# Distributed Complementary Fusion for Connected Vehicles

James Klupacs, Amirali Khodadadian Gostar, Alireza Bab-Hadiashar, Jennifer Palmer, and Reza Hosseinezhad RMIT University

Melbourne, Australia

## Abstract

We present a random finite set-based method for achieving comprehensive situation awareness by each vehicle in a distributed vehicle network. Our solution is designed for labeled multi-Bernoulli filters running in each vehicle. It involves complementary fusion of sensor information locally running through consensus iterations. We introduce a novel label merging algorithm to eliminate double counting. We also extend the label space to incorporate sensor identities. This helps to overcome label inconsistencies. We show that the proposed algorithm is able to outperform the standard LMB filter using a distributed complementary approach with limited fields of view.

## **Index Terms**

Random Finite Sets, Intelligent Transport Systems, Multi-Object Tracking, Information Fusion, Distributed Algorithms

## I. INTRODUCTION

The emergence of *connected vehicles* in near future, as a dominant part of intelligent transport systems, is inevitable. At any time, there may be tens of vehicles in an area, with an unknown number of potential obstacles such as roadside debris or pedestrians. Whether the connected vehicle is driven, semi-autonomous, or fully-autonomous, achieving *comprehensive* situational awareness is essential for provision of effective driver-assist or self-driving commands, ensuring safety and efficiency.

In practice, the connected vehicles are equipped with sensors that each have limited field-of-view (FoV). Hence, to achieve *comprehensive* situational awareness, a *complementary* sensor fusion algorithm should be in place.

Our in proposed distributed vehicle network structure, we assume that situational awareness is acquired from a statistical model; a multi-object density to be more precise. Having the density, we can then estimate the number of objects and their states (e.g. location, speed). In addition, we can measure the *accuracy* of or *confidence* in the reliability of the estimates, by investigating statistical properties of the density.

Assuming that a multi-object posterior is formed in each vehicle node of the network as the result of running a multi-object tracking (MOT) filter. The recently developed MOT filters under the random finite set (RFS) framework [1]–[3] have shown promising results in various applications such as track-before-detect visual tracking [4], sensor management in target tracking applications [5], [6] and information fusion [7]. We choose the labeled multi-Bernoulli (LMB) filter as the filter of choice. An LMB posterior is fully characterized by a number of possibly existing objects each associated with label, a probability of existence and a single-object density for the object state.

Distributed networks are the preferred method of computing these problems as they are infinitely scalable with nodes, that is they scale in a linear fashion. This is in contrast to centralized networks which although are simpler to implement, and allow for the optimal solution to be computed the cost scales exponentially. In an ITS scenario the only possible method to potentially achieve real time networks is to use distributed computing. Much work has been done in the area of distributed RFS filters [8]–[11] however they have generally been limited to using Kullback-Leibler Averaging (KLA) or consider unlimited FoVs and have not been applied to the use case of connected vehicles with their challenges of complex motion, large sensor networks and limited FoVs.

The distributed network of connected vehicles has a *dynamic* nature, in the sense that its structure changes with time. Note that in this dynamic network, each node itself is among the objects that need to be detected and its state to be estimated by others, i.e. it is part of situational awareness. The dynamic changes mean that there is no way of guaranteeing object labels computed and propagated within the MOT filter in one node will be the same at another [12]. For example, a vehicle that is already labeled in the one node, is given a different (newly born label) when entering the FoV of another node. In addition, complementary fusion does not intrinsically avoid double-counting of information.

The main contributions of this paper are: (1) a distributed implementation of the complementary fusion method, which was previously proposed in [13], revised for use in a time-varying connected vehicle network, (2) a merging solution to remove double-counting that is inevitable with complementary fusion, and (3) a novel solution to overcome the label inconsistency present in distributed network.

## II. BACKGROUND

# A. Network model

In this paper we consider a sensor network denoted by  $(\mathcal{N}, \mathcal{A})$  where  $\mathcal{N}$  is the set of nodes and  $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$  represent the connections between each node. Two nodes *i* and *i'* are connected, that is able to communicate and exchange information, if  $(i, i') \in \mathcal{A}$ . Each node  $i \in \mathcal{N}$  has a set of neighbors (including itself) denoted by  $\mathcal{N}^i \triangleq \{i' \in \mathcal{N} : (i, i') \in \mathcal{A}\}$ . The above sensor network is fully distributed where each node *i* can only communicate directly with its neighbors. All operations (computation, measurement acquisition) occur locally at each node, without knowledge of the entire network.

