IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 July 2022, accepted 27 July 2022, date of publication 8 August 2022, date of current version 17 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197665

TOPICAL REVIEW

Multiagent Information Fusion for Connected Driving: A Review

JAMES KLUPACS, AMIRALI KHODADADIAN GOSTAR, THARINDU RATHNAYAKE, IQBAL GONDAL, ALIREZA BAB-HADIASHAR, (Senior Member, IEEE), AND REZA HOSEINNEZHAD

Royal Melbourne Institute of Technology, Melbourne, VIC 3000, Australia

Corresponding author: Amirali Khodadadian Gostar (amirali.khodadadian@rmit.edu.au)

This work was supported by the Australian Research Council under Grant DE210101181.

ABSTRACT This paper reviews the state-of-the-art multi-sensor fusion approaches applicable in the next-generation intelligent transportation systems where connected vehicles are cooperatively driven for maximum safety and efficiency. The review finds out that complementary sensor fusion in a time-varying distributed network is required, and for such applications, the state-of-the-art is sensor fusion in the random finite set filtering framework. The fundamental bases of random finite set filters are reviewed with more elaboration on a particular filter called the Labeled Multi-Bernoulli filter. An information-theoretic approach for data fusion based on minimizing information divergence between statistical densities is presented, along with how different divergence functions can be used for sensor fusion. Different approaches are evaluated for their tracking performance and computational cost in a realistic simulation scenario. Their advantages, and disadvantages in the context of real-time implementation in a connected driving scenario are discussed.

INDEX TERMS Random finite sets, intelligent transport systems, multi-object tracking, information fusion.

I. INTRODUCTION

Connected devices are increasingly making the world around us smarter, safer, and more efficient. The world of transportation and driving is no different. Connected vehicles can help us avoid obstacles, reduce risks on the road and make the driving experience more enjoyable. Having its own connection to the internet, a connected vehicle shares data with other devices around it. This makes it possible for vehicles connected in a centralized or distributed network to share their sensory information with each other.

Through the integration of all the information received from onboard sensors with those from neighboring vehicles, a connected vehicle can achieve a more accurate and comprehensive situational awareness. This can take multiple forms such as multi-modal information fusion from multiple onboard sensors such as audio sensors and video capturing devices or external GPS signals combined with onboard localization devices to obtain more accurate localization results. Finally, information from multiple vehicles may be combined such as video from opposite ends of a road to gain a more complete awareness of the surroundings, known as *complementary fusion*, which is the main focus of this paper.

This will effectively contribute to increased intelligence of the connected vehicle in making various trajectory planning and local maneuvering actions, whether it is part of an advanced driver-assist capability of the vehicle or its own self-driving capability. In this context, efficient multi-sensor data fusion is an intrinsic part of the design of any intelligent transportation system (ITS) that involves connected vehicles.

Sensor fusion solutions have attracted strong interest in multi-vehicle applications within the ITS domain [1]. Several methods have been proposed to fuse the information gathered by a relatively large number of sensors in a multi-vehicle network [2]. These include different information fusion techniques and metrics for the main types of network topologies.

Although a single vehicle could be configured to have 360° sensing capabilities through radar, lidar or camera that provide information in the immediate surroundings, events outside the sensors' field-of-view (FoV) range cannot be

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

captured and could potentially lead to catastrophic outcomes. The most straightforward approach to solve this problem is to fuse the information acquired by several agents together in a *complementary* manner through vehicles working together cooperatively. An example is*vehicle platooning* where information can be shared between mobile agents through vehicles working together cooperatively [3].

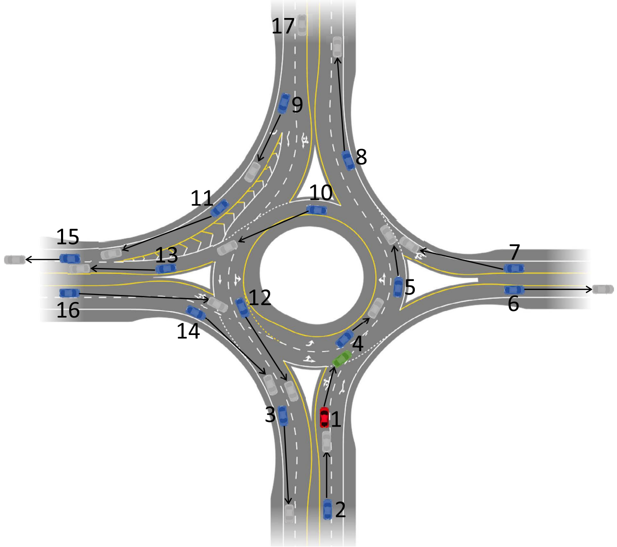


FIGURE 1. An example of connected vehicles and their movements in an urban road. Initial positions are represented by blue with their future positions in gray. This highlights a potential use case for cooperative fusion because in order for each vehicle to successful enter the roundabout it must know the positions of oncoming vehicles.

To further clarify the necessity of the integration of the information fusion method in an ITS system that includes connected vehicles, consider the scenario shown in Fig. 1. In this scenario, there are a total of 16 vehicles at $k = 1$, labeled by their numbers, and colored in red and blue. As time passes, at $k = 100$, the vehicles are displaced to locations shown in gray and green. Also during $k = 1 : 100$, Vehicles '6' and '15' exit, and a new vehicle '17' enters.

The vehicles communicate their sensor information in a "sensor network" and each sensor mounted on the vehicle has a limited FoV. For example, in the scenario shown in Fig. 1, at time $k = 1$, the FoV of the sensor mounted on node '1' cannot cover all the targets in the FoV of the sensor mounted on node '4'. Thus, communication and fusion of the information received at nodes is a necessary step for each vehicle to get the complete view of the environment.

Importantly, the topology of the sensor network evolves with time and the connection between nodes is continuously changing. This is shown in Fig. 2. Initially, vehicle/sensor node '1' is connected to nodes '2', '3' and '4'. After a while, at $k = 100$, connections change and node '1' is now connected to nodes '12' and '14' as well. This happens to all nodes. In addition, some nodes (corresponding to vehicles exiting the scene) disappear and some (corresponding to vehicles entering the scene) appear and are added to the network. In the case shown in Fig. 1 and Fig. 2, '6' and '15' disappear and '17' appears with connections to nodes '8' and '9'.

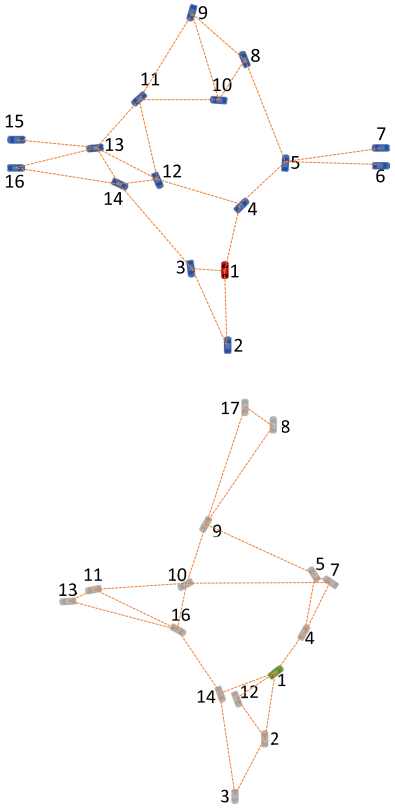


FIGURE 2. Evolution of the sensor network associated with the urban road condition shown in Fig. 1 for two different times. Top: network at $k = 1$. Bottom: network at $k = 100$.

The above example clearly demonstrates the essence of effective solutions for fusion of multiple sources of information in intelligent transportation systems where vehicles are connected through a data network. The solution must be applicable where the sensors have different fields-of-view and the network topology dynamically varies with time.

There are three common categories of sensor network topology: *centralized*, *decentralized*, and *distributed* sensor networks [4], [5], [6]. As illustrated in Fig. 3(a), in a centralized network, all local information are transmitted to a central node for further fusion processing. Communication through a central node enables easy implementation; however, it becomes infeasible as the number of the nodes (sensors or vehicles) increases. Farhadi *et al.* [7] have analyzed the computational overhead of two algorithms used in their application of automated irrigation networks. It was found that the centralized network's overhead grew by $\mathcal{O}(n^5)$ for fixed n sub systems, i.e agents. In contrast the distributed version only grew linearly by $\mathcal{O}(n)$.

A decentralized network (Fig. 3(b)), aims to overcome the above issue by utilizing multiple fusion centers to communicate with their neighbors. This decreases the system's vulnerability to node failure; however, decentralized networks still suffer from exponential growth of computational burden as the number of the nodes increases.

Finally, in a distributed sensor network, as shown in Fig. 3(c), each sensor node performs fusion using only the information received from locally connected nodes. Distributed sensor networks provide flexibility and scalability, as each node only communicates with its neighboring nodes. In an ITS scenario where vehicles could number in the hundreds for small areas, such an approach is appropriate and necessary to achieve real-time functionality, as demonstrated by the linear cost scaling in [7].

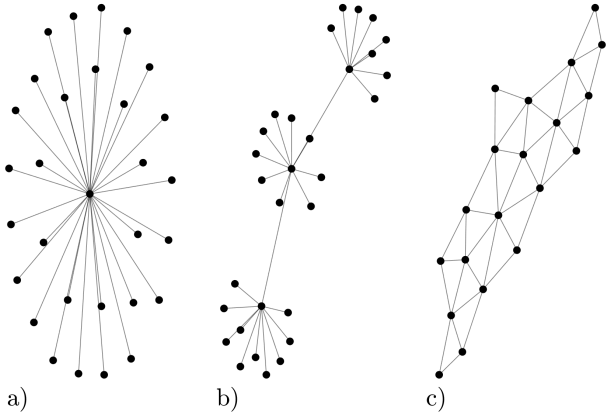


FIGURE 3. Illustration of common categorization of sensor network topologies: a) centralized, b) decentralized, c) distributed.

In many advanced sensor fusion algorithms, it is not the raw sensory data that are communicated between vehicles through the network, but a particular form of data that represents vehicle's perception about the environment. Hence, the overall process is sometimes called *cooperative perception*. In its common form, vehicle perception is formulated as a multi-object distribution (encapsulating information about the statistics of vehicles' and other objects' states such as location, speed, orientation, yaw rate, and so on). In this case, it is the distribution parameters that are communicated and fused in a Bayesian framework.

Multiple methods exist for information fusion in the Bayesian framework, ranging from Kalman filters and more recent probabilistic methods such as random finite set-based (RFS) filters [8], [9], [10], [11], [12].

Alternative approaches towards forming a perception by each connected vehicle include using belief functions (and fusing them via believe propagation or Dempster-Shafer reasoning theory), using scans directly as perception model (and fusing via scan matching) or occupancy grids maps (and fusing them via map merging) or high-level fuzzy models that can be fused via a fuzzy rule-base.

While several recent works have reviewed the state-of-the-art fusion methods for autonomous and connected vehicles [13], [14], these works have either focused on sensor specific fusion methods without considering a specific scenario or have considered a high-level overview of each fusion strategy equally, again with an emphasis on the fusion of different sensor types. A similarly structured paper [15] investigates data fusion for traffic flow applications. This paper looks at the specific case of cooperative perception, or information fusion from complementary features coming from multiple agents.

A key consideration of connected vehicles in ITS scenarios is security attacks and other vulnerabilities such as intentional jamming. Although RFS methods have an inherently low level tolerance to jamming, due to their formulation of false alarms as clutter, these methods are outside of the scope of this paper and will not be considered.

Similarly, heterogeneous sensor networks such as those found in connected vehicles have several challenges which must be considered when designing a solution. The three main issues as discussed in [16] are bandwidth considerations for sensors, namely what type of information is transmitted, is it naturally compressible such as RFS densities, and is a sensor going to bottleneck due to large amounts of clutter, track management between sensors, and most importantly sensor trust.

The main contributions of this paper is to firstly provide an overview of the current methods for information fusion, which have been directly applied to cooperative vehicle networks where the sensor has limited FoV. Furthermore, a comprehensive review of current information fusion methods within the RFS framework has been conducted. Secondly, this paper presents two comparison studies using three fusion methods used within the RFS framework, where the FoV is limited in each sensor.

The rest of the paper is structured as follows. Section II will review the literature towards constructing a categorization of the various recent solutions. Section III provides a more in-depth review of the statistical data fusion methods in which the locally processed sensor data are communicated in the form of statistical densities. The main focus is on introducing the general Bayesian filtering approach to information fusion and more specifically, the RFS filters to enable the reader to understand various divergence-based and complementary fusion methods which are discussed in section IV. Network implementation of the fusion methods are reviewed in section V, followed by results and discussions from realistic cooperative drive simulations that are presented in section VI. Section VII concludes the paper.

II. OVERVIEW AND CATEGORIZATION

Multi-sensor fusion solutions that have been applied to ITS are based on using various estimation and information fusion methods, however most fusion methods focus on the problem of multi-modality, not fusion between vehicles. The most common examples of such methods are Kalman Filters [8], [17], [18], [19], neural networks [26], Bayesian decision theory [27], Dempster-Shafer evidential reasoning [21], Blackboard Architecture [28], and fuzzy logic [22].

The majority of the papers listed in Table 1 do not use a RFS based tracking method for target detections. Instead,

TABLE 1. A summary of sensor fusion implementations that have been applied cooperative driving applications.

