape into account. Secondly, we present a general method for collaborative fusion of GM-PHD densities, in which vehicles' FOVs do not need to overlap. Our approach enables extending the FOV of a single vehicle beyond the limits of its sensors by using communication, and decreasing uncertainty of estimation where a target is observed by multiple vehicles.

II. COLLABORATIVE GM-PHD FILTER

To tackle the problem of collaborative multi-object tracking, we propose a Collaborative GM-PHD (C-GM-PHD) filter. It extends the GM-PHD filter from [6] with the fusion block, that has the ability to fuse PHD intensities coming from other GM-PHD or C-GM-PHD filters with its own PHD intensity, therefore increasing its estimation confidence and extending its FOV. The concept is illustrated in Fig. 1.

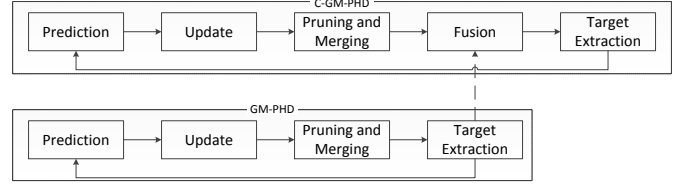


Fig. 1. Block diagram of C-GM-PHD steps. It fuses the intensity received from a GM-PHD (or C-GM-PHD) filter, if available.

Assume that the posterior intensity at time $k-1$ is a Gaussian mixture of J_{k-1} components with weights w_{k-1} , means m_{k-1} and covariances P_{k-1} of the form

$$v_{k-1}(\mathbf{x}) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(\mathbf{x}; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \quad (1)$$

In order to compute the posterior $v_k(\mathbf{x})$, the C-GM-PHD filter contains five fundamental steps, as described below. While steps 1), 2), 3), and 5) are taken from [6], step 4) is our contribution towards the C-GM-PHD filter.

1) *Prediction*: The predicted intensity for time k is also a Gaussian mixture

$$v_{k|k-1}(\mathbf{x}) = \sum_{j=1}^{J_{k-1}} p_{S,k}(m_{k|k-1}^{(j)}) w_{k-1}^{(j)} \mathcal{N}(\mathbf{x}; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}) + \gamma_k(\mathbf{x}) \quad (2)$$

where $\gamma_k(\mathbf{x})$ is the birth intensity with $J_{\gamma,k}$ components given by

$$\gamma_k(\mathbf{x}) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathcal{N}(\mathbf{x}; m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}) \quad (3)$$

and $p_{S,k}(m_{k|k-1}^{(j)})$ is the probability of survival of Gaussian component j at time k . Note that, for simplicity, we do not include spawning of targets in our filter implementation.

The predicted mean and covariance of estimated GM components can be computed using any Bayesian estimator. In our work we use an Unscented Kalman Filter (UKF):

$$m_{k|k-1} = \sum_{i=0}^{2\Lambda} \Upsilon^{(i)} \chi_{k|k-1}^{(i)} \quad (4)$$

$$P_{k|k-1} = \sum_{i=0}^{2\Lambda} \Upsilon^{(i)} [\chi_{k|k-1}^{(i)} - m_{k|k-1}][\chi_{k|k-1}^{(i)} - m_{k|k-1}]^\top \quad (5)$$

where $\chi_{k|k-1}^{(i)}$ is a sigma point computed by the Unscented Transform and $\Upsilon^{(i)}$ its associated weight. The number of state dimensions is denoted by Λ , and the total number of sigma points used by UKF is $2\Lambda + 1$.

2) *Update*: For a set of measurements Z_k , the update step yields a posterior intensity given by:

$$v_k(\mathbf{x}) = \sum_{i=1}^{J_{k-1}} [1 - p_{D,k}(m_{k|k-1}^{(i)})] w_{k|k-1}^{(i)} \mathcal{N}(\mathbf{x}; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) + \sum_{\mathbf{z} \in Z_k} \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(\mathbf{z}) \mathcal{N}(\mathbf{x}; m_{k|k}^{(j)}(\mathbf{z}), P_{k|k}^{(j)}) \quad (6)$$

where

$$w_k^{(j)}(\mathbf{z}) = \frac{p_{D,k}(m_{k|k-1}^{(j)})w_{k|k-1}^{(j)}q_k^{(j)}(\mathbf{z})}{\kappa_k(\mathbf{z}) + \sum_{l=1}^{J_{k|k-1}} p_{D,k}(m_{k|k-1}^{(l)})w_{k|k-1}^{(l)}q_k^{(l)}(\mathbf{z})} \quad (7)$$

$$q_k^{(j)}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; H_k^{(j)}m_{k|k-1}^{(j)}, R_k + H_k^{(j)}P_{k|k-1}^{(j)}[H_k^{(j)}]^\top) \quad (8)$$

$$m_{k|k}^{(j)}(\mathbf{z}) = m_{k|k-1}^{(j)} + K_k^{(j)}(\mathbf{z} - H_k m_{k|k-1}^{(j)}) \quad (9)$$

$$P_{k|k}^{(j)} = [I - K_k^{(j)}H_k]P_{k|k-1}^{(j)} \quad (10)$$

$$K_k^{(j)} = P_{k|k-1}^{(j)}H_k^\top(H_kP_{k|k-1}^{(j)}H_k^\top + R_k)^{-1} \quad (11)$$

The parameters used in the update step are the clutter level $\kappa_k(\mathbf{z})$, the probability of detection $p_{D,k}(m_{k|k-1}^{(i)})$ dependent on the mean of the Gaussian component i , the observation model H_k and the observation noise covariance R_k .

3) *Pruning and Merging*: After the update step, the number of Gaussian components increases quadratically with the number of measurements. It is therefore necessary to limit the number of Gaussians, as to keep the problem tractable. A good approximation of the Gaussian mixture posterior intensity can be obtained by truncating components with weak weights. In other words, only a set containing Gaussian components $I = \{i = 1, \dots, J_k | w_k^{(i)} > T_p\}$ with weights higher than a certain pruning threshold T_p are kept.