## B. Information model

At each node, the local sensor measurements are processed via a statistical filter, resulting a multi-object density denoted by  $\pi_{i,k}$  where k is an index to discrete time of measurement acquisition. Situational awareness is achieved by estimating the number and states of various objects (with labels) in the scene from this multi-object density. Being a statistical model enables us to also assess the *confidence* in the resulting situational awareness (the estimates) too.

We note that different nodes (vehicles) in the network may be equipped with different types of sensors with various modalities, resolutions and sensitivities. The locally computed multi-object density plays the role of *common information representation* that is independent form sensor types and modalities. Importantly, there are application-specific (non-measurement) information such as statistical models for object state transitions (motion model), object birth and object death that are incorporated into the resulting multi-object density via applying the statistical filter.

In the network, each node communicates it locally formed multi-object density to its neighbors, and in-turn, receives theirs. The received densities are then fused to acquire a more informed density that is anticipated to return more reliable situational awareness.

## C. LMB density

The multi-object density of choice in this work is a particular RFS density called *labeled multi-Bernoulli* (LMB). This formulation approximates an RFS with the ensemble of multiple *possibly and independently existing* objects. Each object has its label, and is parametrized by its probability of existence and the multivariate density of its state vector. The LMB density is entirely characterized by the parameters of its (possibly existing) objects.

In the context of distributed fusion in connected vehicle networks, at each vehicle node i, the LMB density that is locally acquired at time k, is parametrized as:

$$\boldsymbol{\pi}_{i,k} \sim \left\{ \left( \boldsymbol{r}_{k,i}^{(\ell)}, \, \boldsymbol{p}_{k,i}^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_k} \tag{1}$$

where  $\ell$  denotes a single-object label. In addition,  $r^{(\ell)}$  and  $p^{(\ell)}(\cdot)$  denote the probability of existence and the multivariate density of the single-object state vector, for the object labeled  $\ell$ , respectively.  $\mathbb{L}_k$  denotes the space of all object labels discovered and tracked up to time k.

This paper considers the sequential Monte Carlo (SMC) approximation of the single-object densities. Each multivariate density,  $p_{k,i}^{(\ell)}(\cdot)$  is represented by particles and their weights as follows:

$$p_{k,i}^{(\ell)}(\cdot) \cong \sum_{j=1}^{J_{k,i}^{(\ell)}} w_{k,i,j}^{(\ell)} \,\delta(x - x_{k,i,j}^{(\ell)}) \tag{2}$$

where  $J_{k,i}^{(\ell)}$  denotes the number of particles which are themselves denoted each by  $x_{k,i,j}^{(\ell)}$  and their weights, each by  $w_{k,i,j}^{(\ell)}$ .

**Remark 1.** Given the multi-object density as an LMB given by (1) and (2), situational awareness can be simply achieved by: (1) considering each object whose probability of existence exceeds a large threshold (e.g. 0.90) as existing, and (2) estimating its state as the weighted average of its particles (this is indeed an Expected A Posteriori - EAP - estimate).

# D. LMB filter

The parameters of the LMB posterior are propagated through the prediction and update steps of an LMB filter [14]. At each time k, the current LMB density  $\pi_{i,k}$  parameterized by (1) is input as *prior* to the prediction step.

In prediction step, the state of each possibly existing object is propagates according to a single-object state transition model, also called *motion model*. The possible exit of each object from the scene (FoV of the vehicle/sensor node in the network) is also taken into consideration via a particular parameter, called *probability of survival*, which is commonly defined as a function of the object state. More importantly, the set of possibly existing objects with propagated states is appended with the set of *possibly born* objects at time k. This set formulates the *birth process* and normally includes a number of components with small probabilities of existence and densities that are peaked around possible points of entry to the scene. Each element of

this set is labeled (k, m) where k denotes the time of birth and m is an index. For instance, if there are four birth objects, their labels are  $\mathbb{B}_k = \{(k, 1), \dots, (k, 4)\}$ . Thus, the space of labels is extended as well to  $\mathbb{L}_{k+1} = \mathbb{L}_k \cup \mathbb{B}_k$ .