Method Ref.	Algorithm Type	Fusion Method	Network Architecture
Liu et al. (2017) [17]	Kalman Filter	Cubature Kalman Filter	Distributed
Karam et al. (2006) [8]	Kalman Filter	Extended Kalman Filter	Distributed
Kianfar et al. (2012) [18]	Kalman Filter	Extended Kalman Filter	Distributed
Li et al. (2013) [19]	Kalman Filter	Split Covariance Intersection	Decentralized
Brambilla et al. (2020) [9]	Belief Propagation	Implicit Cooperative Positioning	Distributed
Fröle et al. (2018) [10]	Bayesian Filter	Kalman Filter	Centralized
Merdrignac et al. (2017) [20]	Multiple Hypothesis Tracker	Multiple Hypothesis Tracker	Distributed
Radak et al. (2016) [21]	Dempster-Shafer Theory	Dempster-Shafer Theory	Distributed
Milanes et al. (2010) [22]	Fuzzy Logic	Fuzzy Logic Decision Algorithm	Distributed
Wang et al. (2020) [23]	PHD	Generalized Approx. Message Passing	Distributed & Centralized
Li et al. (2014) [24]	Occupancy Grid	Occupancy Grid Map Merging	Distributed
Kim et al. (2015) [25]	Scan Matching	Correlative Scan Matching (CSM)	Distributed

they rely on other stochastic methods such as Kalman filters. These methods are typically used for single-object detection only, as the methods do not natively support multi object detection, and must be done heuristically in such cases. In contrast, RFS methods are able to natively track multiple objects in areas with high clutter and measurement false alarms.

Table 1 summarizes the most common methods recently reported for sensor fusion between vehicles within a cooperative driving scheme. For each method, the network architecture (distributed, centralized or decentralized), and the algorithm family and the fusion method are specified.

A. STOCHASTIC METHODS

Liu*et al.* [17] built upon principles outlined in [29], [30], [31] using the Cubature Kalman Filter (CKF) to fuse dedicated short-range communications (DSRC) with global navigation satellite systems (GNSS). The authors adapted CKF by introducing a local Huber-based technique, improving the CKF, and employing an adaptive strategy for the restraint factor. The algorithm was implemented for a dynamic traffic simulation using data from google maps.

Karam *et al.* [8] presented a state exchange based cooperative localization method to overcome the large quantity of data exchange presented by other methods [32], [33], [34]. This algorithm is implemented using Extended Kalman Filter (EKF). A simulation featuring three vehicles was employed. The results show that when the vehicles communicated and shared location information, the error was less than the GPS standard deviation.

Kianfar *et al.* [18] developed a cooperative driving algorithm for the Grand Cooperative Driving Challenge (GCDC) [35]. The vehicle needed to cooperatively drive in a platoon in two different scenarios. These consisted of an urban scenario where the vehicles had to cross an intersection in a platoon following a green light signal. The other scenario was a highway scenario where acceleration shock waves were introduced. To complete these scenarios, the vehicle made use of local sensor data and Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication. Information was fused using an extended Kalman filter (EKF), while the platoon leader state was estimated using a conventional Kalman filter (KF). Real-time sensor fusion results were achieved using this method.

Li and Nashashibi [19] proposed a new method of cooperative vehicle positioning, using a fusion method known as the Split Covariance Intersection Filter (SCIF), part of the covariance intersection filter family [36], which has been implemented using a KF for state updates. In this work, the authors demonstrate that a centralized fusion scheme was infeasible for real-world ITS, so a decentralized method had to be adopted. However, decentralized networks suffer from inter-estimate correlation. Covariance Intersection Filter (CIF) is a method which essentially removes the risk of over convergence, where estimates converge to incorrect values, due to inter-estimate correlation. This method built upon the work done in [37], [38] and was tested using synthetic data, where the implemented scenario was a platoon of eight vehicles in single file formation. Each vehicle could only communicate with its nearest neighbors, *i.e.* the vehicle in front and behind itself. The SCIF was compared to three different scenarios: Single Vehicle Localization (SL), Naïve Cooperative Localization (NCL) and State Exchange Based Cooperative Localization methods (SECL). It was shown that the proposed SCIF method consistently performed better than the others, able to match the other methods when all vehicles have low positioning accuracy and outperform when the ego vehicle has high positioning accuracy (0.1 m in the test).

Brambilla *et al.* [9] extended a V2V positioning algorithm known as Implicit Cooperative Positioning [39] to incorporate data association (DA), which differs from previous works [40], [41]. The authors propose a low complexity sub-optimal method using *hard* Maximum-A-Posteriori Bayesian detection and a Kalman filter for cooperative localization. The aim is to use GNSS information to augment communication between vehicles in order to improve localization performance. Vehicles must detect a set of common passive objects within the environment, which can then be shared through inter-vehicle communication methods. This method was found to outperform conventional GNSS methods.

Fröle *et al.* [10] examined the case of uncertain sensor states, which happens when the sensor is attached to a moving object. The proposed filter was based on a Bayesian filter modified to limit the data associated with each step and was a continuation of the work proposed in [42]. The state was updated using a Kalman filter. Position measurements were obtained from a GNSS, while feature detection was obtained from a stereo vision camera. When birth and death of pedestrian objects were introduced, the proposed multi object tracking filter was able to outperform the track-oriented marginal multiple target multi-Bernoulli/Poisson (TOMB/P) filter, however, detections were considered as point objects.

Merdrignac *et al.* [20] used the Multiple Hypothesis Tracker (MHT) to fuse vehicle perception data with Vehicle to Pedestrian (V2P) data. The work was an extension of the authors' previous works [43], [44] Vehicle perception was handled by laser data and was used for the detection and classification of objects which lie inside of the FoV of the sensor. Vulnerable Road User (VRU) data is obtained via GPS data and is transmitted to surrounding vehicles. Fusion is achieved by weighing the information consisting of object position from both systems and implementing the option with the highest probability for each object. Both lines of sight (LOS) and non-line of sight (NLOS) scenarios were tested. By using V2P communication it was found that a 100% True positive rate (TPR) for pedestrian detections was able to be achieved using the Receiving Operator Characteristic (ROC). The fusion error was found to be zero up to 60 m and was consistently better than either of the two systems individually. However, it was noted that performance in real-world scenarios would be worse. It was found that for distances over 70m, the V2P communication system works better, however, below this distance the perception system is superior.

Radak *et al.* [21] made use of the algorithm presented [45] for use in an icy road detection scenario. The algorithm used was a self-stabilizing distributed data fusion algorithm based on Dempster-Shafer theory [45]. Each vehicle calculated a basic belief, represented by a mass function, using information from the neighboring vehicles and roadside units (RSU).

Milanes *et al.* [22] used fuzzy logic derived from earlier work [46] to control two vehicles in an intersection scenario. Cooperative maneuvers based on multiple sensors have previously been explored in [47], [48], [49]. By using V2V communications, the position and velocity of each vehicle can be determined. Fuzzy logic then controls the vehicles to maneuver correctly through the intersection.

Wang and Jiang [23] proposed a method of 3D cooperative localization using a Bayesian Probability Hypothesis Density (PHD) filter. As the PDFs are approximated by Gaussian messages, the localization problem can be categorized as a generalized linear mixing problem and can be solved by importance sampling. The algorithm has been expanded to incorporate NLOS measurements.

B. DETERMINISTIC METHODS

Li *et al.* [24] implemented a method to fuse V2V information using an objective function based occupancy grid [50] map merging method first adopted in [51]. Each vehicle obtains its occupancy grid via a Simultaneous localization and mapping (SLAM) algorithm. Using the map merging method outlined in the paper, the occupancy maps belonging to vehicles A and B can be merged together. 1155 pairs of local occupancy grids were generated by the test vehicles and tested using the proposed method. Each pair was iterated until the evolution result was within 20 cm in position and 0.5° in orientation around the ground truth. It was found that the average convergence evolution number was 5. With the 3Ghz processor used, each genetic evolution took around 70ms. Therefore the average time taken was 0.35s. The proposed method outperformed the previous Iterative Closest Point (ICP) based method proposed in [52]

Kim *et al.* [25] have used two different methods for map fusion for vehicle platooning, using methods outlined in [53], [54]. The authors use set up several experiments using a Mitsubishi iMiEV and three Yamaha golf carts. The sensors used 2D lidars and a RGB vision camera. Various communication interfaces were used such as WiFi and 4G LTE. The vehicles were set up in single formation, such that the second and third vehicles in the formation (*i+1* and *i+2*) respectively could perceive information regarding the vehicles ahead, which was otherwise outside the vehicle's normal FoV. Two implementations for map fusion were used: ICP and a probabilistic scan matching method is known as Correlative Scan Matching (CSM). In order to map vision data into the spatial coordinates of the ego vehicle, Inverse Perspective Matching (IPM) was used. It was found that CSM was able to provide the lowest error for both translation and rotation for the first leader. The time taken was far greater than either ICP or lidar at 108ms. For the second leader, CSM was superior to ICP. However, there was a far smaller increase in performance at the cost of computation time.

III. BACKGROUND AND INFORMATION

A. STATISTICAL INFORMATION FUSION FOR CONNECTED DRIVING

As it was mentioned previously, sensor fusion methods applied to connected driving scenarios are mostly devised in a statistical information fusion framework. In this framework, the objects of interest (both road objects and other surrounding objects) are represented via their states, and it is the statistics of those states that are updated as the result of sensor fusion. The most common approach is to use Bayes' rule for fusing sensor data while updating the probability density of objects' state.

1) BAYESIAN FILTER

Let us assume that the information about an object is described by a state vector x_k at each time k. We assume that a sensor indirectly observes the object state via a noisy measurement vector z_k. The filter sequentially estimates the state x_k of the dynamic system given the measurement history $z_{1:k} \equiv (z_1, \ldots, z_k)$.

The time evolution of the object's state vector, also called its *motion model* is described by a stochastic model in the form of a Markov transition

$$x_k = t_k(x_{k-1}, v_k), \tag{1}$$

which specifies the transformation of any given state vector x_{k-1} at time $k-1$ and system noise v_k at time k into a new state vector x_k at time k.

Alternatively, the time evolution of the state vector is described by a *Markov transition density* $f_{k|k-1}(\cdot|\cdot)$ where $f_{k|k-1}(x_k|x_{k-1})$ is the probability density that the state x_{k-1} transitions to the state x_k at time k. In a Kalman filter, a linear motion model is assumed

$$x_k = F_{k-1}x_{k-1} + v_k, \tag{2}$$

where v_k is assumed to be normally distributed with zero mean and covariance Q_k. Therefore, the motion model is described with the Markov transition density

$$f_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; F_{k-1}x_{k-1}, Q_k). \tag{3}$$

A *sensor model* is described by an observation equation

$$z_k = h_k(x_k, w_k) \tag{4}$$

which specifies how at time k, the measurement acquired by the sensor, z_k, is related to the state vector of the object x_k with the measurement noise incorporated by including a random variable w_k. Alternatively, the sensor model is described by a likelihood function $g_k(z_k|x_k)$ which is the probability density that at time k, the state x_k generates the measurement vector z_k.

In the classical form of Kalman filter, the sensor model is linear with additive zero-mean Gaussian noise,

$$z_k = H_k x_k + w_k, \tag{5}$$

where $w_k \sim \mathcal{N}(0, R_k)$. This leads to the likelihood function

$$g_k(z_k|x_k) = \mathcal{N}(z_k; H_k x_k, R_k). \tag{6}$$

In the Bayesian filter, the object's state density is recursively propagated from time $k-1$ to k, through two steps: prediction and update. At each time k, the prior density $\pi_{k-1}(\cdot)$ is propagated to the posterior $\pi_k(\cdot)$ which will be applied as the prior in the next time $k+1$.

In the prediction step, Chapman-Kolmogorov equation is used as follows

$$\pi_{k|k-1}(\cdot) = \int f_{k|k-1}(\cdot|x)\pi_{k-1}(x)dx \tag{7}$$

where $\pi_{k|k-1}(\cdot)$ is called the *predicted* density of the object. It is then used in the update step which is formulated based on Bayes' rule:

$$\pi_k(\cdot) = \frac{g_k(z_k|\cdot)p_{k|k-1}(\cdot)}{\int g_k(z_k|x)p_{k|k-1}(x)dx}. \tag{8}$$

In the traditional Kalman filter, the prior and posterior are assumed to be Gaussians with means m_{k-1} and m_k and covariance matrices P_{k-1} and P_k. Substitution of the motion and likelihood models into the predtion and update equations leads to the Kalman filter equations:

$$m_k = F_{k-1}\,m_{k-1} + K_k[z_k - H_k F_{k-1}\,m_{k-1}] \tag{9}$$

$$P_k = [I - K_k H_k]\left(Q_{k-1} + F_{k-1}P_{k-1}F_{k-1}^{\top}\right) \tag{10}$$

where

$$K_k = \left(Q_{k-1} + F_{k-1}P_{k-1}F_{k-1}^{\top}\right)H_k^{\top}S_k^{-1}$$

and

$$S_k = R_k + H_k\left(Q_{k-1} + F_{k-1}P_{k-1}F_{k-1}^{\top}\right)H_k^{\top}.$$

The extended Kalman filter is similar to above, with a difference that non-linearities are approximated by first-order linear functions.

The above general formulation and specific cases for Kalman and extended Kalman filter are mainly applicable for a single-object filter. In the connected driving application, we are dealing with multiple (if not numerous) objects of interest, including vehicles and surrounding objects. Hence, a multi-object filter is needed.

The most common approach to using a Bayesian filtering for multiple objects is to combine the multiple object state vectors by stacking them into a long vector. Sensor fusion is also performed in a similar manner: multiple sensor measurements of the same type are stacked on top of each other into a long measurement vector. The Bayesian filter then needs to solve the *data association* problem, i.e. to determine which object state is corresponding to which measurement.

There have been a large number of works dedicated to devise and implement efficient solutions in the above approach. The multiple hypothesis tracker (MHT) is one of the best solutions proposed so far. However, this approach is naturally applicable when the number of objects and sensors are *previously* known and do not change randomly with time. As it was mentioned earlier (see the example depicted in Fig. 1, in a connected driving application, the number of vehicles can vary with time due to new vehicles entering and some exiting the scene. In addition, for each vehicle, the number of sensors communicating their information can change with time (see Fig. 2).

A natural remedy for statistical multi-sensor fusion in presence of uncertain and time-varying number of objects and measurements, is to use *finite set statistics* as outlined by Mahler [55]. In this framework, instead of stacking multiple object states and measurement vectors into longer vectors, they are treated as *random finite sets*. An RFS is a random variable in the form of a set in which not only the elements of the set but also its *cardinality* (the number of elements in the set) vary randomly.