Furthermore, all Gaussian components that are close to each other can be approximated by a single Gaussian. This is done by selecting a Gaussian component with a highest weight, and finding all Gaussians whose mean is within the Mahalanobis distance U from the selected Gaussian. This yields a set $L = \{i \in I | (m_k^{(i)} - m_k^{(j)})^\top (P_k^{(i)})^{-1} (m_k^{(i)} - m_k^{(j)}) \leq U\}$ of Gaussian components that are merged as follows:

$$\tilde{w}_k^{(l)} = \sum_{i \in L} w_k^{(i)} \quad (12)$$

$$\tilde{m}_k^{(l)} = \frac{1}{\tilde{w}_k^{(l)}} \sum_{i \in L} w_k^{(i)} x_k^{(i)} \quad (13)$$

$$\tilde{P}_k^{(l)} = \frac{1}{\tilde{w}_k^{(l)}} \sum_{i \in L} w_k^{(i)} (P_k^{(i)} + (\tilde{m}_k^{(l)} - m_k^{(i)})(\tilde{m}_k^{(l)} - m_k^{(i)})^\top) \quad (14)$$

4) *Fusion*: After information (in the form of the GM-PHD intensity) has been received from the collaborative vehicle \mathcal{C} over the communication link, it needs to be fused with the local intensity of the ego vehicle \mathcal{E} .

The first step is to transform the intensity from the collaborative vehicle's coordinate frame to the ego vehicle-centered frame. However, the transformation between the two frames is not exactly known, due to the pose uncertainty of the two vehicles. Hence, we will refer to it as an *approximate transformation*. Approximate transformations [18] allow for calculating the nominal location and associated error of any object relative to any other object. In Fig. 2 we first describe the coordinates $\mathbf{x}_{\mathcal{NC}}^W$ of a non-collaborative vehicle \mathcal{NC} (target) with respect to the world reference frame W , by carrying out transformations from W to the collaborative vehicle frame \mathcal{C} ($\mathbf{x}_{\mathcal{C}}^W$) and from the frame of \mathcal{C} to the target ($\mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}$). If the explicit transformation is given by a vector function

$$\mathbf{x}_{\mathcal{NC}}^W = f(\mathbf{x}_{\mathcal{C}}^W, \mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}) \quad (15)$$

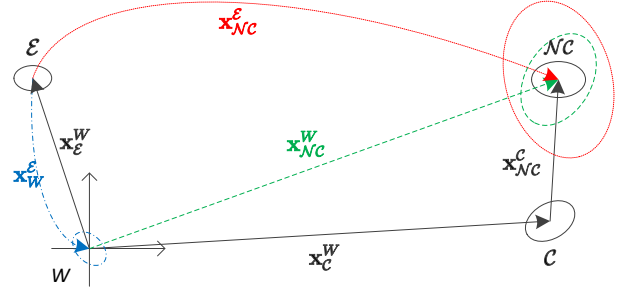


Fig. 2. A sequence of approximate transformations. The transformation from the world frame W to the collaborative vehicle frame \mathcal{C} is denoted by $\mathbf{x}_{\mathcal{C}}^W$. Collaborative vehicle tracks a target \mathcal{NC} in its own frame; the target's position is defined by the transformation $\mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}$. The location of the tracked target can be expressed in the world frame using the transformation $\mathbf{x}_{\mathcal{NC}}^W = \mathbf{x}_{\mathcal{C}}^W + \mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}$. The transformation $\mathbf{x}_{\mathcal{NC}}^{\mathcal{E}}$ can be obtained as $\mathbf{x}_{\mathcal{W}}^{\mathcal{E}} + \mathbf{x}_{\mathcal{NC}}^W$, where $\mathbf{x}_{\mathcal{W}}^{\mathcal{E}} = -\mathbf{x}_{\mathcal{E}}^W$. Reference frames other than W are omitted for clarity.

and the vector function is approximated by a first-order Taylor series expansion about the means of the variables, the mean value of the approximate transformation is obtained by applying the function to variable means, i.e., $\hat{\mathbf{x}}_{\mathcal{NC}}^W = f(\hat{\mathbf{x}}_{\mathcal{C}}^W, \hat{\mathbf{x}}_{\mathcal{NC}}^{\mathcal{C}})$. The covariance matrix of this transformation is

$$P_3 = J \begin{pmatrix} P_1 & 0 \\ 0 & P_2 \end{pmatrix} J^\top = HP_1H^\top + KP_2K^\top \quad (16)$$

where P_1 and P_2 are respectively covariances of transformations $\mathbf{x}_{\mathcal{C}}^W$ and $\mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}$, and J is the Jacobian of the transformation

$$J = \left(\frac{\partial f}{\partial \mathbf{x}_{\mathcal{C}}^W} \middle| \frac{\partial f}{\partial \mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}} \right) = [H|K] \quad (17)$$

Transformation $\mathbf{x}_{\mathcal{W}}^{\mathcal{E}}$ (see Fig. 2) is obtained by reversing the transformation from W to \mathcal{E} ($\mathbf{x}_{\mathcal{E}}^W$), and the covariance matrix for the reversed transformation (e.g., P') is estimated from the given covariance matrix of the transformation $\mathbf{x}_{\mathcal{E}}^W$ (e.g., matrix P) as

$$P' = R \cdot P \cdot R^\top \quad (18)$$

where R is the Jacobian of the reversed transformation equations. Finally, the transformation from \mathcal{E} to the target ($\mathbf{x}_{\mathcal{NC}}^{\mathcal{E}}$) and its covariance matrix are obtained using the transformations $\mathbf{x}_{\mathcal{W}}^{\mathcal{E}}$ and $\mathbf{x}_{\mathcal{NC}}^W$, similarly as in the case of the transformation $\mathbf{x}_{\mathcal{NC}}^W$ being obtained from the transformations $\mathbf{x}_{\mathcal{C}}^W$ and $\mathbf{x}_{\mathcal{NC}}^{\mathcal{C}}$.

We use the above method to perform coordinate transformations. Once the means and covariances of all Gaussians are expressed in the local reference frame of the ego vehicle, they can be fused with the local C-GM-PHD intensity.