The update step inputs the outcome of the prediction step (predicted LMB). Then, Bayes' rule is applied to incorporate the information *measured by sensors*. Uncertainties in terms of false negatives (missing an object and returning no measurements for it) and false positives (false measurements that are not associated with any object - also clutter measurements) are modeled by a state-dependent probability of detection and an RFS model assumed for clutter measurements (normally a Poisson RFS), respectively.

Details of prediction and update steps and their derivations can be found in [14]. The resulting LMB density is the new posterior, denoted by:

$$\pi_{i,k+1} \sim \left\{ \left( r_{k+1,i}^{(\ell)}, \, p_{k+1,i}^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_{k+1}}.$$
(3)

**Remark 2.** For the sake of computational tractability, the components with very low probability of existence are *pruned*. Additionally, the particles of each remaining object are *resampled* to avoid particle deprivation.

# **III. DISTRIBUTED INFORMATION FUSION**

In order to achieve comprehensive situational awareness in each vehicle node, we suggest a distributed information fusion scheme that is comprised of two major components: complementary fusion and consensus iterations.

## A. Complementary fusion

In each vehicle node *i* at time *k*, all the locally acquired information are encapsulated in the posterior  $\pi_{i,k}$ . Information fusion occurs by fusing this locally generated posterior with all those communicated from neighboring nodes. We use the complementary fusion scheme first devised by Gostar et al. in [13]. In this scheme, the fused posterior is density of the union of all the posteriors:

$$\bar{\pi}_{i,k} \sim \bigcup_{i' \in \mathcal{N}^i \cup \{i\}} \left\{ \left( r_{k,i'}^{(\ell)}, p_{k,i'}^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_k} \tag{4}$$

approximated by an LMB. The resulting LMB parameters are then given by:

$$\bar{r}_{k,i}^{(\ell)} = \sum_{i' \in \mathcal{N}^{i} \cup \{i\}} \varrho_{k,i'}^{(\ell)} / \left[ 1 + \sum_{i' \in \mathcal{N}^{i} \cup \{i\}} \varrho_{k,i'}^{(\ell)} \right]$$
(5)

$$\bar{p}_{k,i}^{(\ell)}(\cdot) = \sum_{i' \in \mathcal{N}^i \cup \{i\}} \varrho_{k,i'}^{(\ell)} p_{k,i'}^{(\ell)}(\cdot) / \sum_{i' \in \mathcal{N}^i \cup \{i\}} \varrho_{k,i'}^{(\ell)}$$
(6)

where

$$\varrho_{k,i}^{(\ell)} = r_{k,i}^{(\ell)} / \left[ 1 - r_{k,i}^{(\ell)} \right]. \tag{7}$$

**Remark 3.** For each object label  $\ell$ , the fused probability of existence given by (5) approaches one if one of the  $r_{k,i}$  values is close to 1.

The observation made in Remark 3 demonstrates the *complementary* nature of information fusion. The safety-critical nature of connected vehicle system technology necessitates comprehensive situational awareness, and avoiding false negatives (miss-detection) is a high priority. This is surmounted by the limited FoV of sensors onboard the vehicles, and the complementary fusion method is hence, a suitable solution.

## B. Consensus

Despite the complementary nature of fusion and its power of inclusion, still the resulting multi-object posterior would be updated up to one layer of neighbors around each vehicle. To ensure a more comprehensive situational awareness is achieved, we propose that during each filtering step (from each time k to k+1), the information gathered by vehicle nodes are iteratively propagated through multiple layers of neighbors to each node.

The most common approach for information propagation in distributed networks is called the *consensus* approach [15]. Consider each node i, with a set of neighbors  $j \in \mathcal{N}^i$ . Let us denote the index to consensus iterations by  $\xi$ , and in general, consider a parameter estimate  $\hat{\theta}_{k,i}$  at node i (after fusion). In the original *consensus* process, the parameter estimate is iteratively updated as follows:

$$\hat{\theta}_{k,i}(\xi) = \sum_{i' \in \mathcal{N}^i} \omega_{i,i'} \hat{\theta}_{k,i}(\xi - 1), \tag{8}$$

where  $\omega_{i,i'}$  denotes normalized importance weights for neighboring nodes  $(\sum_{i' \in \mathcal{N}^i} \omega_{i,i'} = 1)$ , not to be confused with particle weights  $(w_{k,i,j}^{(\ell)})$ . The iterations stop when the change in estimate becomes negligible.