Note that Mahler [55] has demonstrated that fuzzy logic-based and belief propagation-based methods for information fusion have theoretically equivalent forms formulated in the RFS framework.

2) RFS FILTER

Before we review the formulation of a generic multi-object filter in the RFS framework, we need to present the notation that is used throughout this paper. A single-object state is denoted by lower-case letters (e.g. $\boldsymbol{x}$ and x), multi-object state (set) is represented using upper-case letters (e.g. X and $\boldsymbol{X}$), spaces are shown by blackboard bold letters (e.g. $\mathbb{N}$, $\mathbb{X}$ and $\mathbb{L}$). Bold-face letters (e.g. $\boldsymbol{X}$ and $\boldsymbol{x}$) represent labeled states. The density of a labeled RFS is also bold-face (e.g. $\boldsymbol{\pi}(\cdot)$). Furthermore, the standard inner product notation is denoted by $\langle f, g\rangle \triangleq \int f(x)g(x)dx$. The L^2-norm notation is denoted by $\|f\| \triangleq \sqrt{\langle f, f\rangle}$. The number of elements in a set X is called its *cardinality* and denoted by $|X|$.

With a RFS filter in place, what is communicated through the network (in a centralized or distributed manner) is the parameters of the RFS density from each source (connected vehicle) after the locally acquired measurements are effected through a local update step. It is then the RFS densities that are fused. Such a fusion is normally devised based on minimizing some sort of distance or *divergence* between the fused density and those being fused. In this paper, we will review fusion methods based on the Kullback-Leibler divergence (KLD) (denoted by $D_{\mathrm{KL}}(\cdot,\cdot)$) and Cauchy-Schwarz divergence (CSD) (denoted by $D_{\mathrm{CS}}(\cdot,\cdot)$) between two RFS densities.

The rest of symbols are similar to the single-object Bayesian filter but the arguments of functions can be sets instead of vectors. For instance, the multi-object motion model and measurement likelihood functions are denoted by $f_{k|k-1}(X_k|X_{k-1})$ and $g(Z_k|X_k)$, respectively. Table 2 summarizes the notation for easy access.

TABLE 2. Basic notation used for RFS filter formulation.

Notation	Definition
x	single object state
X	multi-object state (a set)
Z_k	a set of measurements acquired at time k
$\pi(\cdot)$	a multi-object RFS density
$\boldsymbol{\pi}(\cdot)$	a labeled RFS density
$\lvert\cdot\rvert$	cardinality of an RFS
$\langle\cdot,\cdot\rangle$	inner product of two functions
$D_{\mathrm{CS}}(\cdot,\cdot)$	Cauchy-Schwarz divergence between two densities
$D_{\mathrm{KL}}(\cdot,\cdot)$	Kullback-Leibler divergence between two densities

In finite set statistics, the notions of integration and density for RFSs are defined as follows. Let us denote the space of single-object state x, by $\mathbb{X}$. A RFS

$$X = \{x_1, \cdots, x_n\}$$

can then vary in the space of all finite subsets of $\mathbb{X}$ denoted by $\mathcal{F}(\mathbb{X})$. The set integral of a function of RFSs, $f(\cdot)$, over any subset space

$$\mathcal{T} \subseteq \mathcal{F}(\mathbb{X})$$

is given by:

$$\int_{\mathcal{T}} f(\mathbf{X})\delta\mathbf{X} = \sum_{i=0}^{\infty}\frac{1}{i!}\int_{\mathbb{X}^i\cap\mathcal{T}} f(\{x_1,\ldots,x_i\})d(x_1,\ldots,x_i). \tag{11}$$

The density of a RFS is defined as a function $\pi(X)$ that satisfies:

$$\Pr(\mathcal{T}) = \int_{\mathcal{T}} \pi(X)\delta X, \quad \forall\mathcal{T} \subseteq \mathcal{F}(\mathbb{X}) \tag{12}$$

Based on the above definitions, we can then apply Chapman-Kolmogorov equation and Bayes' rule for the prediction and update steps of the multi-object Bayesian filter in the RFS framework:

$$\pi_{k|k-1}(\cdot) = \int f_{k|k-1}(\cdot|X)\pi_{k-1}(X)\delta X \tag{13}$$

$$\pi_k(\cdot|Z_k) = \frac{g(Z_k|\cdot)\pi_{k|k-1}(\cdot)}{\int g(Z_k|X)\pi_{k|k-1}(X)\delta X}. \tag{14}$$

It is important to note that object entry (birth) and exit (death) are naturally modeled by incorporating the following parameters within the multi-object motion model:

- probability of survival, $p_S(x)$, which models object exit points; it is close to 1 around those regions of $\mathbb{X}$ where a single-object can disappear (no longer survives from $k-1$ to k); and
- a birth model in the form of a separate RFS which is the union of possibly existing objects with their probabilities of existence and densities (conditioned on existence) are larger in those regions of $\mathbb{X}$ where new objects can enter the scene.

Furthermore, extra uncertainties become possible to be modeled by incorporating the following parameters into multi-object measurement likelihood:

- probability of detection, $p_D(x)$, which models the sensor's detection constraints due to limited FoV or environmental conditions (e.g. brightness or temperature); and
- possible false alarms that are modeled as a separate RFS of false alarms appended to the measurement set Z_k received from the sensor.

How the above parameters are incorporated into the multi-object motion model and likelihood function depends on the implementation of the RFS filter based on making assumptions on particular mathematical forms of multi-object densities. Indeed, similar to Kalman filters where the single-object prior and posterior densities are assumed Gaussian, such particular assumptions for the density of the RFS prior lead to implementations such as the Probability Hypothesis Density (PHD) filter, the multi-Bernoulli filter, the labeled multi-Bernoulli (LMB) filter and the Generalized labeled multi-Bernoulli (GLMB) filter.

B. RFS FILTER IMPLEMENTATIONS

As it was mentioned earlier, in order to devise numerically tractable implementations of the general Bayesian RFS filter, different particular RFS density models have been devised.

This section briefly reviews those models and the underlying theory of the resulting filters.

1) PHD AND CPHD FILTERS

A particular form of RFS density, called *Poisson* RFS is defined as a random set of finite independent and identically distributed (i.i.d.) with the cardinality being Poisson distributed. A Poisson RFS is completely characterized with its *intensity function* (also called its PHD). The PHD is defined as a function $\nu : \mathbb{X} \to \mathbb{R}^+ \cup \{0\}$ having which, the statistical mean of cardinality will be given by $\mathbb{E}(|X|) = N_X = \int_{\mathbb{X}} \nu(x)dx$. In addition, the identical density of each element of the set will be given by $\nu(\cdot)/N_X$.

Applying a Poisson density approximation, a PHD filter is derived in which, instead of the whole multi-object density, the PHD is propagated from a prior $\nu_{k-1}(\cdot)$ to a predicted PHD $\nu_{k|k-1}(\cdot)$ through the prediction step, then to a PHD posterior $\nu_k(\cdot)$ through the update step [56], [57], [58].

In many applications, enforcing a Poisson density on cardinality leads to biased estimates. Therefore, this is relaxed by simply assuming that the multi-object RFS has is an *i.i.d. cluster*, which means its elements are i.i.d. with the same density $p(\cdot) = \nu(\cdot)/\int \nu(x)\,dx$ but the cardinality has a particular discreet distribution $\rho(\cdot)$ which is not necessarily Poisson. Similar to PHD filter, the resulting filter that is called *Cardinalized* PHD filter, or CPHD filter for short, propagates a prior i.i.d. RFS density characterized by $\left(\rho_{k-1}(\cdot), \nu_{k-1}(\cdot)\right)$ to a predicted i.i.d. RFS density $\left(\rho_{k|k-1}(\cdot), \nu_{k|k-1}(\cdot)\right)$ through the prediction step, then to a posterior i.i.d. density $\left(\rho_k(\cdot), \nu_k(\cdot)\right)$ through the update step [57], [59].

2) MULTIBERNOULLI FILTER

The multi-Bernoulli RFS density is an approximation that explicitly models possible existence and state density of objects in the state space $\mathbb{X}$. In this approximation, the multi-object density is entirely characterized by the maximum number of possible objects, M, and the probability of existence $r^{(m)}$ as well as the density $p^{(m)}(\cdot)$ of each possibly existing object, where $m = 1, \ldots, M$.

In a multi-Bernoulli filter, at each time, k, the prior is a multi-Bernoulli denoted by

$$\pi_{k-1}(\cdot) \sim \left\{\left(r_{k-1}^{(m)}, p_{k-1}^{(m)}(\cdot)\right)\right\}_{m=1}^{M_{k-1}}$$

which is propagated through the prediction step to

$$\pi_{k|k-1}(\cdot) \sim \left\{\left(r_{k|k-1}^{(m)}, p_{k|k-1}^{(m)}(\cdot)\right)\right\}_{m=1}^{M_k}$$

then through the update step to the posterior

$$\pi_k(\cdot) \sim \left\{\left(r_k^{(m)}, p_k^{(m)}(\cdot)\right)\right\}_{m=1}^{M_k}.$$

Details of prediction and update equations can be found in [60].

3) LABELED MULTIBERNOULLI FILTER

The labeled multi Bernoulli filter (LMB) [61] is *labeled RFS filter* and a generalization of the multi Bernoulli filter. It can output object tracks and has been shown to have no cardinality bias and to be able to cope with low signal-to-noise ratios (SNR).

In labeled RFS filters, each single object state is appended with a particular label that remains unchanged during the object's existence. Indeed, each single-object state is denoted by $\mathbf{x} = (x, \ell)$ and The space of labels of all possibly existing objects at each time k is denoted by $\mathbb{L}_k$.

Similar to the multi-Bernoulli filter, in an LMB filter, at time k, the prior is entirely characterized by probabilities of existence and state densities of all possibly existing object labels,

$$\boldsymbol{\pi}_{k-1}(\cdot) \sim \left\{\left(r_{k-1}^{(\ell)}, p_{k-1}^{(\ell)}(\cdot)\right)\right\}_{\ell \in \mathbb{L}_{k-1}}.$$

The density parameters are then predicted to

$$\boldsymbol{\pi}_{k|k-1}(\cdot) \sim \left\{\left(r_{k|k-1}^{(\ell)}, p_{k|k-1}^{(\ell)}(\cdot)\right)\right\}_{\ell \in \mathbb{L}_k}$$

and updated to the posterior

$$\boldsymbol{\pi}_k(\cdot) \sim \left\{\left(r_k^{(\ell)}, p_k^{(\ell)}(\cdot)\right)\right\}_{\ell \in \mathbb{L}_k}.$$

The LMB filter is among the most sophisticated multi-object filters and will be used in our comparative simulations. Therefore, we elaborate on its parameters in more detail as follows.

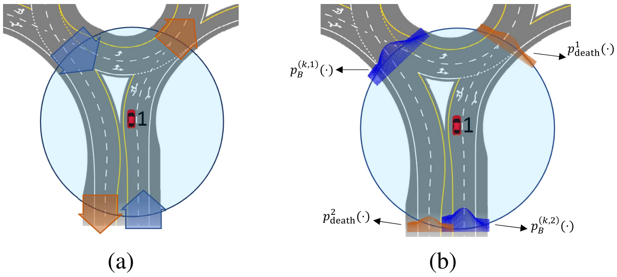
As it was mentioned earlier, the prediction step incorporates all the information available about possible random changes that can occur to each object state (based on a *motion model*), as well as possible entry of new objects into the sensor's FoV (birth) or some objects exiting it (death). At each time, the possibly born objects are modeled as an LMB denoted by

$$\boldsymbol{\pi}_B = \{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{B}_k}$$

where $\mathbb{B}_k$ is the space of new object labels at time k. In practice, the densities $p_B^{(\ell)}(\cdot)$ have their peaks around those possible areas of object entry to the sensor's FoV. Since, there are usually more than one such areas, the birth LMB normally has more than one Bernoulli component.

The possible disappearance of objects from the sensor's FoV is modeled via a state-dependent probability of survival function, denoted by $p_S(x, \ell)$. It is a probability (between 0 and 1) for continuing survival of an object with state x and label ℓ from time $k-1$ to k.

To demonstrate how the road information is incorporated into the LMB filter via the probability of survival and the birth model, an example is presented in Fig. 4 based on the scenario shown in Fig. 1 but focused on Vehicle number 1. From road information (boundaries and directions of vehicle travel), in the example shown, the birth LMB can have two components, each with a small probability of existence (which can change with time depending on the traffic reports), and a Gaussian density centered around the possible areas of object entry into Vehicle 1 sensor's FoV. Those densities are shown in Fig. 4(b) as $p_B^{(k,1)}(\cdot)$ and $p_B^{(k,2)}(\cdot)$ where $\ell_1 = (k, 1)$ and $\ell_2 = (k, 2)$ are the labels with two components: k which is the time of birth, and 1 or 2 which is just an index to differentiate between objects that are born at the same time. The probability of survival, however, is a probability and in this example, is modeled as $p_S(\cdot) = 1 - p^1_{\text{death}}(\cdot) - p^2_{\text{death}}(\cdot)$ where the two death probabilities are simply Gaussian exponential elements (without the normalization factors - so ranging between 0 and 1) centered around the possible exit regions.



(a)
(b)

FIGURE 4. An example of how the birth and death elements are incorporated into the LMB prediction, based on road and sensor FoV information: (a) the possible areas of object entry and exit for vehicle 1 (b) the associated Gaussian components to be included in the birth LMB and the $p_S(\cdot)$ function.

The update step of the LMB filter incorporates the information acquired from measurements. The sensor's detection profile information is also used here. This information includes the probability of detection, $p_D(\cdot)$, which is normally close to 1 within the sensor's FoV and close to zero around its borders, and zero beyond those borders. We also model the false alarms (wrong detections that do not associate to any existing object) as a Poisson RFS with its intensity function denoted as $\kappa(z)$ where $z \in \mathbb{Z}$ and $\mathbb{Z}$ is the space of possible measurements.

For details of the prediction and update steps, see [61].