Optimal (Bayesian) fusion techniques require statistically independent information, but this assumption does not hold if multiple sources are exchanging information between each other. If the common information across the sources is not known and cannot be canceled out, optimal fusion would lead to over-confidence [17]. Covariance Intersection (CI) [19] is a conservative method for fusing two Gaussian distributions, whose cross-correlation is unknown. CI is known to estimate the fused uncertainty that will neither be too small (over-confidence) nor too large (under-confidence). In [20], Mahler proposed a generalized method for fusion of multi-object functions, called Generalized Covariance Intersection (GCI). It is shown in [15]

that the location density of the two fused GM-PHD posteriors becomes

$$v^{\mathcal{W}}(\mathbf{x}) = \frac{v_1^{\mathcal{W}}(\mathbf{x})v_2^{1-\mathcal{W}}(\mathbf{x})}{\int v_1^{\mathcal{W}}(\mathbf{y})v_2^{1-\mathcal{W}}(\mathbf{y})d\mathbf{y}} \quad (19)$$

where v_1 and v_2 are the two individual GM-PHD posterior intensities and $0 \leq \mathcal{W} \leq 1$ is a fusion parameter.

Given that posterior intensities are Gaussian Mixture Models, the following approximation of the Exponential Gaussian Mixture is used in order to preserve the GM form of the location PDF [17], [21]:

$$\begin{aligned} \left[\sum_{j=1}^{J_k} w_j \mathcal{N}(\mathbf{x}; m_j, P_j) \right]^{\mathcal{W}} &\cong \sum_{j=1}^{J_k} [w_j \mathcal{N}(\mathbf{x}; m_j, P_j)]^{\mathcal{W}} \\ &= \sum_{j=1}^{J_k} w_j^{\mathcal{W}} \frac{[\det(2\pi P_j^{\mathcal{W}})]^{\frac{1}{2}}}{[\det(2\pi P_j)]^{\frac{\mathcal{W}}{2}}} \mathcal{N}(\mathbf{x}, m_j, P_j / \mathcal{W}) \end{aligned} \quad (20)$$

Using (20), GCI fusion reduces to applying CI to each pair of Gaussian components from v_1 and v_2 in turn, resulting in the total number of $J_k(v_1) \times J_k(v_2)$ components in the fused intensity.

The aforementioned approach was developed for fusion of sensors whose FOVs overlap completely, and it does not work well in the general case, when the domains over which GMs are defined are not the same (i.e., when the FOVs of different sensors do not entirely overlap). It is evident that GCI assigns significant weight only to components located in the common FOV of multiple sensors.

Our fusion algorithm is given in Algorithm I. We apply CI only to Gaussian components that are close to each other (lines 5-17), while keeping the rest of Gaussian components intact (lines 19-23). Closeness of Gaussian components is expressed through Mahalanobis distance with the threshold U_F (line 7). The cardinality of the fused set is computed as the weighted cardinality of the two initial sets (line 18). After fusing, we merge the components that are close to each other, similar to Sec. II-3. This approach enables us to have the same fusion experience in the common FOV of the two sensors as when GCI is used, while at the same time keeping track of all the targets that are in the FOV of only one sensor. We note that the accuracy of vehicles' pose plays an important role in the sensor fusion step.

5) *Extraction*: The extraction of multiple-target state estimates is straightforward from the GM representation of RFS intensity. According to [6], means of Gaussians that have weights greater than some threshold T_e (e.g., $T_e > 0.5$) are extracted and represent the most likely estimates.

III. COLLABORATIVE CAR TRACKING

Section II shows a general method that can be applied to tracking of any kind of object using any sensor modality. In this section, we provide additional details and models that we use to perform tracking of cars using lidars.

A. Kinematic state and motion model

Since in this paper we focus on car tracking, we can afford to use a rectangular model for representing targets, whose

Require: $\{m_1^{(i)}, P_1^{(i)}, w_1^{(i)}\}_{i=1}^{N_1}, \{m_2^{(j)}, P_2^{(j)}, w_2^{(j)}\}_{j=1}^{N_2}, U_F, \mathcal{W}$

```

1:  $I_F = J_F = m = \emptyset$ 
2:  $\alpha_1^{(i)} = w_1^{(i)} / \sum_k w_1^{(k)}, i = \{1, \dots, N_1\}$ 
3:  $\alpha_2^{(j)} = w_2^{(j)} / \sum_l w_2^{(l)}, j = \{1, \dots, N_2\}$ 
4:  $N_{12} = 0$ 
5: for  $i = 1$  to  $N_1$  do
6:   for  $j = 1$  to  $N_2$  do
7:     if  $(m_1^{(i)} - m_2^{(j)})^\top (0.5(P_1^{(i)} + P_2^{(j)}))^{-1} (m_1^{(i)} - m_2^{(j)}) \leq U_F$  then
8:        $I_F = I_F \cup i$  {save indices of fused components}
9:        $J_F = J_F \cup j$ 
10:       $\alpha = (\alpha_1^{(i)})^{\mathcal{W}} (\alpha_2^{(j)})^{1-\mathcal{W}} \kappa(\mathcal{W}, P_1^{(i)}) \kappa(1-\mathcal{W}, P_2^{(j)})$ 
         $\cdot \mathcal{N}(m_1^{(i)} - m_2^{(j)}, 0, P_1^{(i)} / \mathcal{W} + P_2^{(j)} / (1-\mathcal{W}))$ 
        where  $\kappa(\mathcal{W}, P) = [\det(2\pi P^{\mathcal{W}})]^{\frac{1}{2}} / [\det(2\pi P)]^{\frac{\mathcal{W}}{2}}$ 
         $P = [\mathcal{W}(P_1^{(i)})^{-1} + (1-\mathcal{W})(P_2^{(j)})^{-1}]^{-1}$ 
11:       $m = P[\mathcal{W}(P_1^{(i)})^{-1} m_1^{(i)} + (1-\mathcal{W})(P_2^{(j)})^{-1} m_2^{(j)}]$ 
12:       $\alpha_{12} = \alpha_{12} \cup \alpha, P_{12} = P_{12} \cup P, m_{12} = m_{12} \cup m$ 
13:       $N_{12} = N_{12} + 1$ 
14:    end if
15:  end for
16: end for
17: end for
18:  $w_{12}^{(n)} = \frac{\alpha_{12}^{(n)}}{\sum_k \alpha_{12}^{(k)}} (\sum_{i=1}^{N_1} w_1^{(i)})^{\mathcal{W}} (\sum_{j=1}^{N_2} w_2^{(j)})^{1-\mathcal{W}},$  for  $n =$ 
    1, ...,  $N_{12}$  {Scale weights using weighted cardinalities of the two sets}
19: for  $i \in I \setminus I_F \wedge j \in J \setminus J_F$  do
20:    $m_{12} = m_{12} \cup m_1^{(i)} \cup m_2^{(j)}$ 
21:    $P_{12} = P_{12} \cup P_1^{(i)} \cup P_2^{(j)}$ 
22:    $w_{12} = w_{12} \cup w_1^{(i)} \cup w_2^{(j)}$ 
23: end for
24: return  $\{m_{12}^{(n)}, P_{12}^{(n)}, w_{12}^{(n)}\}_{n=1}^{N_{12}}$ 