In the proposed solution, we apply the consensus approach with two specific variations. Firstly, instead of a convergence criterion for stopping the iterations, we iterate information propagation for a fixed number of times. This parameter is the practical radius of coverage (number of layers of coverage in the network around each vehicle). Secondly, we do not discriminate between the neighboring vehicles and assign equal weights.

**Remark 4.** To clarify, let us assume that information exchange occurs for five consensus iterations, and each vehicle has a FoV with 40 m range around it. This means that after completion of the consensus iterations (each including complementary fusion of the multi-object densities received from neighbors), each vehicle will have a map of environment around it that extends to 40 m beyond five vehicles around it.

# IV. SMC IMPLEMENTATION

Consider the vehicle node *i* in the network and its locally filtered multi-object posterior at time *k*, denoted by  $\pi_{i,k}$  in which the density of each object state with label  $\ell$  is approximated by particles and weights  $\left\{ \left( w_{k,i,j}^{(\ell)}, x_{k,i,j}^{(\ell)} \right) \right\}_{j=1:J_{k,i}^{(\ell)}}$ .

Implementation of complementary fusion in each vehicle node is straightforward for probabilities of existence. First the  $\varrho_{k,i}^{(\ell)}$  terms are computed as in equation (7), then the fused probability of existence is calculated via equation (5). Implementing the fusion of single-object densities via (6) means that at each node *i*, for each object label  $\ell$ , all the particles at node *i* and its neighbors  $i' \in \mathcal{N}^i$  are concatenated, and their weights are rescaled by a factor of  $\varrho_{k,i}^{(\ell)}$  and normalized. To keep the operation computationally tractable, the large ensemble of concatenated particles are resampled.

The consensus process is implemented as through iterations  $\xi = 1 : N_{\text{consensus}}$  during the sampling time period between k and k + 1. In each iteration  $\xi$ , the vehicle node i receives the local posteriors  $\pi_{i',k}(\xi)$  from its neighbors and fuses them together with its own local posterior  $\pi_{i,k}(\xi)$ . The result will be  $\pi_{i,k}(\xi + 1)$ .

## A. Tracking Challenges

The dynamic and distributed nature of the vehicle network brings in some challenges related to tracking of objects as part of situational awareness. Speaking of *tracking*, we emphasize on the labels of objects and accurate propagation of those labels through the fusion and consensus processes running in each vehicle/sensor node.

The repeated unification of all Bernoulli components through complementary fusion through consensus iterations, and the movements of vehicles in the network can lead to two major challenges. The first challenge is the possibility of two different objects being assigned the same label. An example is shown in Fig. 1 where two different objects enter the FoV of two different vehicle nodes at the same time k. Hence, as part of the birth process locally running at each node, they may be both labeled as (k, m), and their probabilities of existence will increase due to their detection within the FoV of each node. When the resulting LMB posteriors at nodes 1 and 2 are fused, the label inconsistency will cause drastic errors treating two separate objects as one.

Our proposed solution is straightforward and very intuitive. We propose that at each node, the node identity is appended to the labels of newly detected objects, i.e. object labels are formed as a triple (k, i, m) where i is the identity of the sensor node where the object was detected for the first time.

Another major challenge is that the complementary fusion scheme is designed to be comprehensive and the occurrence of information *double-counting* is very likely. In particular, we may end up with multiple labels assigned to the same object. An example is shown in Fig. 2.

An object T moves from time  $k_1$  till time  $k_2$ . At time  $k_1$ , it enters the FoV of vehicle node  $V_1$  (with identity i = 1) and is detected as a newly born object with label  $(k_1, 1, m_1)$  where  $m_1$  can be any number from 1 to the number of birth components in the local filter running at the node. Later at time  $k_2$ , the same object enters the FoV of vehicle node  $V_2$  (i = 2) and is given a new label of  $(k_2, 2, m_2)$ . This is while, in the LMB posterior that is formed in node  $V_1$  and communicated and fused in node  $V_2$ , the same object is represented with a Bernoulli component labeled  $(k_1, 1, m_1)$ .