IV. SENSOR FUSION WITH RFS FILTERS

Consider a vehicle V_0 that is connected to multiple vehicles $V_1, \ldots, V_N$ in a cooperative driving application. Each vehicle in the connected network is equipped with sensors and in its local processing unit, a RFS filter is running, through which a multi-object density (representing the vehicle's perception of its surrounding) is propagated through time. The locally acquired sensor information is utilized by the local filter in its update step.

At time k, let us denote the local multi-object posteriors computed in each vehicle node i, by $\boldsymbol{\pi}_{i,k}$. Each vehicle V_0 receives all the locally computed posteriors from its connected neighboring vehicles, $V_1, \ldots, V_N$. An example is shown in Fig. 5 which is consistent with the scenario presented in Figs. 1 and 2(top).

In the general statistical information fusion framework, it is the communicated multi-object densities that are fused. Indeed, in each vehicle node, e.g. V_0, the locally acquired posterior and all those received from other vehicles are fused, noting that each of those local posteriors have incorporated information from their neighbours (through the Bayesian update step of the local filter running in each vehicle).

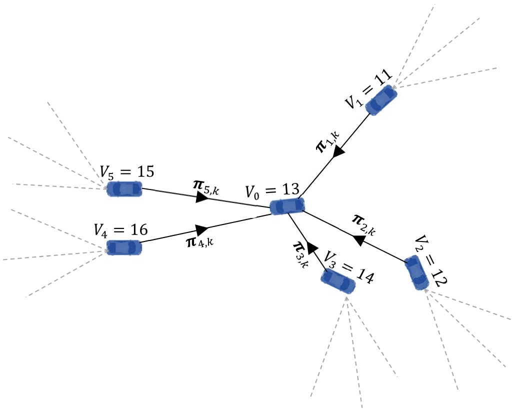


FIGURE 5. An example of how a connected vehicle receives locally computed multi-object posteriors from surrounding vehicles in a cooperative driving scenario. Note that vehicle numbers are compatible with part of the example shown in Fig. 1 and Fig. 2 (top).

The most common approach for fusion of multiple densities is to meet the requirements of the *Principle of Minimum Discrimination Information* (PMDI) according to which the probability density which best represents the current state of knowledge is the one which produces an *information gain* as small as possible. Note that from a control perspective it is desirable to maximize gain for control algorithms to have the most information to work with, however from a fusion perspective all must be treated equally with a fused result which is as similar to all fused information as possible, minimizing the information gain. Information gain is usually quantified in terms of some form of divergence between two densities. Hence, the fused multi-object density $\boldsymbol{\pi}_{\text{fused},k}$ should have the minimum divergence from the fusing densities. Let us assume that all vehicles V_i connected to the ego vehicle V_0 and itself are assigned each an importance weight ω_i, and the weights are normalized $\sum_{i=0}^{N} \omega_i = 1$. Then the fused density at the ego vehicle at time k, denoted by $\bar{\boldsymbol{\pi}}_{0,k}$, is the one that minimizes the weighted sum of all information gains measured in terms of divergences,

$$\bar{\boldsymbol{\pi}}_{0,k} = \underset{\boldsymbol{\pi}}{\text{argmin}} \sum_{i=0}^{N} \omega_i D(\boldsymbol{\pi} || \boldsymbol{\pi}_{i,k}) \tag{15}$$

where $D(\boldsymbol{\pi} || \boldsymbol{\pi}_{i,k})$ denotes the divergence or information gain from $\boldsymbol{\pi}_{i,k}$ to $\boldsymbol{\pi}$.

A. KULLBACK-LEIBLER DIVERGENCE-BASED FUSION

In Bayesian statistics, the KLD is well-known to represent the *information gain* achieved when moving from a prior to a posterior. The KLD from a density $\boldsymbol{\pi}_{i,k}$ to $\boldsymbol{\pi}$ is defined by:

$$D_{\text{KL}}\left(\boldsymbol{\pi} \,||\, \boldsymbol{\pi}_{i,k}\right) \triangleq \int \boldsymbol{\pi}(\boldsymbol{X}) \log \frac{\boldsymbol{\pi}(\boldsymbol{X})}{\boldsymbol{\pi}_{i,k}(\boldsymbol{X})} \delta \boldsymbol{X}. \tag{16}$$

For any particular filter (PHD, CPHD, or LMB), the parametric form of the locally acquired posteriors, $\boldsymbol{\pi}_{i,k}$'s, and the fused posterior, $\bar{\boldsymbol{\pi}}_{0,k}$, can be substituted in the above equation then in (15), and the parameters of the fused posterior calculated via optimization. For example, if each local posterior is a CPHD with parameters $\big(\rho_{i,k}(\cdot), \nu_{i,k}(\cdot)\big)$ then given the importance weights

$$\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N),$$

the fused CPHD posterior, that minimizes the weighted sum of KLDs from all local posteriors, turns out to be characterized by: [62]

$$\bar{\nu}_k(x; \boldsymbol{\omega}) = \prod_{i=1}^{N} [\nu_{i,k}(x)]^{\omega_i} \Big/ \int \prod_{i=1}^{N} [\nu_{i,k}(x')]^{\omega_i} \, dx' \quad (17)$$

$$\bar{\rho}_k(n; \boldsymbol{\omega}) = \bar{q}_k(n; \boldsymbol{\omega}) \Big/ \sum_{n'=1}^{\infty} \bar{q}_k(n'; \boldsymbol{\omega}) \quad (18)$$

where

$$\bar{q}_k(n; \boldsymbol{\omega}) = \left[\prod_{i=1}^{N} [\rho_{i,k}(n)]^{\omega_i}\right] \left\{\int \prod_{i=1}^{N} \left[p_{i,k}(x)\right]^{\omega_i} \, dx\right\}^n$$

and $p_{i,k}(x) = \nu_{i,k}(x) / \int \nu_{i,k}(x') \, dx'$.

In a separate scenario, consider that an LMB filter is operating in each vehicle node. For the ego vehicle V_0, its local posterior is calculated and the parameters are denoted by

$$\boldsymbol{\pi}_{0,k}(\cdot) \sim \left\{\left(r_{0,k}^{(\ell)}, p_{0,k}^{(\ell)}(\cdot)\right)\right\}_{\ell \in \mathbb{L}_k}.$$

The ego vehicle also receives the parameters of all the other vehicles connected to it, which are denoted by

$$\boldsymbol{\pi}_{i,k}(\cdot) \sim \left\{\left(r_{i,k}^{(\ell)}, p_{i,k}^{(\ell)}(\cdot)\right)\right\}_{\ell \in \mathbb{L}_k}. \quad i = 1, \ldots, N$$

Substituting the above LMB densities in (16), and solving the optimization problem (15), gives the following closed-form for the fused parameters: [63]

$$\bar{r}_k^{(\ell)}(\boldsymbol{\omega}) = \frac{\int_{\mathbb{X}} \prod_{i=0}^{N} \left[r_{i,k}^{(\ell)} p_{i,k}^{(\ell)}(x)\right]^{\omega_i} dx}{\int_{\mathbb{X}} \prod_{i=0}^{N} \left[r_{i,k}^{(\ell)} p_{i,k}^{(\ell)}(x)\right]^{\omega_i} dx + \prod_{i=0}^{N} \left[1 - r_{i,k}^{(\ell)}\right]^{\omega_i}}$$

$$\bar{p}_k^{(\ell)}(\cdot; \boldsymbol{\omega}) = \frac{\prod_{i=0}^{N} \left[p_{i,k}^{(\ell)}(\cdot)\right]^{\omega_i}}{\int_{\mathbb{X}} \prod_{i=0}^{N} \left[p_{i,k}^{(\ell)}(x)\right]^{\omega_i} dx} \quad (19)$$

An important aspect of employing KLD as the divergence from one density to another is the "geometric averaging" nature of the fusion rule that is achieved. This makes the resulting fusion rule very sensitive to miss-detection due to limited FoV of sensors on vehicles. To clarify this limitation, consider an existing object A that can be another vehicle, or any other object required to be included as part of the ego vehicle's perception of the environment, with its state in some sub-region $\mathbb{X}_A$ of the state space $\mathbb{X}$ (i.e. $\mathbb{X}_A \subset \mathbb{A}$). Let us assume that due to its sensors' limited FoV, the vehicle V_i does not detect A. Hence, its local PHD, $\nu_{i,k}(\cdot)$ will be close to zero in the sub-region $\mathbb{X})A$. From equation (17), it is obvious that the resulting fused PHD, $\bar{\nu}_k(\cdot; \boldsymbol{\omega})$ will be very small in $\mathbb{X}_A$ too, even though the other sensors may have detected the object A and have relatively large PHD in that sub-region.

A similar issue occurs when LMB filters are operating on vehicles with limited FoV. If an object is out of one vehicle's FoV and its corresponding probability of existence is zero, then from equation (19), the fused probability of existence for that object becomes zero as well. KLD-based fusion appears to be unable to directly handle multiple sensors with different FoVs.

B. CAUCHY-SCHWARZ DIVERGENCE-BASED FUSION

To derive a fusion rule that incorporates all the complementary information acquired by various vehicles (and indeed by their sensors which have different FoVs), a recent remedy has been proposed by Gostar *et al.* [11], [64]. Inspired by the simple and intuitive, yet mathematically solid derivation of the *Cauchy-Schwarz* divergence between two Poisson processes [65], they suggested to use the CSD as the divergence of choice.

The CSD between two densities can be interpreted as the cosine of the angle subtended by the two density functions in the space of square-integrable functions [11]. The CSD between a density $\boldsymbol{\pi}_{i,k}$ and $\boldsymbol{\pi}$ is defined by:

$$D_{\text{CS}}\left(\boldsymbol{\pi}, \boldsymbol{\pi}_{i,k}\right) \triangleq -\log \frac{\langle \boldsymbol{\pi}, \boldsymbol{\pi}_{i,k} \rangle}{||\boldsymbol{\pi}|| \, ||\boldsymbol{\pi}_{i,k}||}, \quad (20)$$

where

$$\langle \boldsymbol{\pi}_1, \boldsymbol{\pi}_2 \rangle \triangleq \int \boldsymbol{\pi}_1(\mathbf{X}) \boldsymbol{\pi}_2(\mathbf{X}) \delta \mathbf{X}$$

and

$$||\boldsymbol{\pi}|| \triangleq \sqrt{\langle \boldsymbol{\pi}, \boldsymbol{\pi} \rangle}.$$

Hoang *et al.* [66] have proven that the CSD between two Poisson densities with intensity functions $\nu_1(\cdot)$ and $\nu_2(\cdot)$ are proportional to the squared geometric distance between the two functions, $\int_{\mathbb{X}} |\nu_1(x) - \nu_2(x)|^2 \, dx$. Hence, in a scenario where PHD filters are locally running on each vehicle, the general CSD-based fusion rule,

$$\bar{\boldsymbol{\pi}}_{0,k} = \underset{\boldsymbol{\pi}}{\text{argmin}} \sum_{i=0}^{N} \omega_i \, D_{\text{CS}}(\boldsymbol{\pi} || \boldsymbol{\pi}_{i,k}) \quad (21)$$

leads to

$$\bar{\nu}_{0,k}(\cdot) = \underset{\nu(\cdot)}{\text{argmin}} \sum_{i=0}^{N} \omega_i \int_{\mathbb{X}} |\nu(x) - \nu_{i,k}(x)|^2 \, dx \quad (22)$$

and the solution turns out to be given by simply arithmetic averaging of the individual intensity functions:

$$\bar{\nu}_{0,k}(\cdot; \boldsymbol{\omega}) = \sum_{i=1}^{N} \omega_i \, \nu_{i,k}(\cdot). \quad (23)$$

Similarly, with LMB filters being in place on each vehicle, the fusion rule turns out as follows [11]:

$$\begin{aligned}\bar{r}_k^{(\ell)}(\boldsymbol{\omega}) &= \sum_{i=0}^{N} \omega_i\, r_{i,k}^{(\ell)} \\ \bar{p}_k^{(\ell)}(\cdot;\boldsymbol{\omega}) &= \sum_{i=0}^{N} \omega_i\, r_{i,k}^{(\ell)}\, p_{i,k}^{(\ell)}(\cdot) \Big/ \sum_{i=0}^{N} \omega_i\, r_{i,k}^{(\ell)} \end{aligned} \tag{24}$$

Interestingly, the above fusion rules, both (23) and (24) have a *complementary* nature, in the sense that if an object is detected by a few vehicles and not by others (due to limited FoVs), the arithmetic averaged intensity, $\bar{v}_{0,k}(\cdot;\boldsymbol{\omega})$, returned by fusion via (23) can still be large around the object's state, and the fused probability of existence, $\bar{r}_k^{(\ell)}(\boldsymbol{\omega})$ returned by (24), can still be close to 1 even if one of the probabilities of existence are close to zero.

C. EXPLICTLY COMPLEMENTARY FUSION WITH LMB FILTER

The particular form of an LMB density allows an explicitly complementary fusion rule to be developed for combining LMB densities in such a way that *all* the detected objects by any vehicles connected to the ego vehicle are included in the fused LMB posterior. Consider the ego vehicle V_0 with the locally computed LMB posterior

$$\boldsymbol{\pi}_{0,k}(\cdot) \sim \left\{\left(r_{0,k}^{(\ell)}, p_{0,k}^{(\ell)}(\cdot)\right)\right\}_{\ell\in\mathbb{L}_k}.$$

We note that $\boldsymbol{\pi}_{0,k}(\cdot)$ is the density of an LMB RFS that itself is the union of multiple single-Bernoulli RFS:

$$\boldsymbol{X}_{0,k} = \cup_{\ell\in\mathbb{L}_k} X_{0,k}^{(\ell)} \tag{25}$$

where $X_{0,k}^{(\ell)}$ is a single-Bernoulli RFS parameterized by $\left(r_{k,0}^{(\ell)}, p_{k,0}^{(\ell)}(\cdot)\right)$. A similar definition can be made for LMBs corresponding to the posterior densities calculated locally in all the connected vehicles V_i.