```

state space is a combination of kinematic and shape parametric variables

$$\mathbf{x} = [x, y, \nu, \theta, \omega, L, W]^\top \quad (21)$$

where x and y are the Euclidean coordinates of the center of the rectangle, ν is the speed, θ denotes the heading angle and ω is the turn rate. Length and width of the rectangle are denoted by L and W , respectively. For state propagation we have selected a constant turn rate and velocity (CTRV) motion model, in literature also known as coordinated turn with polar velocity (see for example [22]–[24]), which we have augmented with object dimensions, assumed to be constant. The dynamic state equation is given by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + G(\mathbf{x}_k)\xi_k \quad (22)$$

where

$$f(\mathbf{x}) = \begin{bmatrix} x + \left(\frac{2\nu}{\omega}\right) \sin\left(\frac{\omega t}{2}\right) \cos\left(\theta + \frac{\omega t}{2}\right) \\ y + \left(\frac{2\nu}{\omega}\right) \sin\left(\frac{\omega t}{2}\right) \sin\left(\theta + \frac{\omega t}{2}\right) \\ \nu \\ \theta + \omega t \\ \omega \\ L \\ W \end{bmatrix} \quad (23)$$

$$G = \begin{bmatrix} \frac{t^2}{2} \cos(\theta) & \frac{t^2}{2} \sin(\theta) & t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{t^2}{2} & t & 0 & 0 \end{bmatrix}^\top \quad (24)$$

$$\xi = [a \quad \alpha]^\top = \left[\frac{dv}{dt} \quad \frac{d\omega}{dt} \right]^\top \quad (25)$$

Input noise on linear and rotation acceleration a and α are treated as uncorrelated random variables with zero mean and standard deviations σ_a m/s² and σ_α rad/s², and t represents time.

Require: $\{[r^{(i)}, \beta^{(i)}]^\top, R_{r,\beta}^{(i)}\}_{i=1}^{N_{pts}}$ {set of range & bearing measurement points}

- 1: **for** each i **do** {augment points with Cartesian coordinates}
- 2: $([x^{(i)}, y^{(i)}]^\top)^\top = \text{Polar-to-Cartesian}([r^{(i)}, \beta^{(i)}]^\top)$
- 3: **end for**
- 4: clusters = DBSCAN($[x, y]^\top$)
- 5: **for** each cluster in clusters **do**
- 6: Fit a line
- 7: Fit a corner (corner point is point with largest distance to the line connecting first and last measurement point)
- 8: Choose a fit (line or corner) with smaller RMS error
- 9: Fit a rectangle (with constant L, W) to a chosen line/corner
- 10: Extract rectangle features x, y, θ
- 11: Construct measurement covariance $R_{x,y,\theta}$
- 12: **end for**
- 13: **return** $\{[x^{(j)}, y^{(j)}, \theta^{(j)}]^\top, R_{x,y,\theta}^{(j)}\}_{j=1}^{N_{obj}}$ {set of object observations}

B. Measurement Model

At each time step, a lidar returns a set of measurement points (a point cloud). Each point is defined in polar coordinates in 2D (using range r and bearing β), and the uncertainty is expressed through the covariance matrix $R = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_\beta \end{bmatrix}$. It is assumed that range and bearing measurements are mutually independent. The point cloud is preprocessed using an object detection algorithm (Algorithm II) in order to obtain measurements of a form $\mathbf{z} = [x, y, \theta]^\top$, where one measurement corresponds to what is believed to be one object (x and y represent the center of the object rectangle, and θ its orientation). The PHD intensity is updated using the UKF and object measurements \mathbf{z} .

Object detection is accomplished in several steps: (i) conversion of measurement points from polar to Cartesian coordinates, (ii) clustering using the DBSCAN algorithm [25], while ignoring clusters with less than two points, (iii) line and corner fitting to each of the clusters, and (iv) rectangle fitting and feature point extraction. A processed measurement contains the center and orientation of one rectangle. The measurement noise (including the noise of the detection algorithm) in the observation space $[x, y, \theta]^\top$ is evaluated empirically in simulation using object ground truth data.

Measurements of rectangle center are directly used to update x and y variables of the state vector \mathbf{x} . The rectangle orientation has a π rad ambiguity (from a static lidar scan one cannot determine object heading but only its orientation). Therefore, before updating the heading $\theta_{\mathbf{x}}$ in the state vector, we adapt the orientation measurement $\theta_{\mathbf{z}}$ such that the difference between the orientation measurement and the state heading is wrapped to the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$:

$$\begin{aligned} \delta &= \text{mod}(\theta_{\mathbf{z}} - \theta_{\mathbf{x}} + \frac{\pi}{2}, \pi) - \frac{\pi}{2} \\ \theta_{\mathbf{z}} &= \theta_{\mathbf{x}} + \delta \end{aligned} \quad (26)$$

C. Occlusion and FOV Model

The aim of the occlusion and FOV model is to enable tracking of objects that are temporarily occluded or are located at the boundaries of the sensing FOV. This is achieved by setting the probability of detection $p_{D,k}(m_{k|k-1}^{(i)})$ to a low value for targets that are not likely to be detected by a sensor. If this is not performed, the C-GM-PHD filter would quickly decrease weights of those targets and discard them.