Our remedy for the double-counting issue is formulated based on the fact that same object represented with two different labels is expected to have density particles that are centered very close to each other. Indeed, by investigating the distance between the EAP state estimates of the two objects, we can merge them into one if the distance is below a small threshold.

When the two Bernoulli components with different labels are merged, their labels, probabilities of existence, and state densities need to combine. The merged label can be simply chosen as the *older* label. The probabilities of existence can be combined according to the union rule: sum of the two minus their product. The densities are merged by concatenating the particles with their weights being normalized. We propose to scale the weights of particles of each Bernoulli component by its probability of existence, i.e. giving more emphasis on the particles of the Bernoulli component that is more likely to exist. Obviously, the particles need to be resampled at the end, to maintain computational tractability. Algorithm 1 presents the step-by-step pseudocode for the merging operation.



Figure 1: An example of label inconsistency in the proposed distributed fusion scheme. Circles represent each node's FoV. Two sensor nodes in the network (blue) each detect a vehicle (orange) that is entering their FoV at the same time k. The orange vehicles are different objects but may be assigned the same label (k, m).

## V. SIMULATION RESULTS

We evaluated the performance of our proposed distributed fusion method using the driving scenario designer application in MATLAB. The scenario contains 10 vehicles following a realistic and complex trajectory including sharp turns with large acceleration variations. The scenario encompasses a  $200 \text{ m} \times 200 \text{ m}$  area, as shown in Fig. 3.

Each vehicle has a single radar sensor with a limited field of view. Mobile radars have 50 m range. There is also a stationary radar with a range of 120 m. All sensors have 360 degree coverage. Vehicles are modeled as  $0.1 \text{ cm} \times 0.1 \text{ cm}$  points to ensure that a single measurement is generated per sensor. Each vehicle's state includes its position, velocity and turning rate:  $x = [p_x \ p_y \ \dot{p}_x \ \dot{p}_y \ \omega]^\top$ . Motion was approximated by a constant turn (CT) model with a 0.1 second sampling period. In CT model, the single-object state transition density is given by  $f_{k|k-1}(\cdot|x) = \mathcal{N}(\cdot;m(x),Q)$  where  $m(x) = [(F(\omega) \times x)^\top \ \omega]^\top$ , and  $Q = \text{diag}(\sigma_{\omega}^2 G G^\top, \sigma_u^2)$  in which  $\sigma_{\omega} = 15 \text{ m/s}^2$  and  $\sigma_u = 30^\circ/\text{s}$ , and

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

Finally, for each sensor, object birth is modelled as an LMB with its elements covering the rim of the detection range.

The dynamic nature of scenario entails the vehicles entering and exiting the scene at various times. The ground-truth times are listed in Table I.

In our experiments, we examined the distributed complementary fusion with consensus once without and once with our proposed solutions for tracking challenges. In the former case, we use the term "Distributed LMB", and in the latter, "Our method" in the figure legends.



Figure 2: An example of double counting in a distributed RFS network (top) and label inconsistency (bottom). Circles represent each nodes FoV. The double counting illustration highlights the potential for a nodes posterior to have multiple labels belonging to the same target. The evolution between two times is shown, with an orange target vehicle moving between  $k_1$  and  $k_2$ . Blue vehicles represent vehicles at the first time, while yellow is their evolution at the second. The bottom shows two vehicles (blue) each running independent RFS filters detecting a target vehicle (orange) at the same time and assigning the same label to each, which when communicated presents inconsistency in target information.

| Tabl | le I | : ( | Ground | l-truth | entry | (birth) | and | exit | (death) | ) times |
|------|------|-----|--------|---------|-------|---------|-----|------|---------|---------|
|------|------|-----|--------|---------|-------|---------|-----|------|---------|---------|

| Veh. no. | $k_{ m birth}$ | $k_{ m death}$ | Veh. no. | $k_{ m birth}$ | $k_{ m death}$ |
|----------|----------------|----------------|----------|----------------|----------------|
| 1        | 1              | 232            | 6        | 12             | 153            |
| 2        | 32             | 274            | 7        | 12             | 211            |
| 3        | 2              | 224            | 8        | 2              | 193            |
| 4        | 22             | 155            | 9        | 72             | 311            |
| 5        | 52             | 289            | 10       | 52             | 205            |