In order to ensure that *all* object detections are included in the fused LMB, the explicit approach is to unify all the LMBs corresponding to the locally calculated posteriors:

$$\bar{\boldsymbol{X}}_{0,k} = \cup_{i=0}^{N} \boldsymbol{X}_{i,k} \tag{26}$$

and calculate the multi-object density of the result. Note that the outcome of (26) is not a proper labeled RFS, because two distinct members of the set can have the same label. Indeed, equation (26) can be further expanded to

$$\bar{\boldsymbol{X}}_{0,k} = \cup_{i=0}^{N}\left[\cup_{\ell\in\mathbb{L}_k} X_{i,k}^{(\ell)}\right] \tag{27}$$

$$= \cup_{\ell\in\mathbb{L}_k}\left[\cup_{i=0}^{N} X_{i,k}^{(\ell)}\right] \tag{28}$$

$$= \cup_{\ell\in\mathbb{L}_k} Y_{0,k}^{(\ell)} \tag{29}$$

where $Y_{0,k}^{(\ell)} \triangleq \cup_{i=0}^{N} X_{i,k}^{(\ell)}$ is the union of all the single-Bernoulli RFSs associated with label ℓ in each vehicle node. Gostar *et al.* [67] have approximated this set with a single-Bernoulli RFS that has the smallest KLD from the set. Substituting that approximation in equation (29) leads to the following fused LMB posterior parameters: [67, equations 9 and 10]

$$\begin{aligned}\bar{r}_k^{(\ell)} &= \sum_{i=0}^{N} \varrho_{k,i}^{(\ell)} \Big/ \left[1+\sum_{i=0}^{N} \varrho_{k,i}^{(\ell)}\right] \\ \bar{p}_k^{(\ell)}(\cdot) &= \sum_{i=0}^{N} \varrho_{k,i}^{(\ell)}\, p_{k,i}^{(\ell)}(\cdot) \Big/ \sum_{i=0}^{N} \varrho_{k,i}^{(\ell)} \end{aligned} \tag{30}$$

where

$$\varrho_{k,i}^{(\ell)} = r_{k,i}^{(\ell)} \big/ \left[1 - r_{k,i}^{(\ell)}\right].$$

The strong emphasis of the above fusion rule on *inclusion* of every detection in the result is evidenced by the observation that if any of the vehicles, e.g. V_i, deduces that object ℓ exists with a probability $r_{k,i}^{(\ell)}$ that is very close to 1, then for that i and that object label ℓ, $\varrho_{k,i}$ will be extremely large. Hence, fused the probability of existence, $\bar{r}_k^{(\ell)}$ will be very close to 1 as well, regardless of how small or large the other probabilities of existence are for that object label. In addition, it also has the advantage of requiring less convergence time for information consistency between sensors compared to CSD due to the inclusion of *all* information despite both methods being complementary in nature.

D. EVALUATION METRICS

In order to fully evaluate the performance of RFS methods, an easily computed, robust evaluation metric must be used. The most commonly used method is known as optimal sub-pattern assignment (OSPA) first devised by Schuhmacher *et al.* in [68]. At its core it is a method of evaluating the performance of RFS multi object trackers by computing the distance between two RFS states in a way which is easily interpretable. The OSPA error incorporates the cardinality error, as well as the state error of an RFS state.

Beard *et al.* [69], [70] demonstrated that the OSPA metric could be used to develop a rigorous evaluation of multi-target tracking performance. They introduced OSPA(2), similar to the original OSPA metric; however, the base distance is an OSPA-based distance. Note that the original OSPA distance computes the error between the true and estimated multi-target states at the time step. In contrast, the OSPA(2) distance captures the error between the true and estimated sets of tracks over a window of time steps.

V. NETWORK IMPLEMENTATIONS

A. CENTRALIZED NETWORK IMPLEMENTATIONS

Gostar *et al.* [11] utilized the CSD in the LMB filter framework to fuse the information in a centralized network scenario. The authors observed that the KLD-based fusion method performs poorly when the fields of view of the sensors differ and therefore, proposed and derived a CSD-based method, formulated upon the previous work by Hoang *et al.* [66]. The CSD was formulated in both the PHD and the LMB (by approximating the LMB density to its first moment Poisson density) filters. The proposed approach was implemented with both Gaussian Mixture (GM) and sequential Monte Carlo (SMC) approaches and was compared to the KLD-based fusion results. It was found that the CSD-based fusion method significantly outperformed the KLD-based method in both PHD and LMB implementations. However, when used with the CSD-based fusion rule, the LMB filter's performance was superior to that of the PHD filter, especially when object birth and death occur because the CSD-based fusion rule can rapidly detect changes in cardinality and correct their estimates.

B. DECENTRALIZED NETWORK IMPLEMENTATIONS

Fröhle *et al.* [71] approximated the Poisson Multi Bernoulli Mixture (PMBM) [72] filter with a Poisson Multi Bernoulli (PMB) distribution to implement multi-sensor fusion in a low cost efficient decentralized sensor network. The fusion between sensors was achieved via the introduction of a fusion map based on KLD between object tracks, which reduced complexity. The EKF approximates the state. The other major consideration in the paper was to treat objects with an extended tracking model as per [72], [73]. Measurements were obtained from stationary lidar sensors. The number of sensors was two, and a PMBM filter was running locally for each sensor. The performance was compared to the filters running independently and against the centralized filter in terms of generalized optimal sub-pattern assignment (GOSPA) metric. It was seen that GOSPA distance was comparable to the centralized filter with access to all measurements.

C. DISTRIBUTED NETWORK IMPLEMENTATIONS

Due to its lack of scalability, information fusion in a centralized network is infeasible in many applications where the number of sensor nodes is large. Battistelli *et al.* [62] proposed a fusion rule for distributed multi-object tracking (DMTT) using CPHD filters in each sensor node implemented by GM approximation. The proposed filter was called the Consensus Gaussian Mixture-CPHD (CGM-CPHD) filter. Information consensus [74] was achieved using the Kullback-Leibler average (KLA). The results were compared with a centralized filter called Global GM-CPHD (GGM-CPHD), which performs global fusion among all network nodes. Using different numbers of consensus steps $L = 1, 2, 3$ for the distributed filter, the performance was compared, with 3 consensus steps having performance comparable to the global filter. Using more consensus steps, performance could be matched with the global filter; however, at the cost of extra computational cost.

Li and Hlawatsch [12] focused on the inherent advantages of a distributed network, allowing for the parallelization of both filtering and fusion steps. They implemented an arithmetic average fusion scheme in a distributed network based on importance sampling, which has been shown in [64], [75] to be equivalent to Gaussian Averaging fusion of PHDs. The key contributions in this work are the GM-particle and particle-GM conversion steps, which allow for both a reduced communication overhead as only Gaussian Components are communicated and enable both linear and non-linear sensors to be combined by performing the filtering locally using particles. In addition, the solution enjoys reduced communication requirements between vehicles and enabled parallel filtering and fusion steps by utilizing importance sampling, which does not require sampling from the fused PHD. Simulation results show that their method is slightly outperformed by the standard sampling method. However, the importance sampling enables parallelization. In another scenario where half the sensors are linear and use a GM implementation, while the non-linear sensors use particles, they showed that with consensus, the method outperforms the flooding scheme. Flooding performed better here as GM-PHD filters perform better with a flooding scheme.

Gao *et al.* [76] devised a distributed method of fusing multiple agent maps with limited sensor FoV without the relative pose between agents being known. The Area of Interest (AoI) must have stationary landmarks to reference the registration algorithm. The Cauchy-Schwarz Fusion metric was used to minimize the map discrepancy. The method was implemented using the PHD filter and a distributed network, where the densities are represented as a GM. The drift and orientation (DaO) parameters are minimized by an Instant Cost (IC) function in a neighbor-wise strategy according to the weighted CSD paradigm. The algorithm was evaluated through the optimal sub-pattern assignment (OSPA) metric [68] against distributed fusion with perfect DaO parameters and local mapping with no cooperation. After some time, when fusion occurs, the proposed method is able to almost match the optimal method.

Gao *et al.* [77] proposed to use the linear opinion pool (linOP) to minimize information loss when fusing multiple objects. Normal linOP cannot be directly applied to multi-object fusion, as the produced averages are not of the same type as the multi-object densities. By exploiting the minimum information loss (MIL) paradigm, linOP could be applied to multi-object densities. It was implemented using KLD minimization. The implementation was tested with a distributed Peer to Peer (P2P) network running on a GM-CPHD filter with consensus steps. It was compared to optimal KLD-based fusion and MIL fusion in a centralized network. With a probability of detection $P_D = 0.98$, the proposed method was able to match the optimal case. Simultaneously, for $P_D = 0.5$ it was shown to outperform the optimal methods; however, it was observed that the MIL method performed worse for high clutter rates than optimal KLD-based fusion in terms of OSPA metric.

Gao *et al.* [78] have built upon the work in [76] where only a single object was considered. In this work, KLD-based fusion was used along with the consensus method. The resulting filter called the Joint Sensor registration and distributed multi-object tracking (JRT-DMT) algorithm uses the PHD filter implemented using GM approximation, where consensus is achieved via consecutive KLD-based fusion operations.

Experimental results showed that compared to a CPHD filter with perfect knowledge of the registration parameters, the proposed method's performance was very similar once registration was achieved, and the consensus algorithm ran.

Li *et al.* [79] have extended the cardinality consensus AA (CC-AA) fusion method [80], [81] for Multi Bernoulli (MB) filters. This was based on previous observations of AA fusion [12]. The authors first proved that both AA and GA are fréchet means. The main contribution in their work was to propose a *target-wise fusion rule* which was able to split the MB fusion into parallel *Bernoulli-to-Bernoulli* problems in order to firstly extend AA fusion to MB filters, and secondly to improve performance with so-called *large scale sensor networks*. Both flooding and consensus methods were investigated. It was found that the CC-AA method had minimal computational cost and provided decent tracking performance in terms of OSPA error. However, compared to flooding, it performed much worse. The flooding approach rapidly became computationally infeasible compared to the consensus-based methods.

TABLE 3. Summary of state-of-the-art RFS-based sensor fusion methods.

Name	Filter	Fusion Method	Architecture
Gostar et al. [11]	LMB	CSD-based	Centralized
Fröhle et al. [71]	PMB	KLD-based	Decentralized
Battistelli et al. [62]	CPHD	KLD-based	Distributed
Li et al. [12]	PHD	Arithmetic Avg.	Distributed
Gao et al. [76]	PHD	CSD-based	Distributed
Gao et al. [77]	CPHD	Linear Operation	Distributed
Gao et al. [78]	CPHD	KLD-based	Distributed
Li et al. [79]	MB	Arithmetic Avg.	Distributed

The above methodse are all based on using RFS filters running at the core of each sensor or vehicle node. A summary of various fusion methods in different sensor networks are presented in Table 3.

VI. SIMULATION RESULTS

A. SIMULATION SETUP

To test and validate the performance of RFS-based methods for information fusion in ITS, the following scenarios are designed using the Driving Scenario Designer in MATLAB. Each vehicle is traveling at a constant velocity of 16 m/s. Since extended object tracking is not in the scope of this work, for the purpose of comparative investigation of fusion methods performances, vehicles are modeled as 0.1 cm×0.1 cm point objects to ensure that each radar only detects a single measurement per vehicle. Each object state is comprised of 2D position, 2D velocity and turning rate, denoted by $x = [p_x \; p_y \; \dot{p}_x \; \dot{p}_y \; \omega]^\top$. To maintain the simulation simple yet able to provide general indication of performance, we did not include the road information in the filter and considered a uniform survival probability of $p_S = 0.99$.

Each vehicle motion was modeled according to a coordinated turn (CT) model with a sampling period of $\Delta = 0.1$ s. The transition density for a CT model is

$$f_{k|k-1}(\cdot|x) = \mathcal{N}(\cdot; m(x), Q)$$

where

$$m(x) = \left[[p_x \; \dot{p}_x \; p_y \; \dot{p}_y] \times F(\omega)^\top \;\; \omega \right]^\top,$$

and

$$Q = \operatorname{diag}\left(\sigma_\omega^2 G G^\top, \sigma_u^2\right)$$

in which $\sigma_\omega = 10\,\text{m/s}^2$ and $\sigma_u = 30°/\text{s}$, and

$$F(\omega) = \begin{bmatrix} 1 & \frac{\sin\omega}{\omega} & 0 & -\frac{1-\cos\omega}{\omega} \\ 0 & \cos\omega & 0 & -\sin\omega \\ 0 & \frac{1-\cos\omega}{\omega} & 1 & \frac{\sin\omega}{\omega} \\ 0 & \sin\omega & 0 & \cos\omega \end{bmatrix}, G = \begin{bmatrix} \frac{1}{2} & 0 \\ 1 & 0 \\ 0 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \tag{31}$$

Object birth is modeled as a LMB with $\mathbb{L}$ individual components,

$$\boldsymbol{\pi}_{B,k} = \{(r_B^{(k,i)}, p_B^{(k,i)}(\cdot))\}_{i=1}^{\mathbb{L}}$$

where each component has a probability of existence of

$$r_B^{(k,i)} = r, \quad i = 1, \ldots, |\mathbb{L}|$$

and their densities are Gaussian

$$p_B^{(k,i)}(\cdot) = \mathcal{N}(\cdot; m_B^{(i)}, Q_B), \quad i = 1, \ldots, |\mathbb{B}|$$

Measurements are returned in a Cartesian format relative to the ego vehicle. They are converted to global coordinates via the homogeneous transformation matrix

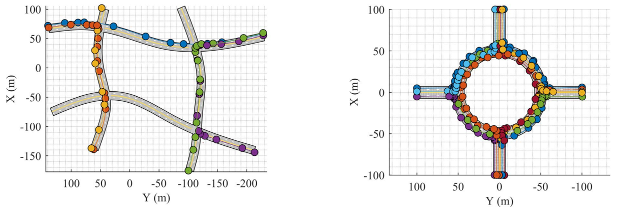
$$\begin{bmatrix} \cos\theta_{r,k} & -\sin\theta_r & x_{r,k} \\ \sin\theta_{r,k} & \cos\theta_{r,k} & y_{r,k} \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

where $\theta_{r,k}$ is the angle of the sensor $r \in \{1, \ldots, s\}$, relative to the global origin at time k and $x_{r,k}, y_{r,k}$ are the x, y Cartesian coordinates of the sensor position relative to the global origin at time k respectively. The standard deviations of the Gaussian distributed measurement noise are $\sigma_x = \sigma_y = 2$.