We adapt the occlusion model of Granstrom et al. [26] by verifying occlusions for the two rectangle corners (with

Require: Object of interest \mathbf{x} , set of all object estimates $\{\hat{\mathbf{x}}^{(i)}\}_{i=1}^N$ with weights $w^{(i)}$, sensor range r_s and FOV β_s

- 1: **Gaussian kernel:** $g(x; \mu, \sigma) = 0.5 \exp(-((x - \mu)/\sigma)^2)$
- 2: $p_D = 0, \sigma_{r_s} = 1\text{m}, \sigma_{\beta_s} = 0.25\text{deg}, \sigma_\beta = 1.5\text{deg}$
- 3: Compute range and bearing to object corners: $r_j, \beta_j, j = 1, 2, 3, 4$
- 4: Minimum/maximum bearings: $j_- = \arg \min_j \beta_j; j_+ = \arg \max_j \beta_j$
- 5: Mean range: $r_-^\pm = 0.5(r_{j_+} + r_{j_-})$
- 6: **Object at the boundary of the FOV:**
- 7: **if** $-\beta_s/2 \leq \beta_{j_-} \leq \beta_s/2$ **then**
- 8: $p_D = p_D + (0.5 - g(\beta_{j_-}; -\beta_s/2, \sigma_{\beta_s}) - g(\beta_{j_-}; \beta_s/2, \sigma_{\beta_s}))$
- 9: **end if**
- 10: **if** $-\beta_s/2 \leq \beta_{j_+} \leq \beta_s/2$ **then**
- 11: $p_D = p_D + (0.5 - g(\beta_{j_+}; -\beta_s/2, \sigma_{\beta_s}) - g(\beta_{j_+}; \beta_s/2, \sigma_{\beta_s}))$
- 12: **end if**
- 13: Enforce maximum probability of detection: $p_D = \min\{p_D, p_D^{max}\}$
- 14: **if** $r_-^\pm \geq r_s$ **then**
- 15: $p_D = 0$
- 16: **else**
- 17: $p_D = p_D - 2g(r_-^\pm; r_s, \sigma_{r_s})$
- 18: **end if**
- 19: **Object occluded:**
- 20: **for** $i = 1, \dots, N$ **do**
- 21: Range/bearing to corners of estimate: $\hat{r}_k^{(i)}, \hat{\beta}_k^{(i)}, k = 1, 2, 3, 4$
- 22: Minimum/maximum bearings: $k_- = \arg \min_k \hat{\beta}_k, k_+ = \arg \max_k \hat{\beta}_k$
- 23: Mean range: $\hat{r}_-^\pm = 0.5(\hat{r}_{k_+}^{(i)} + \hat{r}_{k_-}^{(i)})$
- 24: **if** $\hat{r}_-^\pm < r_-^\pm$ **and** $\hat{\beta}_{k_-}^{(i)} \leq \beta_{j_-} \leq \hat{\beta}_{k_+}^{(i)}$ **then**
- 25: $p_D = p_D - (0.5 - g(\beta_{j_-}; \hat{\beta}_{k_-}^{(i)}, \sigma_\beta) - g(\beta_{j_-}; \hat{\beta}_{k_+}^{(i)}, \sigma_\beta))$
- 26: **end if**
- 27: **if** $\hat{r}_-^\pm < r_+^\pm$ **and** $\hat{\beta}_{k_-}^{(i)} \leq \beta_{j_+} \leq \hat{\beta}_{k_+}^{(i)}$ **then**
- 28: $p_D = p_D - (0.5 - g(\beta_{j_+}; \hat{\beta}_{k_-}^{(i)}, \sigma_\beta) - g(\beta_{j_+}; \hat{\beta}_{k_+}^{(i)}, \sigma_\beta))$
- 29: **end if**
- 30: Enforce minimum probability of detection: $p_D = \max\{p_D, p_D^{min}\}$
- 31: **end for**
- 32: **return** Probability of detection p_D for object of interest

the smallest and the largest bearing angle), and augment it with FOV verification. Occlusions are verified against all other objects that are estimated to be in-between the sensor and the object of interest. The decrease in probability of detection is proportional to the weight of the estimate. One occluded corner decreases p_D by maximum 0.5. Gaussian kernels are used to smooth the transition from visible to occluded area. To keep track of targets that are outside of a sensor's FOV (through collaboration), we place the Gaussian kernels along the boundary of the sensor's FOV. They decrease p_D for targets that are at the boundary or outside of the FOV. The algorithm is described in Algorithm III.

IV. EXPERIMENTS

Experiments are performed in Webots [27], a submicroscopic high-fidelity robotic simulator that we have recently updated with a number of automotive features¹. We use simulated Citroën C-ZERO cars that are created using a CAD model of the real C-ZERO and calibrated using the real car. Each car used for sensing is equipped with a front-facing Ibeo LUX lidar, a GNSS device, a compass and a wireless communication device. Resolution, accuracy, range, and other sensors' characteristics are calibrated using sensors' data sheets. This setup allows us to reproduce the real world realistically. Fig. 3 shows a screenshot of our simulation environment and simulated cars and sensors.

Our dataset contains the data we collected during 20 distinctive simulation runs. In each run, vehicles ran an

¹See <http://disal.epfl.ch/RO2IVSim>

open-loop controller following pre-defined trajectories. Actuator and sensor (e.g., lidar) noise contribute to the randomness of the generated sensing data.

We then run the proposed C-GM-PHD filter in Matlab, feeding it with measurements from the dataset. We vary the localization accuracy by adding Gaussian noise with zero mean and standard deviation (SD) of $\sigma_l \in \{0, 0.5, 1\}$ to outputs of GNSS and compass devices (in m and deg, respectively). We also vary the communication rate relatively to the sensing rate. In particular, we consider $\mathcal{T} \in \{1, 2, 5\}$, with a communication rate of $\mathcal{T} = n$ meaning that information is exchanged between cars each n samplings of the lidars. The lidar sampling rate is instead constant (80 ms) and corresponds to a simulation step in Webots. For the sake of simplicity, there is no package loss and no communication delay in our current implementation (but dealing with communication delays is straightforward when tracks and motion models are shared).