Figure 4 shows the cardinality (number of objects returned in estimates extracted from local LMB posterior) values averaged across all sensors in the network. Evidently, due to double-counting and label inconsistency, objects are included with multiple labels and with original label, there will be different objects with the same labels. The result (green \* points in Fig. 4) shows a persisting overestimate of the number of objects. This is while after putting our proposed extension of labels (to include sensor identities) and merging algorithm, almost all the time, we get accurate cardinality estimates.

The impact of cardinality overestimation on  $OSPA^{(2)}$  errors [16] is demonstrated in Fig. 5. The error is significantly reduced when our method is applied. The tracking results of the proposed method are presented for a single iteration in Fig. 6. We can see that despite the reasonable amount of clutter (representing false alarms and other objects in the scene) for an ITS scenario, each object is successfully identified and tracked for the entire duration of the simulation. Tracks are not lost, although there are occasionally false tracks generated which are quickly killed off. Despite the initial spike in  $OSPA^{(2)}$  (Fig. 5), showing a delay in obtaining the tracks, measured over a window of 10 steps (1 second), once they are obtained, they show stable

<u>OUTPUTS:</u>  $\bar{\ell}, r_{k,i}^{(\bar{\ell})}, \left\{ \left( w_{k,i,j}^{(\bar{\ell})}, x_{k,i,j}^{(\bar{\ell})} \right) \right\}_{i=1:\bar{L}}$ 

5:  $\alpha_1 \leftarrow r_{k,i}^{(\ell_1)} / \left( r_{k,i}^{(\ell_1)} + r_{k,i}^{(\ell_2)} \right); \ \alpha_2 \leftarrow (1 - \alpha_1)$ 

7: for  $j = 1: J_{k,i}^{(\ell_1)}$  do  $x \leftarrow x \cup \left\{ \left( \alpha_1 w_{k,i,j}^{(\ell_1)}, x_{k,i,j}^{(\ell_1)} \right) \right\}$ 

9: for  $j = 1: J_{k,i}^{(\ell_2)}$  do  $x \leftarrow x \cup \left\{ \left( \alpha_2 \, w_{k,i,j}^{(\ell_2)}, x_{k,i,j}^{(\ell_2)} \right) \right\}$ 

1: if  $k_1 < k_2$  then  $\bar{\ell} \leftarrow \ell_1$ 

4:  $r_{k,i}^{(\bar{\ell})} \leftarrow r_{k,i}^{(\ell_1)} + r_{k,i}^{(\ell_2)} - r_{k,i}^{(\ell_1)} r_{k,i}^{(\ell_2)}$ 

11:  $\left\{x_{k,i,j}^{(\bar{\ell})}\right\}_{j=1:\bar{J}} = \text{RESAMPLE}(\boldsymbol{x},\bar{J})$ 12: for  $j = 1:\bar{J}$  do  $w_{k,i,j}^{(\bar{\ell})} \leftarrow 1/\bar{J}$ 

weighted particles.

2: else  $\bar{\ell} \leftarrow \ell_2$ 3: end if

6:  $x \leftarrow \emptyset$ 

8: end for

10: end for

13: end for

$$\underline{\text{INPUTS:}} \ \ell_1, \ell_2, r_{k,i}^{(\ell_1)}, r_{k,i}^{(\ell_2)}, \left\{ \left( w_{k,i,j}^{(\ell_1)}, x_{k,i,j}^{(\ell_1)} \right) \right\}_{j=1:J_{k,i}^{(\ell_1)}}, \left\{ \left( w_{k,i,j}^{(\ell_2)}, x_{k,i,j}^{(\ell_2)} \right) \right\}_{j=1:J_{k,i}^{(\ell_2)}}, \bar{J}$$

 $\triangleright \overline{J}$ : the desired number of particles after merging.  $\triangleright \ell_1 = (k_1, i_1, m_1), \ \ell_2 = (k_2, i_2, m_2), \ k \ge k_1, \ k \ge k_2$ 

▷ Merged prob. of existence and particles and label.