Three fusion schemes were examined for their performance in the above scenario: the KLD-based, CSD-based, and complementary fusion schemes all with LMB filters onboard each vehicle (sensor node). To implement the LMB filter, the density of each Bernoulli component was approximated by 3000 particles. In addition, to maintain computational tractability, Bernoulli components with probability of existence lower than 10^{-3} were pruned. OSPA and OSPA2 errors were computed with order $p = 1$, cutoff $c = 100$ for OSPA, and $c = 50$ for OSPA2. A window length of 10 was used for OSPA2.

1) SCENARIO 1

In this scenario, a typical road layout in an urban setting of 200 m × 200 m size is simulated. It features four intersections, with the curvy roads for a range of vehicle motions. Each road is two-way, and each side has two lanes. The lanes are 3.5 m wide to follow the Australian lane width standard [82]. Fig. 6 shows a snapshot of the simulated environment and vehicle trajectories.



(a) Scenario 1

(b) Scenario 2

FIGURE 6. a) Scenario 1 as shown in the Matlab app window. It shows a common urban road scenario with roads about 100m apart. The colored lines represent vehicle trajectories. b) Scenario 2, based on Fig. 1 representing a much more realistic scenario due to the number of targets and their relative proximity to one another with a highly challenging trajectory.

The scenario contains five vehicles, each equipped with radar with an axis coincident with the vehicle's center. There are four further stationary sensors located at each intersection. The FoV is a disk with a radius of 150 m. This disk only covers a portion of the total area at each time.

in which the five different means that are close to five possible points of entry (with small speeds and turning rates) and the components have all the same covariance

$$Q_B = \mathrm{diag}\big(1.5, 2.5, 1.5, 2.5, \frac{5\pi}{180}\big)^2.$$

In the simulation, entry of new vehicles to the scene (object birth) occur at times times 22, 1, 42, 1, and 42, and exits (object death) at times 272, 220, 218, 266, 288.

Each sensor has a homogeneous detection probability of $p^D = 0.99$ when object size and distance are within given threshold values. This detection depends on the reference range, R_{ref}, in meters, and the reference radar cross-section, R_{RCS}, in DBsm. R_{ref} is the detection range (maximum distance of an object that can be detected with probability of $p_D = 0.99$. In the simulation, $R_{\mathrm{ref}} = 150\,\mathrm{m}$ and $R_{\mathrm{RCS}} = -20\,\mathrm{DBsm}$. The detection area is divided into regions called resolution cells. The number of cells depends on the resolution settings for azimuth and range,

$$n_{\mathrm{cells}} = \frac{R_{\max}}{\delta_r} \times \frac{\theta}{\phi} \tag{33}$$

where $R_{\max}$ is the maximum detection range, δ_r is the range resolution, θ is the azimuth, and ϕ is the azimuth resolution. In the simulations, $R_{\max} = 150\,\mathrm{m}$, $\delta_r = 2.5\,\mathrm{m}$, $\theta = 360°$ and $\phi = 4°$, which gives 5400 resolution cells. Two scenarios were run: one with no noise or false alarms and one with randomly generated uniform noise and a false alarm rate, λ, of $10e^{-3}$. This leads to $n_{\mathrm{cells}} \times \lambda = 5.4$ false alarms per sensor for each time k.

2) SCENARIO 2

The second scenario shares many similarities to the first, with the key differences being the amount of vehicles and the way clutter is generated. The scenario replicates a roundabout which poses a greater challenge to multi object trackers. This is due to the number of targets, the increased proximity of targets and the much more challenging trajectories of each vehicle in the scene. The scenario encloses a 200 m × 200 m. Lanes are 3.5 m wide, with entry into the roundabout having 2 lanes on either side, while the roundabout itself has 4. The maximum number of vehicles in the scene at any time is 10, with one stationary sensor in the center. The vehicles move at a constant velocity of 16 m/s, except when they approach a corner where velocities are reduced to 4m/s. Compared to scenario 1 clutter generation differs in three distinct ways:

- Clutter generation is now produced by a Poisson distribution. In this scenario the false alarm rate is set to $\lambda = 1$.
- Clutter is limited to the FoV of each sensor.
- Clutter is constrained by the road geometry, such that it can only appear on the road itself.

Measurements have noise applied to them with a standard deviation $\sigma_x = \sigma_y = 2\,m$. The covariance for the entry points is the same for all targets.

$$Q_B = \mathrm{diag}\big(0.25, 0.125, 0.25, 0.125, \frac{\pi}{180}\big)^2.$$

Target birth occurs at 1, 32, 2, 22, 52, 12, 12, 2, 72, 52 and target death occurs at 232, 274, 224, 155, 289, 153, 211, 193, 311, 205. All sensors have a $\theta = 360°$ FoV, where $R_{\max} = 50\,\mathrm{m}$ for sensors 1-10 and $R_{\max} = 120\,\mathrm{m}$ for sensor 11.

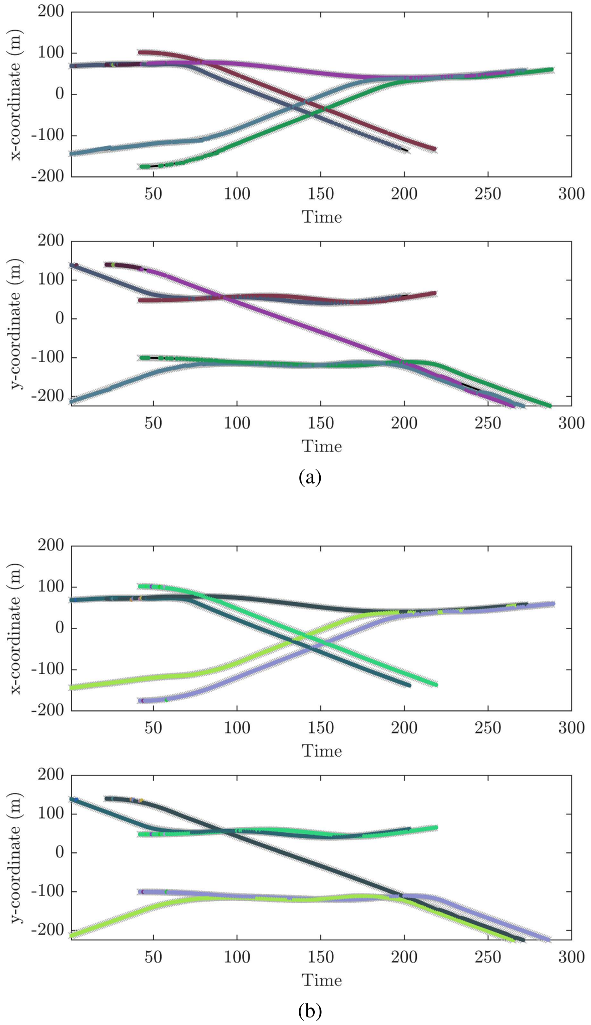
B. RESULTS AND DISCUSSIONS

Scenario 1a: Clean measurements

In our investigation, we generated the simulated ground-truth in an ideal case of no measurement noise or false alarms. The purpose of this study is to isolate tracking errors that are integral to the multi-sensor fusion method and induced by limited sensor FoV rather than by noise and clutter.

Our first observation is that as we anticipated, the KLD-based fusion method entirely fails to detect and track the multiple objects, due to its intrinsic need for agreement between sensor detections and the observation that no vehicle at any time falls within the FoV of all the others.

The second observation is the occurrence of *label switching* which mainly happens when two vehicles are too close to each other (less than lane width which is 3.5 m). Fig. 7(a) and Fig. 7(b) show the estimated tracks in presence of no noise or clutter, returned by the CSD-based and complementary fusion methods, respectively. In both cases, occasional switching of track labels (track colors in the graphs) are evident. Note that both figures show the results



(a)

(b)

FIGURE 7. A single-run sample of the vehicle tracks returned by (a) the CSD-based fusion and (b) the complementary fusion methods, in presence of no noise or clutter.

of the 100 Monte Carlo runs of the simulation that we ran to average for tracking error calculation.

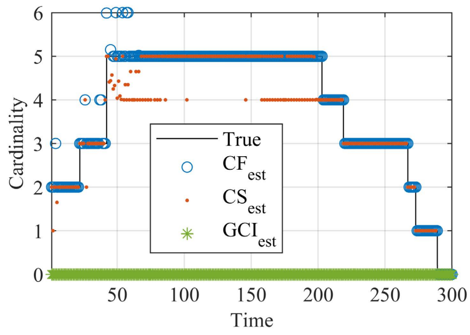


FIGURE 8. The cardinality estimated for all three fusion types averaged over 100 iterations, in a noiseless and clutter free scenario.

The third observation is related to the number of detected and tracked vehicles (cardinality estimates) as presented in Fig. 8. The KLD-based filter returns zero cardinality at all times, due to lack of agreement between all sensors on any vehicle at any time. The CSD-based fusion method underestimates the number of vehicles in a large number of sampling times. That is because some vehicles that are detected by only one sensor may not receive a large enough probability of existence in the fused LMB posterior, and consequently not appear in the estimated tracks. On the other hand, the complementary fusion method returns the correct cardinality estimate most of the times with occasional overestimates.

The extra detected vehicle returned by the complementary fusion method is indeed a close copy of another vehicle that (i) either has recently entered the scene and detected by all sensors in turn - for instance this happens at around $k = 50$, (ii) or has recently exited the scene (object death) and the absolute complementary nature of the fusion method leads to the object death to appear in results with delay - for instance in times of vehicle exits (true cardinality decrements) after $k = 200$. The lag here is one to two steps only.

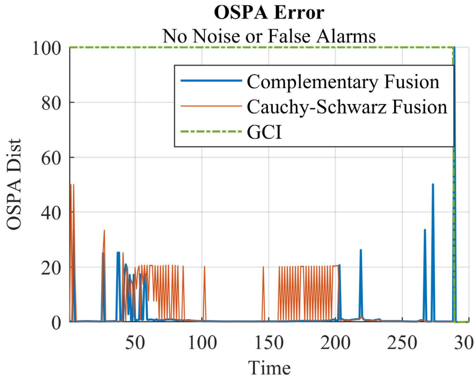


FIGURE 9. Combined OSPA distance graph for all three fusion types averaged over 100 iterations, where noise and clutter is not considered.

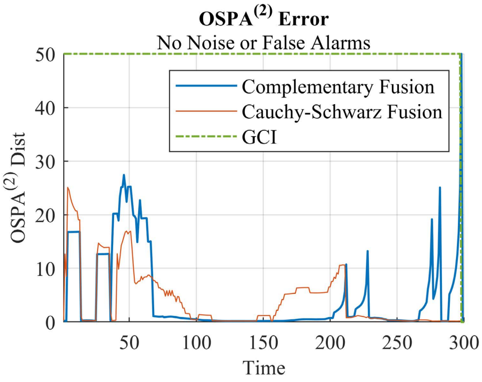


FIGURE 10. Combined OSPA2 distance graph for all three fusion types averaged over 100 iterations, where noise and clutter is not considered.

Overall estimation error (in cardinality and states together) have been calculated in terms of OSPA [68] distance and shown in Fig. 9. We can see that KLD-based fusion returns a maximum error all the times, and the complementary fusion returns an error that is significantly smaller than CSD-based fusion most of the times, except for a few spikes that occur due to the above mentioned lag in detecting object death after fusion.

Scenario 1b: Realistic measurements

To evaluate the estimation and tracking performance of fusion methods in a more realistic scenario, we added uniformly distributed Gaussian noise to sensor measurements and false alarms with a rate of 10^{-3} per resolution cell. Fig. 15(a) and Fig. 15(b) present the returned track estimates by the CSD-based and complementary fusion methods, respectively. In each figure, the measurements (including the false alarms returned by all sensors onboard the vehicles) are also shown. Again note that this is a sample of the 100 Monte Carlo runs of the simulation (and the errors are averaged across all the 100).

The cardinality estimates returned by each of the three fusion methods are presented in Fig. 16. It shows that in presence of the relatively large number of false alarms, the problem of false tracks (overestimated cardinality) signifies with the complementary fusion method. This can be observed as the false tracks (isolated color dots) in Fig. 15(b). However, being isolated means that those false track points do not persist and are quickly corrected by the filter. On the other hand, the CSD-based fusion consistently underestimates cardinality (misses some tracks) for finite intervals of time.

It is noteworthy that those few false tracks returned by the complementary fusion method may be due to the double-counting of some tracks. However, the vehicle trajectories are so close together that a larger threshold for double-track removal would remove the true tracks as well.

Fig. 17 presents the OSPA errors for each method, and we observe that in general, the complementary fusion method performs better than the others (with lower OSPA errors with a few spikes, and the mean OSPA error being still close to zero.). Note that the slight degradation of the complementary fusion performance is due to the extra data association from surrounding measurements.

We note that in connected driving applications, from driving safety point of view, it is more desirable to overestimate the number of vehicles in the surroundings, rather than have some unaccounted for.

Scenario 2

Scenario 1 was able to show the effectiveness of using RFS tracking methods in ITS, however the tested scenario used conditions which were far from realistic when considering ITS. In our second scenario we develop a significantly more complex scene featuring a large central roundabout which vehicles enter at varying points in the scene. The number of vehicles (double that of scenario 1) provides a more likely number of targets where the limiting factor when choosing the amount was computational time. Indeed, simply doubling the amount of targets increased computation by approximately 3 times. This highlights the limitations of using a centralized approach, as the computational cost does not scale linearly. Looking at Fig. 11 it is evident that in this more complex scene CSD fusion performs significantly worse than previously. Looking at Fig. 13, there is the issue of consistent underestimation from times $k = 0$ to $k = 230$ which may be caused by a missed track, as well as the large amount of noise in the in the cardinality for CS fusion. It should be noted that the filter parameters were the same for both CS and complementary fusion, and that increasing them for CS fusion may have improved the results, but would negate the previously established advantages of CS fusion such as reduced computation time with minimal loss of accuracy. Looking at complementary fusion we see that it performs well even with double the amount of vehicles with a relatively large amount of Gaussian noise applied to the measurements. As expected GCI fusion failed completely due to its reliance on all sensors detecting the same targets.