The tracking parameters were the same in all runs. The maximum number of Gaussian components per filter J_{max} is limited to 30. The birth model for the GM-PHD filter is a sum of eight Gaussian components with weight 0.001 located around the sensory FOV boundary. The birth speed is 8 m/s, the heading is normal to the FOV boundary with direction towards inside of the FOV, and the turn rate is set to 0.025 rad/s. The SD for a birth component is 50 m for the lateral and 30 m for the longitudinal position, 6 m/s for the speed, π rad for the heading, and 0.2 rad/s for the turn rate. A new target entering the sensing FOV is very likely to be contained within one SD from one of the birth component means. The SD for the noise in the motion model is set to $\sigma_a = 1$ m/s² and $\sigma_\alpha = 0.1$ rad/s². The measurement noise is determined empirically and is modeled as a Gaussian with SD $\sigma_x = 2$ m, $\sigma_y = 2$ m, and $\sigma_\theta = 0.78$ rad. The clutter model uses a Poisson distribution with an expected cardinality of 10 measurements per sensor surveillance area. The merging parameter $U = 6$ and a Gaussian component is pruned if its weight is less than $T_p = 10^{-5}$. The extraction threshold T_e is set to 0.5. The fusion distance parameter U_F is empirically set to 50. The Ibeo LUX lidar has a FOV of 110 deg and a range of 200 m. The probability of survival in the joint FOV is set to 0.99. The probabilities of detection are set as follows: $p_D^{max} = 0.98$ and $p_D^{min} = 0.02$. The shape of the object is assumed to be rectangular, with length of 3.5 m and width of 1.5 m.

The literature suggests optimization methods for choosing a good value for the CI fusion weight \mathcal{W} [16], as well as approximate solutions based on the fused covariance trace or determinant minimization [28]. However, Uney et al. [29] suggest that fixing \mathcal{W} at 0.5 produces near optimal results on average. Therefore, the value for \mathcal{W} is set to 0.5 in our experiments.

A. Scenarios²

The first scenario includes one stationary ego (\mathcal{E}), one stationary collaborative (\mathcal{C}), and three moving non-collaborative cars ($\mathcal{NC}_i, i \in \{1, 2, 3\}$) in an open space. Two sensing cars are sufficient to showcase the proposed method, although the method supports multiple sensing cars. Cars \mathcal{E} and \mathcal{C} have a forward-facing lidar, localization sensors (a GNSS and a

²A video showing the two scenarios can be viewed at <http://disal.epfl.ch/NetworkedIV>

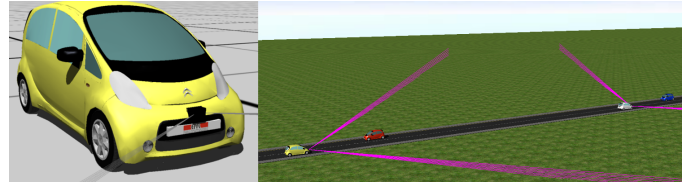


Fig. 3. A screenshot from the Webots simulator. A Citroën C-ZERO car with an Ibeo LUX lidar in the front (left) and a road segment scenario with multiple cars (right).

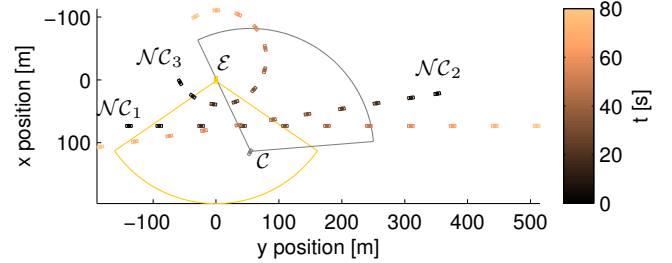


Fig. 4. Ground truth trajectories shown in the local frame of \mathcal{E} . Time is color coded, being black at 0 s and orange at 80 s. Cars \mathcal{NC}_1 and \mathcal{NC}_2 move on a straight line, while \mathcal{NC}_3 moves with rotational motion. Cars \mathcal{E} and \mathcal{C} and their fields of view are shown in yellow and gray, respectively. Note: car rectangles are scaled up 3x for visibility reasons.

compass), and a wireless communication device on-board. Car \mathcal{C} sends its PHD intensity augmented by its own state to \mathcal{E} . \mathcal{NC} cars do not have any sensors and do not share any information. Poses and trajectories of cars are described in Fig. 4.

The second scenario includes \mathcal{E} , \mathcal{C} and seven \mathcal{NC} cars that are driving on a straight road segment. Due to limited space, we provide results for the first scenario only, which is more general and is valid for most real-world situations. Both scenarios are shown in the accompanying video. Given that this paper focuses on car tracking, the scenarios contain no other objects. Tracking is performed in the local frame of \mathcal{E} .

B. Results

We evaluate the performance of the C-GM-PHD filter by comparing its output to the ground truth data. For multi-object performance evaluation we use the Optimal SubPattern Assignment (OSPA) metric [30]. OSPA is comprised of two components, the first accounting for localization and the second for cardinality errors of the estimates. We choose the sensitivity of the metric in penalizing estimated position $p = 2$, and the cut-off parameter for cardinality errors penalties $c = 60$. The distance metric in OSPA is the Euclidean distance between estimated and ground truth position.

Fig. 5 displays the OSPA error for a single run of C-GM-PHD filter, for $\sigma_l = 0$ and $\mathcal{T} = 1$. Fig. 6 shows the localization and cardinality OSPA components. The target set cardinality was correctly estimated 91% of time. Short periods of wrong cardinality estimation, as shown in Fig. 6, are the result of the filter delay (multiple consecutive observations of a new object are needed for the filter to give birth to a new target), transitions between the FOV of \mathcal{E} and \mathcal{C} vehicles, and occlusions. When a target goes from the FOV of only one filter to the common FOV, the C-GM-PHD filter might lose track of it for a

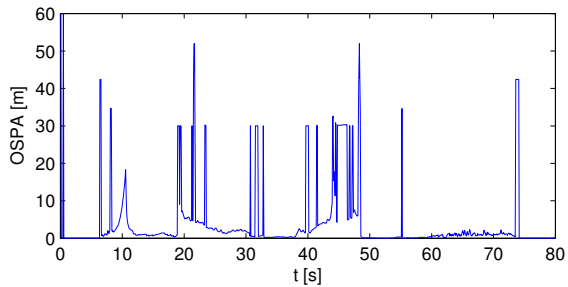


Fig. 5. OSPA distance versus time for a single run.