 $\triangleright$  Merged label is the older one.

▷ Merged prob. of existence computed by union rule.

 $\triangleright$  Scale factors set for particles of  $(\ell_1)$  and  $\ell_2$  $\triangleright x$  to contain all pairs of particles and weights.

 $\triangleright$  Concatenated particles are resampled into  $\overline{J}$  equally

performance, except at the spike between k = 200 and k = 250, as well as around k = 300. This is due to targets leaving the scene and the subsequent cardinality error.

## VI. CONCLUSION

This paper presented a new distributed information fusion algorithm using a consensus-based complementary fusion method. A novel solution to solving the double-counting and label inconsistency problems was introduced and shown to perform with satisfactory in a complex connected driving scenario.

#### ACKNOWLEDGMENT

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

## REFERENCES

- [1] R. Mahler, Advances in Statistical Multisource-Multitarget Information Fusion. Electronic Warfare, Artech House, 2014.
- [2] L. Gao, G. Battistelli, L. Chisci, A. K. Gostar, and R. Hoseinnezhad, "Distributed joint mapping and registration with limited fields-of-view," in 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), pp. 1–6, 2019.
- [3] 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 Processing*, vol. 167, p. 107278, 2020.
- [4] T. Rathnayake, A. Khodadadian Gostar, R. Hoseinnezhad, R. Tennakoon, and A. Bab-Hadiashar, "On-line visual tracking with occlusion handling," Sensors, vol. 20, no. 3, 2020.
- [5] S. Panicker, A. K. Gostar, A. Bab-Hadiashar, and R. Hoseinnezhad, "Tracking of targets of interest using labeled multi-bernoulli filter with multi-sensor control," Signal Processing, vol. 171, p. 107451, 2020.
- [6] S. Panicker, A. K. Gostar, A. Bab-Haidashar, and R. Hoseinnezhad, "Sensor control for selective object tracking using labeled multi-bernoulli filter," in 2018 21st International Conference on Information Fusion (FUSION), pp. 2218–2224, 2018.
- [7] 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 Transactions on Signal Processing*, vol. 69, pp. 878–891, 2021.
- [8] T. Li and F. Hlawatsch, "A distributed particle-PHD filter with arithmetic-average PHD fusion," 2018.



Figure 3: Setup of the simulation. Colored lines represent individual trajectories, which are defined by their respective coloured waypoints.

- [9] M. Fröhle, K. Granström, and H. Wymeersch, "Decentralized Poisson multi-Bernoulli filtering for vehicle tracking," *IEEE Access*, vol. 8, pp. 126414– 126427, 2020.
- [10] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE Journal on Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, 2013. cited By 145.
  [11] K. Shen, P. Dong, Z. Jing, and H. Leung, "Consensus-based labeled multi-bernoulli filter for multitarget tracking in distributed sensor network," *IEEE*
- [11] K. Shen, P. Dong, Z. Jing, and H. Leung, "Consensus-based labeled multi-bernoulli filter for multitarget tracking in distributed sensor network," IEEE Transactions on Cybernetics, pp. 1–12, 2021.
- [12] S. Li, W. Yi, R. Hoseinnezhad, G. Battistelli, B. Wang, and L. Kong, "Robust distributed fusion with labeled random finite sets," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 278–293, 2018.
- [13] A. K. Gostar, T. Rathnayake, A. Bab-Hadiashar, G. Battistelli, L. Chisci, and R. Hoseinnezhad, "Centralized multiple-view information fusion for multiobject tracking using labeled multi-Bernoulli filters," in 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), pp. 238–243, IEEE, 2018.
- [14] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, 2014. cited By 348.
- [15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [16] M. Beard, B. T. Vo, and B.-N. Vo, "Ospa(2): Using the ospa metric to evaluate multi-target tracking performance," in 2017 International Conference on Control, Automation and Information Sciences (ICCAIS), pp. 86–91, 2017.



Figure 4: Cardinality estimates for 10 iterations averaged across all sensors. 3 Consensus Iterations.



Figure 5:  $OSPA^{(2)}$  estimates for 10 iterations averaged across all sensors for a window length of 10, with order 1 and cutoff 50. 3 Consensus Iterations.



Figure 6: track estimates estimates for a single iteration. 3 Consensus Iterations.