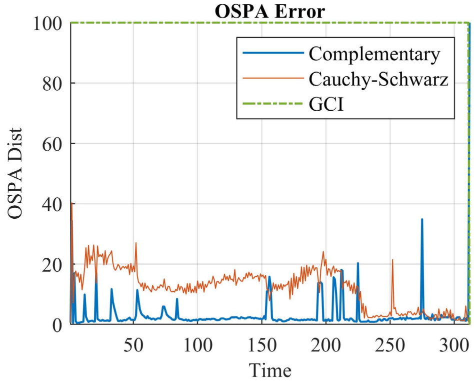


FIGURE 11. Average OSPA distance errors for scenario 2 for each of the three methods tested.

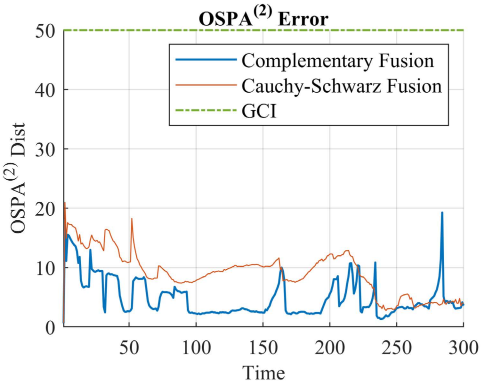


FIGURE 12. Average OSPA2 for scenario 2 for each of the three methods tested.

Computational cost

Table 4 illustrates the difference in run times between two different processors. The first four values look at the average filtering time for one step, that is $k \rightarrow k+1$, across 5 runs. The lower four represent the average update time required per sensor for each filtering step. This was included to illustrate how high-powered mobile hardware compares to desktops, to highlight the possibility of real-time filters running in an ITS system.

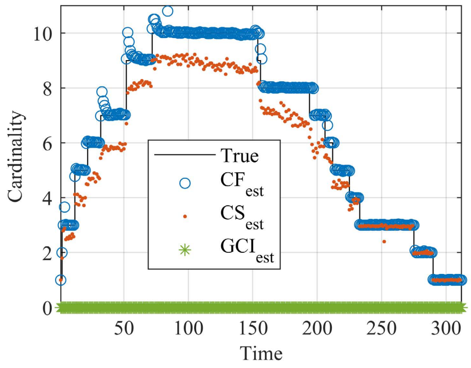


FIGURE 13. Average cardinality for scenario 2 for each of the three methods tested.

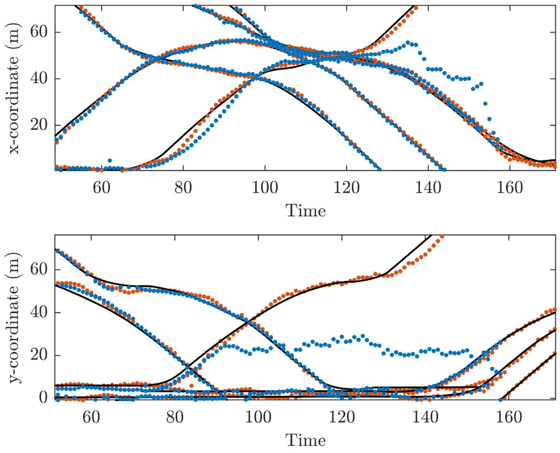


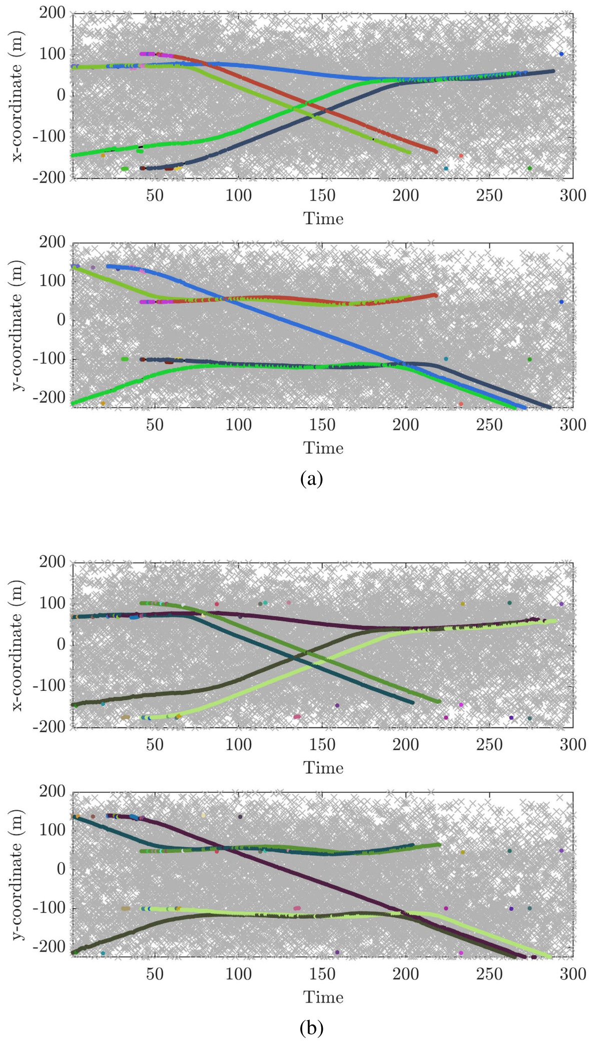
FIGURE 14. A zoomed in comparison of some of the tracks between $k = 60$ and $k = 140$ of the track estimations for CS and complementary fusion. Red points are complementary fusion and blue points are CS fusion.

TABLE 4. Comparison of average simulation times over five runs (s).

Scenario - filter	AMD 5900x	Intel i7
Scenario 1a - Comp. fusion	1.28	2.97
Scenario 1b - Comp. fusion	0.84	1.67
Scenario 1a - CSD-based fusion	0.25	0.52
Scenario 1b - CSD-based fusion	0.26	0.46
Scenario 1a - Comp. fusion - update only	0.088	0.18
Scenario 1b - Comp. fusion - update only	0.059	0.10
Scenario 1a - CSD-based fusion - update only	0.023	0.47
Scenario 1b - CSD-based fusion - update only	0.023	0.40

The two processors compared are: (1) the AMD Ryzen 9 5900x 12 core, 24 thread processor with a base speed of 3.7 GHz and a max single-core boost clock of 4.8 GHz, and (2) the Intel i7-7820HQ 4 core, 8 thread processor with a base clock of 2.9 GHz and a single-core boost clock of 3.9 GHz.

It can be seen that a modern high-end desktop processor has approximately twice the performance of an older laptop processor. Note that modern chips such as Apple's new M1 chip, based on the ARM architecture, show very high single-core performance on par with desktop chips with significantly lower power consumption. Custom-designed chips for ITS could be a solution as they would have chipsets optimized for specific instructions running optimized C++ code instead of a MATLAB implementation.



(a)

(b)

FIGURE 15. A single-run sample of the vehicle tracks returned by (a) the CSD-based fusion and (b) the complementary fusion methods, in presence of measurement noise and clutter with a false alarm rate of $\lambda = 10^{-3}$ per resolution cell.

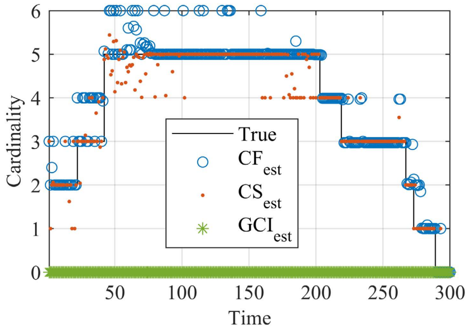


FIGURE 16. Cardinality estimates for the scenario averaged over 100 iterations, where random uniformly distributed noise and a false alarm rate of $\lambda = 10^{-3}$ is considered.

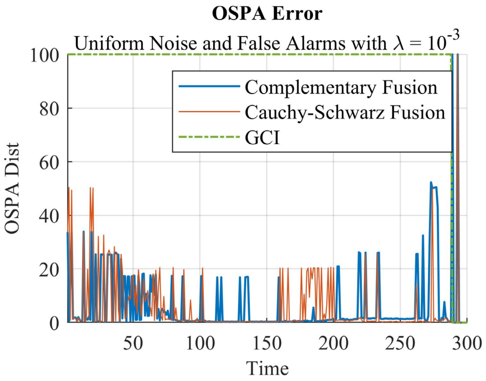


FIGURE 17. Combined OSPA distance graph for all three fusion types averaged over 100 iterations, where noise and clutter is considered. Noise is random and uniformly distributed and $\lambda = 10^{-3}$.

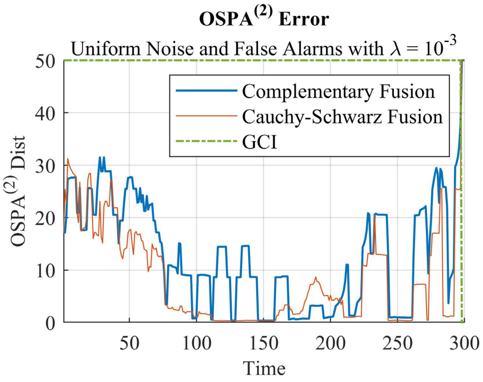


FIGURE 18. Combined OSPA2 distance graph for all three fusion types averaged over 100 iterations, where noise and clutter is considered. Noise is random and uniformly distributed and $\lambda = 10^{-3}$.

In terms of feasibility for real-time applications, neither complementary nor CSD-based fusion are feasible without taking communication latency into account; even the AMD processor takes around 12 times longer than the sensor sampling time for complementary and 2 times longer for CSD-based fusion. For the filter to work in real-time, the iteration time must be less than the sampling time.

VII. CONCLUSION

This paper provided a brief literature review in the current fusion applications in intelligent transportation systems for vehicle tracking, specifically considering cooperative or *complementary* fusion where vehicles or sensors from different locations and perspectives are used to increase global map awareness. It was found that few methods look at this specific case, with more consideration taken toward multimodal fusion. Additionally, a brief review of state-of-the-art fusion implementations for multi-object tracking based on RFS was conducted. Using two simulations developed in MATLAB, RFS tracking methods were determined to apply to ITS due to their strong performance in highly cluttered scenarios where vehicles can appear and disappear mid-scene. MATLAB simulations were run to numerically show how various fusion methods perform based on the Labeled Multi Bernoulli filter running in a centralized implementation. Results showed a compromise between performance and accuracy between CSD-based fusion and complementary fusion. It was found that the use of complementary fusion enabled vehicles to gain a more complete awareness of their surrounding, which originally were outside of their on-board sensors FoV. At the same time, the commonly used KLD-based method does not apply to ITS scenarios due to limited sensor fields of view. This is because of the method's strong requirement for agreement between all sensors prior to fusion, which is an impossibility with limited FoVs. Future works will focus on optimizing complementary fusion due to its tendency to overestimate rather than underestimate. This will be done by developing a new distributed information fusion algorithm utilizing complementary fusion, specifically on minimizing double counting.

REFERENCES

[1] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.

[2] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 74–84, Nov. 2016.

[3] E. Coelingh and S. Solyom, "All aboard the robotic road train," *IEEE Spectr.*, vol. 49, no. 11, pp. 34–39, Nov. 2012.

[4] M. E. Liggins, C.-Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proc. IEEE*, vol. 85, no. 1, pp. 95–107, Jan. 1997.

[5] D.-J. Lee, *Unscented Information Filtering for Distributed Estimation and Multiple Sensor Fusion*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2008, ch. 1, pp. 7426–7441.

[6] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 46–58, May 2011.

[7] A. Farhadi, P. Dower, and M. Cantoni, "Computation time analysis of centralized and distributed optimization algorithms applied to automated irrigation networks," in *Proc. Austral. Control Conf.*, Nov. 2013, pp. 263–269.

[8] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, "Cooperative multi-vehicle localization," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2006, pp. 564–570.

[9] M. Brambilla, M. Nicoli, G. Soatti, and F. Deflorio, "Augmenting vehicle localization by cooperative sensing of the driving environment: Insight on data association in urban traffic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1646–1663, Apr. 2020.

[10] M. Frohle, K. Granstrom, and H. Wymeersch, "Multiple target tracking with uncertain sensor state applied to autonomous vehicle data," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jun. 2018, pp. 628–632.

[11] A. K. Gostar, T. Rathnayake, R. Tennakoon, A. Bab-Hadiashar, G. Battistelli, L. Chisci, and R. Hoseinnezhad, "Cooperative sensor fusion in centralized sensor networks using Cauchy–Schwarz divergence," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107278.

[12] T. Li and F. Hlawatsch, "A distributed particle-PHD filter using arithmetic-average fusion of Gaussian mixture parameters," *Inf. Fusion*, vol. 73, pp. 111–124, 2021.

[13] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[14] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[15] S. A. Kashinath, S. A. Mostafa, A. Mustapha, H. Mahdin, D. Lim, M. A. Mahmoud, M. A. Mohammed, B. A. S. Al-Rimy, M. F. M. Fudzee, and T. J. Yang, "Review of data fusion methods for real-time and multi-sensor traffic flow analysis," *IEEE Access*, vol. 9, pp. 51258–51276, 2021.

[16] A. Buonviri, M. York, K. LeGrand, and J. Meub, "Survey of challenges in labeled random finite set distributed multi-sensor multi-object tracking," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–12.

[17] J. Liu, B.-G. Cai, and J. Wang, "Cooperative localization of connected vehicles: Integrating GNSS with DSRC using a robust cubature Kalman filter," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2111–2125, Aug. 2017.

[18] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar, N. V. Irukulapati, C. Englund, P. Falcone, S. Papanastasiou, L. Svensson, and H. Wymeersch, ''Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 994–1007, Sep. 2012.