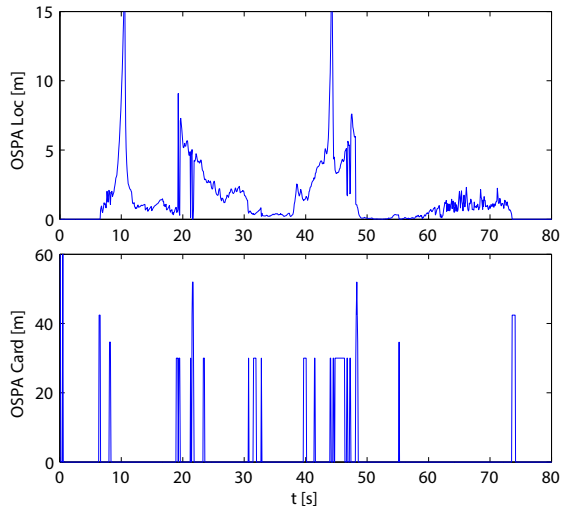


Fig. 6. Separate OSPA components for a single run. (Top) localization component versus time; (Bottom) cardinality component versus time.

short moment. This is due to fusing two Gaussian components, one with $w \approx 1$ and the second one with a low weight (e.g., target in the birth phase). For a chosen value of $\mathcal{W} = 0.5$, CI gives equal weights to the two Gaussian components, hence the weight of the fused component may get lower than the extraction threshold $T_e = 0.5$. Notably, an optimization of \mathcal{W} is needed to correctly tackle this issue. The occlusion and FOV model presented in Sec. III-C is dependent on the targets' position estimate error, and it happens that a detectable target is assigned low p_D or vice-versa. This can as well happen when the target is near to the boundaries of the FOV. Nevertheless, it is important to note that all wrong cardinality estimates represent false negatives, which is desirable when driver assistance systems are considered. The DBSCAN clustering algorithm discards all clusters with less than two measurements. In cases when two or more clutter measurements fall next to each other, the PHD filter successfully filters them out. Statistics over 20 runs for all combinations of σ_l and \mathcal{T} is given in Fig. 7. As expected, OSPA error increases with localization error and decreases with higher communication rate.

The estimated position of car $\mathcal{N}\mathcal{C}_1$ is shown in Fig. 8, together with its associated uncertainty, for $\sigma_l = 1$ and $\mathcal{T} = 5$. Uncertainty is notably higher at around $t = 10$ s, when $\mathcal{N}\mathcal{C}_1$ was occluded and therefore no measurement-based corrections were applied. After $t = 30$ s, $\mathcal{N}\mathcal{C}_1$ leaves the FOV of car \mathcal{E} , and is thus only observable by car \mathcal{C} . The estimated position uncertainty is significantly higher in this region due

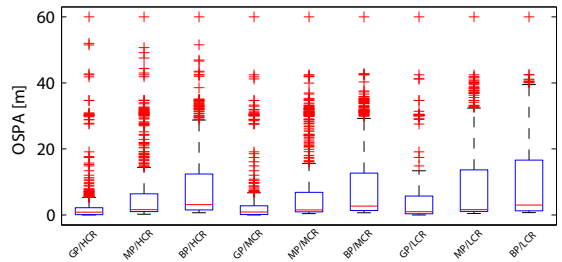


Fig. 7. OSPA statistics for 20 runs, for various values of localization (GNSS and compass) white noise standard deviation σ_l , and communication rate \mathcal{T} parameters. GP, MP, BP stand for good, medium and bad positioning, i.e., $\sigma_l = 0, 0.5$ and 1 , respectively. HCR, MCR and LCR stand for high, medium and low communication rate, i.e., $\mathcal{T} = 1, 2$ and 5 , respectively.

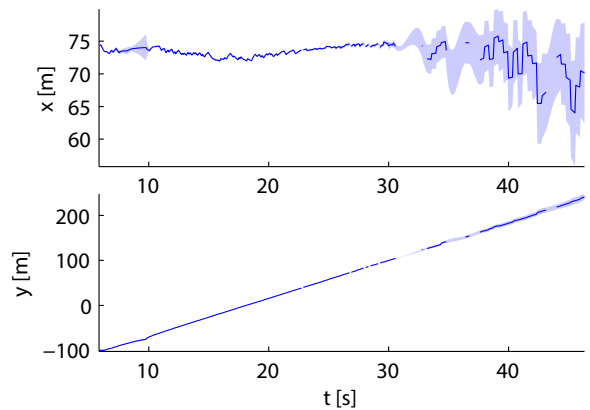


Fig. 8. The estimated trajectory of target $\mathcal{N}\mathcal{C}_1$ with associated uncertainty, for a single run with parameters $\sigma_l = 1$ and $\mathcal{T} = 5$. For periods in which the target is lost (e.g., at around $t = 32$ s), the estimated trajectory is not drawn, whereas the uncertainty is interpolated.

to localization uncertainties of cars \mathcal{C} and \mathcal{E} , as expected.

The performance of the C-GM-PHD filter is further compared to the performance of the non-collaborative GM-PHD filter, and the result is shown in Fig. 9. A fair comparison can only be done in the FOV of \mathcal{E} (because a non-collaborative filter does not track objects outside of this area). Thus, ground truth targets are considered only in this region, and the C-GM-PHD filter is modified to perform the collaborative fusion only if targets are found in the FOV of \mathcal{E} . We observe that, thanks to collaboration, the C-GM-PHD filter needs less time to start tracking the first target, \mathcal{C} . With the exception of two peaks (moments when the C-GM-PHD briefly loses targets), the C-GM-PHD filter performs consistently better or equal than its non-collaborative version. The total tracking time of all objects is listed in Table I. We can observe the added value of the extended FOV and filter reactivity that is offered by the C-GM-PHD filter.

V. CONCLUSION

We have presented a complete approach for tracking multiple vehicles using lidars, based on the GM-PHD filter. Moreover, we have proposed a general approach for fusion of GM-PHD intensities whose definition domains do not completely overlap. The two contributions together form a novel C-GM-PHD filter, that is capable of tracking cars in the FOV of its ego sensor, as well as tracking targets outside

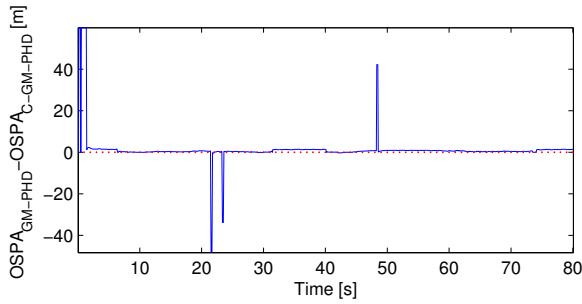


Fig. 9. Difference between a non-collaborative GM-PHD filter and a C-GM-PHD filter expressed using OSPA. Red line is a marker at zero.

TABLE I. TOTAL TRACKING TIME PER TARGET

	\mathcal{C}	\mathcal{NC}_1	\mathcal{NC}_2	\mathcal{NC}_3
GM-PHD	78.7 s	24.1 s	18.6 s	33.5 s
C-GM-PHD	79.9 s	37.9 s	47.3 s	53.5 s

of its FOV by using communication as a remote sensor. We exemplify the method using cars and one lidar per car. The approach is evaluated in simulation and it shows promising results: most of the time, a C-GM-PHD filter yields a more accurate estimate than its non-collaborative version, with the added value that the ego vehicle is no longer limited to tracking targets that are in its FOV.

To further improve the performance of the proposed C-GM-PHD filter, a GCI weight optimization for the Poisson target cardinality will be investigated. Moreover, using a cardinalized PHD filter could improve cardinality estimates, as it would drop the assumption that the cardinality is Poisson distributed. To further validate the proposed approach, we plan to use real sensors, real GNSS devices, and real Citroën C-ZERO cars, and compare the results with the simulation. We will consider in the future a generalization of our approach for more complex scenarios including pedestrians, trees, etc.

REFERENCES

- [1] G. Thomaidis, K. Vassilis, P. Lytrivis, M. Tsogas, G. Karaseitanidis, and A. Amditis, "Target tracking and fusion in vehicular networks," in *IEEE Intelligent Vehicles Symposium*, Jun. 2011, pp. 1080–1085.
- [2] M. Obst, L. Hobert, and P. Reisdorf, "Multi-sensor data fusion for checking plausibility of V2V communications by vision-based multiple-object tracking," in *IEEE Vehicular Networking Conference*, Dec. 2014, pp. 143–150.
- [3] R. MacLachlan and C. Mertz, "Tracking of moving objects from a moving vehicle using a scanning laser rangefinder," in *IEEE Intelligent Transportation Systems Conference*, Sep. 2006, pp. 301–306.
- [4] Y. Bar-Shalom, *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional, Inc., 1987.
- [5] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, Inc., 2007.
- [6] B.-N. Vo, A. Pasha, and H. D. Tuan, "A Gaussian Mixture PHD filter for nonlinear jump Markov models," in *IEEE Conference on Decision and Control*, Dec. 2006, pp. 3162–3167.
- [7] B. Ristic, D. Clark, and B.-N. Vo, "Improved SMC implementation of the PHD filter," in *13th Conference on Information Fusion*, Jul. 2010. doi: 10.1109/ICIF.2010.5711922
- [8] K. Granstrom, C. Lundquist, and U. Orguner, "A Gaussian mixture PHD filter for extended target tracking," in *13th Conference on Information Fusion*, Jul. 2010. doi: 10.1109/ICIF.2010.5711885
- [9] K. Granstrom, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-Mixture PHD filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3268–3286, Oct. 2012.
- [10] K. Granstrom and C. Lundquist, "On the use of multiple measurement models for extended target tracking," in *16th International Conference on Information Fusion*, Jul. 2013, pp. 1534–1541.
- [11] K. Granstrom, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *14th International Conference on Information Fusion*, Jul. 2011, pp. 592–599.
- [12] A. Swain and D. Clark, "The PHD filter for extended target tracking with estimable extent shape parameters of varying size," in *15th International Conference on Information Fusion*, Jul. 2012, pp. 1111–1118.
- [13] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2X-based perception in a high-level fusion architecture for cooperative perception systems," in *IEEE Intelligent Vehicles Symposium*, Jun. 2012, pp. 270–275.
- [14] C. Olaverri-Monreal, P. Gomes, R. Fernandes, F. Vieira, and M. Ferreira, "The see-through system: A VANET-enabled assistant for overtaking maneuvers," in *IEEE Intelligent Vehicles Symposium*, Jun. 2010, pp. 123–128.
- [15] D. Clark, S. Julier, R. Mahler, and B. Ristic, "Robust multi-object sensor fusion with unknown correlations," in *Sensor Signal Processing for Defence*, Sep. 2010. doi: 10.1049/ic.2010.0233
- [16] M. Uney, D. Clark, and S. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 521–531, Jun. 2013.
- [17] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, Jun. 2013.
- [18] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, Dec. 1986.
- [19] J. K. Uhlmann, "General data fusion for estimates with unknown cross covariances," in *Signal Processing, Sensor Fusion, and Target Recognition V*, vol. 2755, 1996, pp. 536–547.
- [20] R. P. S. Mahler, "Optimal/robust distributed data fusion: a unified approach," in *Signal Processing, Sensor Fusion, and Target Recognition IX*, vol. 4052, 2000, pp. 128–138.
- [21] S. Julier, "An empirical study into the use of chernoff information for robust, distributed fusion of gaussian mixture models," in *9th International Conference on Information Fusion*, Jul. 2006. doi: 10.1109/ICIF.2006.301755
- [22] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *11th International Conference on Information Fusion*, Jun. 2008. doi: 10.1.1.476.5164
- [23] X. Yuan, F. Lian, and C. Han, "Models and algorithms for tracking target with coordinated turn motion," *Mathematical Problems in Engineering*, vol. 2014, Jan. 2014. doi: 10.1155/2014/649276
- [24] M. Roth, G. Hendebly, and F. Gustafsson, "EKF/UKF maneuvering target tracking using coordinated turn models with polar/Cartesian velocity," in *17th International Conference on Information Fusion*, Jul. 2014.
- [25] M. Ester, H.-P. Kriegel, J. S., and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, USA, 1996, pp. 226–231.
- [26] K. Granstrom, S. Reuter, D. Meissner, and A. Scheel, "A multiple model PHD approach to tracking of cars under an assumed rectangular shape," in *17th International Conference on Information Fusion*, Jul. 2014.
- [27] O. Michel, "Webots: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.
- [28] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," in *8th International Conference on Information Fusion*, vol. 1, Jul. 2005, pp. 154–160.
- [29] M. Uney, S. Julier, D. Clark, and B. Ristic, "Monte Carlo realisation of a distributed multi-object fusion algorithm," in *Sensor Signal Processing for Defence*, Sep. 2010. doi: 10.1049/ic.2010.0232
- [30] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.