[19] H. Li and F. Nashashibi, ''Cooperative multi-vehicle localization using split covariance intersection filter,'' *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp. 33–44, Summer 2013.

[20] P. Merdrignac, O. Shagdar, and F. Nashashibi, ''Fusion of perception and V2P communication systems for the safety of vulnerable road users,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1740–1751, Jul. 2017.

[21] J. Radak, B. Ducourthial, V. Cherfaoui, and S. Bonnet, ''Detecting road events using distributed data fusion: Experimental evaluation for the icy roads case,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 184–194, Jan. 2016.

[22] V. Milanes, J. Perez, E. Onieva, and C. Gonzalez, ''Controller for urban intersections based on wireless communications and fuzzy logic,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 243–248, Mar. 2010.

[23] S. Wang and X. Jiang, ''Three-dimensional cooperative positioning in vehicular ad-hoc networks,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 937–950, Feb. 2021.

[24] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, ''Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2089–2100, Oct. 2014.

[25] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, ''Multivehicle cooperative driving using cooperative perception: Design and experimental validation,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 663–680, Apr. 2015.

[26] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, ''Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment,'' *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018.

[27] Q. Shi and M. Abdel-Aty, ''Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways,'' *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 380–394, Sep. 2015.

[28] S. G. Ritchie, ''A knowledge-based decision support architecture for advanced traffic management,'' *Transp. Res. A, Gen.*, vol. 24, no. 1, pp. 27–37, Jan. 1990.

[29] X. Wang, N. Cui, and J. Guo, ''Huber-based unscented filtering and its application to vision-based relative navigation,'' *IET Radar Sonar Navigat.*, vol. 4, no. 1, pp. 134–141, Feb. 2010.

[30] C. Karlgaard and H. Schaub, ''Comparison of several nonlinear filters for a benchmark tracking problem,'' in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Aug. 2006, p. 6243.

[31] L. Chang, B. Hu, G. Chang, and A. Li, ''Huber-based novel robust unscented Kalman filter,'' *IET Sci. Meas. Technol.*, vol. 6, no. 6, pp. 502–509, Nov. 2012.

[32] S. I. Roumeliotis and G. A. Bekey, ''Distributed multirobot localization,'' *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[33] R. Madhavan, K. Fregene, and L. E. Parker, ''Distributed heterogeneous outdoor multi-robot localization,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 2002, pp. 374–381.

[34] A. Martinelli, F. Pont, and R. Siegwart, ''Multi-robot localization using relative observations,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 2797–2802.

[35] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson, and J. Didoff, ''The grand cooperative driving challenge 2016: Boosting the introduction of cooperative automated vehicles,'' *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 146–152, Aug. 2016.

[36] S. J. Julier and J. K. Uhlmann, ''A non-divergent estimation algorithm in the presence of unknown correlations,'' in *Proc. Amer. Control Conf.*, vol. 4, Jun. 1997, pp. 2369–2373.

[37] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, ''Localization of a group of communicating vehicles by state exchange,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 519–524.

[38] H. Li and F. Nashashibi, ''Multi-vehicle cooperative localization using indirect vehicle-to-vehicle relative pose estimation,'' in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jul. 2012, pp. 267–272.

[39] G. Soatti, M. Nicoli, N. Garcia, B. Denis, R. Raulefs, and H. Wymeersch, ''Implicit cooperative positioning in vehicular networks,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3964–3980, Dec. 2018.

[40] M. Fröhle, C. Lindberg, K. Granström, and H. Wymeersch, ''Multisensor Poisson multi-Bernoulli filter for joint target–sensor state tracking,'' *IEEE Trans. Intell. Vehicles*, vol. 4, no. 4, pp. 609–621, Dec. 2019.

[41] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, ''A scalable algorithm for tracking an unknown number of targets using multiple sensors,'' *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3478–3493, Jul. 2017.

[42] M. Fröhle, C. Lindberg, K. Granstöm, and H. Wymeersch, ''Multisensor Poisson multi-Bernoulli filter for joint target-sensor state tracking,'' 2017, *arXiv:1712.08146*.

[43] P. Merdrignac, O. Shagdar, I. B. Jemaa, and F. Nashashibi, ''Study on perception and communication systems for safety of vulnerable road users,'' in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1876–1881.

[44] J. J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi, and J. E. Naranjo, ''Vehicle to pedestrian communications for protection of vulnerable road users,'' in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1037–1042.

[45] B. Ducourthial, V. Cherfaoui, and T. Denoeux, ''Self-stabilizing distributed data fusion,'' in *Stabilization, Safety, and Security of Distributed Systems*, A. W. Richa and C. Scheideler, Eds. Berlin, Germany: Springer, 2012, pp. 148–162.

[46] R. G. Rosa and D. P. L. Teresa, ''First applications of the Orbex coprocessor: Control of unmanned vehicles,'' *Mathware Soft Comput.*, vol. 7, pp. 265–273, Mar. 2008.

[47] J. E. Naranjo, C. González, R. García, and T. De Pedro, ''ACC+Stop&go maneuvers with throttle and brake fuzzy control,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 213–225, Jun. 2006.

[48] R. Abou-Jaoude, ''ACC radar sensor technology, test requirements, and test solutions,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 3, pp. 115–122, Sep. 2003.

[49] S. K. Gehrig and F. J. Stein, ''Collision avoidance for vehicle-following systems,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 233–244, Jun. 2007.

[50] A. Elfes, *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 1989.

[51] A. Birk and S. Carpin, ''Merging occupancy grid maps from multiple robots,'' *Proc. IEEE*, vol. 94, no. 7, pp. 1384–1397, Jul. 2006.

[52] H. Li and F. Nashashibi, ''Multi-vehicle cooperative perception and augmented reality for driver assistance: A possibility to 'see' through front vehicle,'' in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 242–247.

[53] S. Tsugawa, S. Kato, T. Matsui, H. Naganawa, and H. Fujii, ''An architecture for cooperative driving of automated vehicles,'' in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Oct. 2000, pp. 422–427.

[54] S.-W. Kim, Z. Jie Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang, ''Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 5059–5066.

[55] R. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion* (Electronic Warfare), Norwood, MA, USA: Artech House, 2014.

[56] D. Clark and B. N. Vo, ''Convergence analysis of the Gaussian mixture PHD filter,'' *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1204–1212, Apr. 2007.

[57] A. F. Garcia-Fernandez and B.-N. Vo, ''Derivation of the PHD and CPHD filters based on direct Kullback–Leibler divergence minimization,'' *IEEE Trans. Signal Process.*, vol. 63, no. 21, pp. 5812–5820, Nov. 2015.

[58] B. N. Vo and W. K. Ma, ''The Gaussian mixture probability hypothesis density filter,'' *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[59] R. P. S. Mahler, B.-T. Vo, and B.-N. Vo, ''CPHD filtering with unknown clutter rate and detection profile,'' *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3497–3513, Aug. 2011.

[60] B.-T. Vo, B.-N. Vo, and A. Cantoni, ''The cardinality balanced multi-target multi-Bernoulli filter and its implementations,'' *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, Feb. 2009.

[61] S. Reuter, B. T. Vo, B. N. Vo, and K. Dietmayer, ''The labeled multi-Bernoulli filter,'' *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3246–3260, Dec. 2014.

[62] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, ''Consensus CPHD filter for distributed multitarget tracking,'' *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 508–520, Jun. 2013.

[63] A. K. Gostar, T. Rathnayake, R. Tennakoon, A. Bab-Hadiashar, G. Battistelli, L. Chisci, and R. Hoseinnezhad, ''Centralized cooperative sensor fusion for dynamic sensor network with limited field-of-view via labeled multi-Bernoulli filter,'' *IEEE Trans. Signal Process.*, vol. 69, pp. 878–891, 2021.

[64] A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar, "Cauchy–Schwarz divergence-based distributed fusion with Poisson random finite sets," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Oct. 2017, pp. 112–116.

[65] H. G. Hoang, B.-N. Vo, B. T. Vo, and R. Mahler, "The Cauchy–Schwarz divergence for Poisson point processes," in *Proc. IEEE Workshop Stat. Signal Process. (SSP)*, Jun. 2014, pp. 240–243.

[66] H. Gia Hoang, B.-N. Vo, B.-T. Vo, and R. Mahler, "The Cauchy–Schwarz divergence for Poisson point processes," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4475–4485, Aug. 2015.

[67] A. K. Gostar, T. Rathnayake, A. Bab-Hadiashar, G. Battistelli, L. Chisci, and R. Hoseinnezhad, "Centralized multiple-view information fusion for multi-object tracking using labeled multi-Bernoulli filters," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Oct. 2018, pp. 238–243.

[68] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[69] M. Beard, B. T. Vo, and B.-N. Vo, "OSPA(2): Using the OSPA metric to evaluate multi-target tracking performance," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Oct. 2017, pp. 86–91.

[70] M. Beard, B. T. Vo, and B.-N. Vo, "A solution for large-scale multi-object tracking," *IEEE Trans. Signal Process.*, vol. 68, pp. 2754–2769, 2020.

[71] M. Frohle, K. Granstrom, and H. Wymeersch, "Decentralized Poisson multi-Bernoulli filtering for vehicle tracking," *IEEE Access*, vol. 8, pp. 126414–126427, 2020.

[72] K. Granstrom, M. Fatemi, and L. Svensson, "Poisson multi-Bernoulli mixture conjugate prior for multiple extended target filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 1, pp. 208–225, Feb. 2020.

[73] K. Granström, M. Fatemi, and L. Svensson, "Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli filter for extended target tracking," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 893–900.

[74] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[75] T. Li, J. M. Corchado, and S. Sun, "Partial consensus and conservative fusion of Gaussian mixtures for distributed PHD fusion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 5, pp. 2150–2163, Oct. 2019.

[76] L. Gao, G. Battistelli, L. Chisci, A. K. Gostar, and R. Hoseinnezhad, "Distributed joint mapping and registration with limited fields-of-view," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Oct. 2019, pp. 1–6.

[77] L. Gao, G. Battistelli, and L. Chisci, "Multiobject fusion with minimum information loss," *IEEE Signal Process. Lett.*, vol. 27, pp. 201–205, 2020.

[78] L. Gao, G. Battistelli, L. Chisci, and P. Wei, "Distributed joint sensor registration and multitarget tracking via sensor network," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1301–1317, Apr. 2020.

[79] T. Li, X. Wang, Y. Liang, and Q. Pan, "On arithmetic average fusion and its application for distributed multi-Bernoulli multitarget tracking," *IEEE Trans. Signal Process.*, vol. 68, pp. 2883–2896, 2020.

[80] T. Li, J. M. Corchado, and S. Sun, "On generalized covariance intersection for distributed PHD filtering and a simple but better alternative," in *Proc. 20th Int. Conf. Inf. Fusion (Fusion)*, Jul. 2017, pp. 1–8.

[81] T. Li, F. Hlawatsch, and P. M. Djuric, "Cardinality-consensus-based PHD filtering for distributed multitarget tracking," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 49–53, Jan. 2019.

[82] *Guide to Road Design Part 3: Geometric Design*, Ausroads, 2021.

JAMES KLUPACS received the bachelor's degree in mechatronics and manufacturing engineering from RMIT University, Melbourne, in 2019, where he is currently pursuing the Ph.D. degree. His current research interests include RFS filters, multitarget tracking, and data fusion.

AMIRALI KHODADADIAN GOSTAR received the B.Sc. degree in electrical engineering, the M.Sc. degree in philosophy of science, and the Ph.D. degree in mechatronics engineering from RMIT University. He is currently a Lecturer and an ARC DECRA Fellow with the School of Engineering, RMIT University. His research interests include sensor management, data fusion, and multitarget tracking.

THARINDU RATHNAYAKE received the B.Sc. degree (Hons.) in computer science from the University of Peradeniya, Sri Lanka, in 2013, and the Ph.D. degree in mechanical and manufacturing engineering from RMIT University, Australia, in 2018. He is currently working as a Research Fellow at the RMIT School of Engineering. His main research interests include computer vision, multi-target tracking, and machine learning.

IQBAL GONDAL led the Internet Commerce Security Laboratory (ICSL), Federation University, as the Director for last seven years to conduct translational research in Cybersecurity. He was responsible for establishing collaborative partnerships between Federation University and Australian Cyber Security Centre (ACSC) and Australian Federal Police (AFP). He was also as the Associate Dean Engagement at the STEM School, Federation University, where he successfully developed partnerships with domestic and international institutions for teaching and research. He is currently the Associate Dean of cloud, systems and security at RMIT University. He has worked in industry and academia for 25 years in Singapore and Australia. He has worked as a member of University Governing Council and a member of engineering advisory committee for Federation University, the Non-Exec Director of the Oceania Cyber Security Centre, and university engagement for the Defence Science Institute. In the past, he was the Director of ICT strategy for the Faculty of IT, Monash. He has received significant funding to support research in intelligent malware analysis, threat intelligence, fraud detection, cyber-attack triage, malware webinject detection, phishing attack identification, remote condition monitoring, and blockchain.

ALIREZA BAB-HADIASHAR (Senior Member, IEEE) received the B.Sc. and M.Eng. degrees in mechanical engineering and the Ph.D. degree in robotics from Monash University. He has held various positions with Monash University, Swinburne University of Technology, and RMIT University, where he is currently a Professor of mechatronics and leads the Intelligent Automation Research Group. His main research interests include intelligent automation in general, and robust data fitting in machine vision, deep learning for detection and identification, and robust data segmentation, in particular.

REZA HOSEINNEZHAD received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the University of Tehran, Iran, in 1994, 1996, and 2002, respectively. He has held various positions at the University of Tehran, Swinburne University of Technology, The University of Melbourne, and RMIT University, where he has been working, since 2010. He is currently a Professor and the Research Development Lead as well as the Discipline Leader of manufacturing and mechatronics at the School of Engineering. His main research interests include statistical information fusion, random finite sets, multi-object tracking, deep learning, and robust multi-structure data fitting in computer vision.